



**FACULTAD DE CIENCIAS E INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA
ELECTRÓNICA CON MENCIÓN EN
TELECOMUNICACIONES**

TESIS

**Para optar el título profesional de Ingeniero Electrónico con
Mención en Telecomunicaciones**

Diseño e implementación de una central telefónica Voip de bajo
costo mediante Asterisk y Raspberry Pi para pequeñas o
medianas empresas

PRESENTADO POR

Sanchez Espinoza, David Junior

ASESOR

Soto Córdova, Martin

Los Olivos, 2021

RESUMEN

La comunicación es un elemento importante en el desarrollo de una sociedad. Tanto en el marco laboral, como social, establecer un sistema de comunicación rápida y confiable se vuelve necesario. Por ello, se implementaron los sistemas y servicios de telefonía tradicional, tal y como se conocen y se vienen utilizando a nivel global. Debido a creciente y continua demanda, estos servicios tienden a mejorar en prestaciones, calidad e integración con nuevas aplicaciones, conduciendo hacia nuevos servicios de telefonía y comunicaciones unificadas, generando un despliegue increíble de los sistemas de comunicación por voz. En este proyecto, se realiza el diseño e implementación de una central telefónica *VoIP*. Se describe el funcionamiento y las características principales que una central telefónica puede ofrecer, sea en pequeñas o medianas empresas, y entornos domésticos, que requieran servicios de telefonía *VoIP*, a un menor costo. Se destaca la utilización del hardware y software. Para este caso, la *Raspberry Pi*, una pequeña y potente computadora de bajo costo, que trabaja con sistemas operativos basados en Linux. En relación al software, se utiliza *Asterisk*, dedicado especialmente a la telefonía, además de ser de código libre y de fácil acceso. Finalmente, para la evaluación y definición de la capacidad del sistema, se realizan pruebas de comunicaciones en condiciones de estrés físicas y virtuales, aplicando *software* libre, como *SIPP*. Todo ello conduce a una gestión correcta, el conocimiento de su operatividad y la evaluación de su rendimiento, con el fin de satisfacer las necesidades funcionales en el campo, teniendo en cuenta la calidad de servicio.

Palabras clave: Comunicación, Telefonía tradicional, Software, Telefonía *VoIP*, *Raspberry Pi*, *Asterisk*, *SIPP*, Calidad de servicio.

ABSTRACT

Communication is an important element in the development of a society. Both in the labor and social framework, establishing a fast and reliable communication system becomes necessary. For this reason, traditional telephony systems and services were implemented, as they are known and have been used globally. Due to increasing and continuous demand, these services tend to improve in performance, quality and integration with new applications, leading to new telephony and unified communications services, generating an incredible deployment of voice communication systems. In this project, the design and implementation of a VoIP telephone exchange is carried out. It describes the operation and the main characteristics that a telephone exchange can offer, whether in small or medium-sized companies, and domestic environments, which require VoIP telephony services, at a lower cost. The use of hardware and software is highlighted. For this case, the Raspberry Pi, a small and powerful low-cost computer that works with Linux-based operating systems. In relation to the software, Asterisk is used, dedicated especially to telephony, in addition to being open source and easily accessible. Finally, for the evaluation and definition of the system capacity, communications tests are carried out under physical and virtual stress conditions, applying free software, such as SIPPP. All this leads to correct management, knowledge of its operability and evaluation of its performance, in order to satisfy functional needs in the field, considering the quality of service.

Keywords: *Communication, Traditional telephony, Software, VoIP telephony, Raspberry Pi, Asterisk, SIPPP, Quality of service.*

CONTENIDO

RESUMEN.....	ii
ABSTRACT.....	iii
CONTENIDO.....	iv
Lista de figuras	vii
Lista de tablas.....	ix
INTRODUCCIÓN.....	1
CAPÍTULO I: PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA	3
1.1. DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN	4
1.1.1. Planteamiento y descripción del problema.....	4
1.1.2. Formulación del problema general.....	5
1.1.3. Formulación de los problemas específicos.....	5
1.2. DEFINICIÓN DE LOS OBJETIVOS DE LA INVESTIGACIÓN	6
1.2.1. Objetivo general	6
1.2.2. Objetivos específicos.....	6
1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN	6
1.3.1. Justificación técnica	7
1.3.2. Justificación económica	7
1.3.3. Justificación social.....	8
1.4. ALCANCES Y LIMITACIONES DE LA INVESTIGACIÓN	8
1.4.1. Alcances	8
1.4.2. Limitaciones	9
CAPÍTULO II: FUNDAMENTO TEÓRICO.....	10
2.1. ANTECEDENTES DE LA INVESTIGACIÓN	11
2.1.1. Internacionales	11
2.1.2. Nacionales.....	12
2.2. MARCO TEORICO	13
2.2.1. Red Telefónica Publica Conmutada (PSTN).....	13
2.2.2. Voz sobre protocolo de Internet (<i>VoIP</i>) y Telefonía IP.....	14
2.2.3. Protocolos.....	16
2.2.4. <i>Codecs</i>	18
2.2.5. Calidad de Servicio.....	19
2.2.6. <i>Asterisk</i>	20
2.2.7. <i>FreePBX</i>	21

2.2.8. <i>Raspberry Pi 3</i>	21
2.3. MARCO METODOLÓGICO	23
2.3.1. Tipo de investigación	23
2.3.2. Metodología de la investigación	24
2.4. MARCO LEGAL.....	25
2.4.1. Recomendaciones Y.1540 y Y.1541.....	25
2.5. DIAGRAMA DE BLOQUES Y ARQUITECTURA DEL SISTEMA.....	27
CAPÍTULO III: DESARROLLO DE LA APLICACIÓN	30
3.1. DESARROLLO DEL OBJETIVO ESPECÍFICO 1: Evaluar la situación actual de los servicios de telefonía tradicional y los nuevos sistemas que han surgido hasta la actualidad.....	31
3.2. DESARROLLO DEL OBJETIVO ESPECÍFICO 2: Determinar los indicadores principales que se debe tener en cuenta para un servicio de telefonía.....	33
3.2.1. <i>QoS</i> (Calidad de Servicio).....	33
3.2.2. Costos por telefonía	34
3.3. DESARROLLO DEL OBJETIVO ESPECÍFICO 3: Diseñar una central telefónica <i>VoIP</i> que permita reducir los gastos de telefonía y mejorar la calidad de comunicación por voz.....	35
3.3.1. Diseño de la Red <i>VoIP</i> en General y/o Estándar.....	35
3.3.2. Instalación y Configuración de <i>RasPBX</i>	37
3.3.3. Configuración de <i>Asterisk</i>	38
3.3.4. Configuración de <i>Asterisk</i> por <i>FreePBX</i>	60
3.4. DESARROLLO DEL OBJETIVO ESPECÍFICO 4: Definir los <i>codecs</i> y protocolos adecuados para la central telefónica <i>VoIP</i>	70
3.4.1. Elección del <i>Codec</i>	70
3.4.2. Elección del protocolo de Señalización.....	72
3.4.3. Elección del Protocolo de Transporte	74
3.5. DESARROLLO DEL OBJETIVO ESPECÍFICO 5: Realizar pruebas de estrés a la central telefónica <i>VoIP</i> y medir la eficiencia y el rendimiento	75
3.5.1. Eficiencia y Rendimiento de la <i>Raspberry Pi</i> como central <i>VoIP</i>	78
3.5.2. Análisis de rendimiento con <i>SIPP</i>	80
3.5.3. Análisis de una llamada real	86
3.5.4. Análisis de <i>QoS</i> con <i>SIPP</i> y llamadas reales.....	96
3.5.5. Análisis de llamada entre centrales	99
3.5.6. Ancho de banda requerido para la central <i>VoIP</i>	103
3.5.7. Recomendaciones y aplicaciones en la implementación de la central telefónica en el entorno empresarial	106

CAPÍTULO IV: ANÁLISIS DE COSTO Y BENEFICIO	110
REFERENCIAS	119
GLOSARIO	123
ANEXOS	124
ANEXO 1: Instalación de <i>RasPBX</i>	124
ANEXO 2: Configuración Inicial de <i>RasPBX</i>	126
ANEXO 3: Configuración del archivo <i>extensions.conf</i>	127
ANEXO 4: Imágenes de Equipos	128

Lista de figuras

Figura 1: Esquema de una Red Telefónica Tradicional	14
Figura 2: Escenario básico de una red de telefonía IP	15
Figura 3: Arquitectura de Asterisk	21
Figura 4: Escritorio de la Raspberry Pi	23
Figura 5: Diagrama de bloques simplificado	24
Figura 6: Diagrama de bloques general	27
Figura 7: Arquitectura del sistema – Escenario 1	28
Figura 8: Arquitectura del sistema – Escenario 2	29
Figura 9: Comportamiento de la telefonía fija y móvil hasta el 2018	31
Figura 10: Comportamiento del uso de los teléfonos públicos.....	32
Figura 11: Topología Física de la red estándar de una empresa pequeña	35
Figura 12: Topología de la red local de comunicaciones.....	36
Figura 13: Sistema de buzón de voz	42
Figura 14: Acceso remoto de FreePBX	61
Figura 15: Acceso al panel de administración	62
Figura 16: Portada de administración de FreePBX.....	62
Figura 17: Panel general de SIP	63
Figura 18: Panel de configuración general de SIP	63
Figura 19: Pestaña Aplicaciones	64
Figura 20: Creación de una extensión SIP	65
Figura 21: Creación de una extensión IAX2.....	66
Figura 22: Adición de colas y agentes	66
Figura 23: Pestaña Conectividad	67
Figura 24: Configuración Outgoing de troncal SIP	67
Figura 25: Configuración Incoming de troncal SIP	68
Figura 26: Pestaña Reportes.....	69
Figura 27: Motor de búsqueda de CDR en FreePBX.....	69
Figura 28: Topología de red física para el sistema de comunicaciones	75
Figura 29: Topología Lógica de la red de comunicaciones	76
Figura 30: Escenario 5 llamadas simultaneas.....	81
Figura 31: Escenario 20 llamadas simultaneas.....	82
Figura 32: Escenario 40 llamadas simultaneas.....	82
Figura 33: Escenario 50 llamadas simultaneas.....	83
Figura 34: Escenario 60 llamadas exitosas.....	83
Figura 35: Escenario 80 llamadas simultaneas.....	84
Figura 36: Escenario 100 llamadas simultaneas	84
Figura 37: Escenario 500 llamadas simultaneas	85
Figura 38: Escenario 400 llamadas simultaneas	86
Figura 39: Captura de tráfico SIP en una llamada con Wireshark	88
Figura 40: Captura de tráfico RTP (G711u) en una llamada con Wireshark.....	89
Figura 41: Captura de tráfico RTP (G729) en una llamada con Wireshark	90
Figura 42: Llamadas fallidas y exitosas.....	91
Figura 43: Audio capturado con codec G729	92
Figura 44: Audio capturado con codec G711 ulaw	92
Figura 45: Flujo de llamada	93

Figura 46: Flujo de llamada fallida.....	94
Figura 47: Jitter y paquetes perdidos.....	94
Figura 48: Análisis de los paquetes de llamadas con G729 y G711.....	95
Figura 49: Escenario 50 llamadas y una real.....	96
Figura 50: Escenario 100 llamadas simultaneas y una llamada real.....	97
Figura 51: Flujo de llamadas entre centrales VoIP.....	100
Figura 52: Datos del primer mensaje INVITE(SDP) del flujo de llamada.....	100
Figura 53: Flujo RTP entre las llamadas realizadas.....	101
Figura 54: Jitter y Paquetes perdidos de unas llamadas entre centrales.....	102
Figura 55: Análisis de la media enviada del usuario al servidor.....	102
Figura 56: Calculadora de BW con G729.....	105
Figura 57: Calculadora de BW con G729.....	105
Figura 58: Calculo del valor actual neto en una tabla de excel.....	114
Figura 59: Calculo del TIR en una tabla de excel.....	115

Lista de tablas

Tabla 1: Protocolos de Señalización usados en sistemas <i>VoIP</i>	17
Tabla 2: Protocolos de Transporte más usados en sistemas <i>VoIP</i>	18
Tabla 3: Codificadores para <i>VoIP</i>	19
Tabla 4: Especificaciones técnicas de la <i>Raspberry Pi 3</i>	22
Tabla 5: Parámetros de <i>QoS</i> en una red	27
Tabla 6: Configuración [general] SIP	39
Tabla 7: Configuración de [extensión] SIP	39
Tabla 8: Configuración [general] IAX.....	40
Tabla 9: Configuración [general] del <i>voicemail.conf</i>	43
Tabla 10: Configuración [zonemessages] del <i>voicemail.conf</i>	44
Tabla 11: Configuración [default] del <i>voicemail.conf</i>	45
Tabla 12: Configuración [general] del <i>features.conf</i>	46
Tabla 13: Configuración [featuremap] del <i>feature.conf</i>	47
Tabla 14: Configuración [general] del <i>agents.conf</i>	48
Tabla 15: Configuración [agents] del <i>agents.conf</i>	48
Tabla 16: Configuración [general] del <i>queues.conf</i>	49
Tabla 17: Configuración [queues] para agentes dinámicos.....	49
Tabla 18: Configuración [queues1] para agentes estáticos	50
Tabla 19: Configuración [default] del <i>musiconhold.conf</i>	51
Tabla 20: Configuración de un enlace troncal SIP	52
Tabla 21: Configuración de una extensión para la PSTN.....	53
Tabla 22: Parámetros generales del <i>cdr.conf</i>	54
Tabla 23: Comparación entre <i>Codecs G711 VS G729</i>	72
Tabla 24: Diferencias entre IAX y SIP	73
Tabla 25: Información de <i>Codecs</i>	77
Tabla 26: Requerimientos para ciertas cantidades de llamadas simultáneas.....	78
Tabla 27: Uso del CPU y memoria frente a distintas cantidades de llamadas simultáneas.....	80
Tabla 28: Análisis de <i>QoS</i> en distintos escenarios	98
Tabla 29: Calidad de la central <i>VoIP</i> en distintos escenarios:.....	98
Tabla 30: Parámetros de <i>QoS</i>	99
Tabla 31: Costo de recursos humanos empleados en el proyecto	111
Tabla 32: Recursos materiales usados para el proyecto	112
Tabla 33: Flujo de caja anual.....	113

INTRODUCCIÓN

Uno de los factores importantes para que la humanidad se desarrolle ha sido y será la comunicación. El simple de hecho de hablar y escuchar, ha permitido entender, comprender y conocer los intereses y propósitos de todo ser vivo que habita en la tierra. En consecuencia, con el pasar de los tiempos, la humanidad con el fin de comunicarse con sus semejantes, sus métodos o sistemas de comunicación han ido evolucionando. Sobre todo, en comunicaciones a larga distancia. Desde simples actos primitivos (señales de humo) hasta la redacción de cartas o grabación de voz en casets. (Raffino, 2019) Se han venido desarrollando nuevos sistemas, equipos o dispositivos tecnológicos, con el fin de acortar distancias entre países, ciudades, pueblos, empresas, etc., y obtener una comunicación estable y segura.

El sistema o método de comunicación por voz más utilizado (inclusive hasta la actualidad) viene siendo la telefonía tradicional o llamado en general, la red telefónica publica conmutada (PSTN, *Public Switched Telephone Network*). Presentaba un sistema analógico (basada en conmutación de circuitos principalmente) y el manejo de la conexión entre usuarios era manual, pero al pasar los años, este proceso paso a ser automático dado los avances tecnológicos (ElastixTech, 2019). En la actualidad, se tienen nuevas tecnologías, y es notable la expansión del internet a nivel mundial. Es exorbitante el uso que se le ha dado, se ha sumergido en la vida diaria, presentando cada día, nuevas aplicaciones, programas, etc., volviéndose indispensable para el ser humano. Por supuesto, la telefonía no se quedó atrás, con el uso del internet y sus protocolos, permitió a la telefonía integrarla e implementarla a nivel de *software* (digital), surgiendo la telefonía IP o tecnologías de Voz sobre IP (*VoIP, Voice Over Internet Protocol*), que brinda mejoras en la calidad de servicio, reducción de costos, y lo más importante, permite adaptar los sistemas tradicionales de telefonía (PSTN) sin necesidad de implementar nuevos sistemas o equipos (Sinaeepourfard y Mohamed, 2011). En la actualidad, la mayoría de grandes y medianas empresas, poseen recursos tecnológicos para el transporte de datos e información (Sistema

de protocolos TCP/IP). La telefonía IP aprovecha estos medios, realizando un sistema de comunicación en tiempo real, a través de una red local.

Los sistemas *VoIP* permiten la transmisión de voz a través de una red IP (en forma de paquete de datos) alámbrica o inalámbrica. Los servicios de telefonía IP son aplicaciones directas de esta tecnología, permitiendo la realización de llamadas comunes, por medio de aplicaciones, entre computadoras, teléfonos inteligentes, teléfonos en *software* (*softphones*), teléfonos IP y los tradicionales. Mientras estén conectados a una red local o de internet (Rivera y Poma, 2014).

La telefonía IP ofrece múltiples ventajas frente a la telefonía convencional, como reducción de costos, mejor calidad de servicio de voz, optimización de funciones avanzadas, como IVR (*Interactive Voice Response*), ACD (*Automatic Call Distributor*), CTI (*Computer Telephony Integrations*), etc. (Querea. 2019). Por ello, en el mercado actual, existen numerosos servicios y aplicaciones para telefonía IP, sin embargo, muchos de ellos implican costos elevados por la implementación de estos servicios o por la prestación de sus sistemas de red telefónica. Teniendo en cuenta este inconveniente, surge la idea de este proyecto.

En el presente trabajo de investigación se propone diseñar e implementar una central telefónica *VoIP* de bajo costo mediante *Asterisk* y *Raspberry Pi* para pequeñas o medianas empresas y entornos domésticos. El dispositivo *Raspberry Pi* es un pequeño pero potente computador accesible (debido a su bajo costo), mientras que *Asterisk* es un *software* libre y gratuito dedicado especialmente a *VoIP*, siendo básicos para la implementación del presente proyecto. En el capítulo I se tiene el planteamiento y formulación del problema, se describe tanto el problema general como los problemas específicos, se define los objetivos, justificación, alcances y limitaciones de la investigación. El fundamento teórico en el capítulo II, junto con los antecedentes, el marco teórico y metodológico, además del marco legal y la arquitectura del sistema. Para el capítulo III se tiene el desarrollo de la aplicación. Este capítulo comprende el análisis de la situación actual, diseño y desarrollo de los objetivos que se plantearon en el proyecto de tesis. Todo ello se desarrollará según la metodología elegida. En el capítulo IV, se exponen los análisis de costo y beneficios correspondientes.

CAPÍTULO I: PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA

1.1. DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN

Los sistemas *VoIP* avanzan tecnológicamente cada día, ofreciendo mejoras en la calidad de servicios por voz, aplicaciones multimedia, administración de sistemas telefónicos fáciles de manejar, etc. A pesar de los numerosos beneficios que presenta, y de la exorbitante expansión del servicio de internet, la telefonía tradicional continúa siendo muy utilizada, sobre todo en el sector doméstico. Si bien, en sus inicios, el costo era menor frente a la telefonía móvil, y solventaba las necesidades de comunicación en hogares y empresas, con el pasar de los años, este servicio convencional ha elevado sus costos, sobre todo en comunicaciones a largas distancias o por mantenimiento complejo de la red telefónica (Bastidas, 2016).

1.1.1. Planteamiento y descripción del problema

Como todo servicio de comunicación que se adquiere, implica un egreso económico. Para un uso conveniente, es recomendable realizar una gestión correcta para reducir costos innecesarios. Normalmente, este proceso es indispensable en el sector empresarial, que incluyen una gran cantidad de usuarios con la necesidad de comunicarse entre ellos o exteriormente. Del mismo modo, con los avances tecnológicos, se han introducido nuevos servicios de mensajería, transmisión de voz y video, o envío de datos a través de una red de internet (Dong, 2011). Si bien estos servicios han solventado muchas necesidades, tanto técnicas como económicas, es aún necesario poseer un sistema de comunicación por voz, (sea local o externa) dada la velocidad en la que se transmite. Lo cual lo ocupa la telefonía tradicional, que en sus inicios era la solución perfecta, pues ahorra tiempo y lograba comunicación entre sus usuarios. Sin embargo, con el pasar de los años, este servicio viene presentando varias deficiencias, principalmente con la calidad de voz, la vulnerabilidad del sistema y el gasto económico que requiere. Adicionalmente a ello, se requiere la integración a nuevas aplicaciones para una comunicación unificada.

Si se tiene en cuenta la calidad de voz, la telefonía fija es analógica, de este modo, presenta serios problemas cuando de interferencia de ruido se trata. Normalmente las conexiones son alámbricas, esto implica el tratamiento del cable, ya que, al fallar este medio, imposibilita la comunicación. Así mismo, presentan más vulnerabilidad frente a otros sistemas de comunicación, lo cual implica el requerimiento de personal técnico que realice una administración de la seguridad; implicando un egreso económico extra, que tal vez para corporaciones grandes no sean un problema, pero si para medianas o pequeñas empresas, inclusive los entornos domésticos, pues requieren gastos financieros que a largo plazo no puedan afrontar (Bastidas, 2016). Esto conlleva a los costos por contratar un servicio de telefonía convencional, si bien la adquisición de sus productos no son un precio elevado, el mantenerlos activos y sin fallas técnicas, requieren gastos totalmente innecesarios. Sobre todo, para la creación de nuevos anexos (usuarios) y la implementación de nuevos equipos telefónicos, que según las empresas proveedoras de este servicio, mejoran la calidad de servicio, lo cual no es necesariamente cierto.

1.1.2. Formulación del problema general

¿Cómo disminuir los gastos en telefonía y mejorar la calidad de servicio de comunicación por voz en medianas o pequeñas empresas?

1.1.3. Formulación de los problemas específicos

P.E.1: ¿Cuál es la situación actual de los servicios de telefonía tradicional y que nuevos sistemas han surgido hasta la actualidad?

P.E.2: ¿Qué indicadores principalmente se debe tener en cuenta para un servicio de telefonía?

P.E.3: ¿Cómo diseñar una central telefónica *VoIP* que permita reducir los gastos de telefonía y mejorar la calidad de comunicación por voz?

P.E.4: ¿Cuáles serían los *codecs* y protocolos adecuados para la central telefónica *VoIP*?

P.E.5: ¿De qué manera se puede medir la eficiencia y el rendimiento de la central telefónica *VoIP*?

1.2. DEFINICIÓN DE LOS OBJETIVOS DE LA INVESTIGACIÓN

1.2.1. Objetivo general

Implementar una central telefónica *VoIP* mediante *Asterisk* y *Raspberry Pi* para reducir los gastos en telefonía y mejorar la calidad de servicio de comunicación por voz.

1.2.2. Objetivos específicos

O.E.1: Evaluar la situación actual de los servicios de telefonía tradicional y los nuevos sistemas que han surgido hasta la actualidad.

O.E.2: Determinar los indicadores principales que se debe tener en cuenta para un servicio de telefonía.

O.E.3: Diseñar una central telefónica *VoIP* que permita reducir los gastos de telefonía y mejorar la calidad de comunicación por voz.

O.E.4: Definir los *codecs* y protocolos adecuados para la central telefónica *VoIP*.

O.E.5: Realizar pruebas de estrés a la central telefónica *VoIP* y medir la eficiencia y el rendimiento.

1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN

Poco a poco la telefonía tradicional se viene tornando obsoleta debido a sus deficiencias y el uso expansivo de la red móvil, llevándolo a gastos innecesarios. Sin embargo, la telefonía es un elemento de suma importancia, sobre todo en el sector empresarial. El presente trabajo de investigación plantea diseñar e implementar una central telefónica utilizando sistemas *VoIP*, que permiten la transmisión de voz y video a través de redes de datos locales o globales. La telefonía IP enlista múltiples beneficios si de comunicación se trata. La implementación de una solución

VoIP lleva consigo sobre todo beneficios económicos y técnicos, pues estos sistemas se enfocan en reducir costos y mejorar la calidad de servicio, que en consecuencia aumenta la productividad de los que la implementan, sacando el máximo provecho de los sistemas *VoIP*.

1.3.1. Justificación técnica

Uno de los indicadores que se tiene fuertemente en telefonía es la calidad de voz ofrecida. Se podría etiquetar a este como el pilar fundamental para desarrollar sistemas de comunicación por voz. Tema que obviamente se tienen en cuenta para los sistemas *VoIP*. Estas tecnologías aplican señal digital, que, frente a la analógica, tienen numerosas ventajas. Por otra parte, al transportar la información (voz y video) por redes de datos, permite gestionar el protocolo adecuado para realizar el sistema de comunicación. Por supuesto, los *codecs* ideales para el sistema también se incluyen, además del nivel de seguridad que se puede aplicar. El diseño de la central telefónica abarca estos puntos clave. El hardware utilizado, (*Raspberry Pi*) cumple con los requisitos necesarios para su implementación. Al trabajar con *software* libres, se puede gestionar de acuerdo a la aplicación que se quiera diseñar y obtener los mejores resultados.

1.3.2. Justificación económica

Respecto a los costos en los servicios de telefonía, la utilización de los sistemas *VoIP* resulta en una reducción de costos frente a los sistemas convencionales. Muchas empresas vienen optando por implementar la telefonía IP en sus sedes administrativas y operativas, debido a la rápida adaptación que tiene al integrarse con la red de datos existente y el ahorro de costos que se genera. Tanto en infraestructura cableada y tarifas telefónicas, pues *VoIP* hace uso de las redes de datos interna para el transporte de los paquetes de video y voz, sea red alámbrica o inalámbrica. Esto evita cableado innecesario, instalación y soporte técnico como lo requiere la telefonía tradicional. Asimismo, la Telefonía IP le añade mayor flexibilidad y nuevas formas de comunicación. Por otro lado, para

este proyecto, se hace uso de *software* libre base como *Asterisk*, y hardware de bajo costo como *Raspberry Pi*. Esto implica que tanto pequeñas o medianas empresas y hogares pueden implementar telefonía IP sin necesidad de realizar gastos realmente grandes y que con el tiempo puedan afrontar los costos que se presenten, debido a la versatilidad de la central telefónica *VoIP*.

1.3.3. Justificación social

La comunicación es un elemento de vital importancia para el ser humano y las organizaciones, por lo tanto, mejorar sus sistemas de comunicación es de gran beneficio. El proyecto que implica el diseño de una central telefónica de bajo costo, ofrece métodos eficaces de comunicación, y aumenta en gran manera de que la telefonía IP se expanda de modo natural y flexible, ya que el sistema es accesible y no requiere gastos inmensurables. Al mejorar los sistemas de comunicación, se genera un desarrollo social que trae beneficios colectivamente. Aportando a que las TIC (Tecnologías de Comunicación e Información) progresen y se integren aun en más en la sociedad, notando la importancia de conocer su funcionamiento y como puede beneficiar a todos, sacando el máximo provecho de ello.

1.4. ALCANCES Y LIMITACIONES DE LA INVESTIGACIÓN

1.4.1. Alcances

- La central telefónica *VoIP* ofrece un sistema de comunicación por voz a bajo costo. Con tarifas telefónicas que el usuario mismo define. La central implementa servicio de llamadas de voz y video entre usuarios a nivel local y global, presenta bandejas de mensajería y buzón de voz. Se puede administrar en manera adecuada y libre, para desarrollar la central a las necesidades requeridas. Teniendo en cuenta esto, el fin del proyecto es presentar una solución frente a las deficiencias de la telefonía tradicional (técnicas o económicas) y a los costosos sistemas *VoIP* que empresas pequeñas y hogares no puedan adquirir. El

objetivo del trabajo de investigación es diseñar e implementar una central telefónica *VoIP* de bajo costo, que ofrezca los mismos servicios de telefonía, con mejoras frente a la calidad de la voz y el costo por llamadas. Además de realizar un análisis de rendimiento y eficiencia en base al trabajo realizado por el servidor telefónico, que, en este caso, es la *Raspberry Pi*.

- A futuro, se planea investigar y desarrollar análisis profundos en base a centrales telefónicas, que involucra el desarrollo de protocolos de comunicación y *codecs* para los sistemas de comunicación de voz y video. Además de desarrollar aplicaciones sencillas a nivel usuario y a nivel administrador.

1.4.2. Limitaciones

- Existen limitaciones en la disponibilidad de información profunda del hardware y *software* de los sistemas de Voz sobre IP, protocolos que se manejan y emplean, etc. También, en relación a su incorporación sobre sistemas operativos Linux, funcionalidad a bajo nivel e interacción de *Asterisk* que presenta su propio lenguaje. Además, claro de tener conocimiento en manejo de lenguajes de programación e instalación de *software* por parte de la *Raspberry Pi*.
- No contar con un ambiente especializado y suficiente para desarrollos en Telefonía IP se convierte en una dificultad, pues al implementar el hardware de la central telefónica y luego realizar las mediciones del rendimiento del servidor. Es necesario obtener análisis de rendimiento y eficiencia para desarrollar o solucionar defectos innecesarios. Para ello, contar con *software* especializado y gran cantidad de teléfonos IP es ideal para realizar pruebas de estrés confiables para obtener buenos resultados.

CAPÍTULO II: FUNDAMENTO TEÓRICO

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1. Internacionales

Para Bastidas (Bastidas, 2016), la expansión del servicio de internet ha generado increíbles cambios en la vida de cada ser humano, pues se ha incrustado tanto, que tener internet se ha vuelto indispensable. Sin embargo, a pesar del exorbitante uso que se le da, la telefonía tradicional aún prevalece en los hogares y algunas empresas. Si bien el costo en sus inicios era reducido, la gestión, el soporte técnico y el mantenimiento del servicio, generan egresos económicos que a largo plazo, familias o pequeñas empresas no puedan manejar. Por ende, el autor en su trabajo plantea como objetivo, diseñar y el desarrollo un prototipo de central telefónica que permita gestionar la comunicación telefónica doméstica.

La comunicación es vital, generar nuevos sistemas que mejoren su calidad de servicio y que sea accesible para todos los usuarios, son grandes aportes hacia la sociedad misma. Los sistemas *VoIP* presentan múltiples ventajas, desarrollar proyectos aplicando esta tecnología, traerá grandes beneficios.

Estévez (Estévez, 2013), en su tesis de *Diseño e implementación de un prototipo de central telefónica con aplicación domótica de bajo costo*, plantea como objetivo diseñar y desarrollar un prototipo de central telefónica, que le permita al usuario un control remoto de dispositivos en su hogar, implementado un sistema domótico, cuyo acceso se da mediante una llamada telefónica y el control por medio de aplicaciones web y móviles.

Para el autor, implementar sistemas inteligentes en los hogares, facilita la vida diaria de los usuarios. Puesto que se pueden integrar sistemas de seguridad, controles a distancia de sus equipos, etc. Todo a través de una llamada telefónica, cuyo nivel de vulnerabilidad es mínimo.

Del resumen de Rivera y Poma (Rivera y Poma, 2014), se extrae que su trabajo tuvo como objetivo, diseñar e implementar un laboratorio

de comunicaciones unificadas de telefonía IP, con el fin de poder realizar análisis, estudios y desarrollar prácticas de laboratorio para profundizar el conocimiento de esta tecnología *VoIP*.

El autor pretende con su proyecto favorecer a los estudiantes de la carrera de telecomunicaciones, fortaleciendo esta rama de la telefonía IP, elevar el conocimiento y su aprendizaje en esta materia, es muy beneficioso, tanto para la institución como para los estudiantes. Los sistemas *VoIP* presentan grandes ventajas, y el poder manejar y desarrollar aplicaciones con esta tecnología, eleva los niveles de calidad en comunicación, aportando bienes hacia la sociedad que vive en telecomunicaciones.

2.1.2. Nacionales

Del resumen de Bardales y Flores (Bardales y Flores, 2015), se extrae que la tesis con título “Diseño y simulación de una red de voz sobre IP para la reducción de costos derivados de las llamadas telefónicas”, tuvo como objetivo el diseño de un sistema de comunicación por voz IP, es decir, una red de telefonía IP, mejorando la calidad de servicio e integrar la interoperabilidad de la red telefónica tradicional, haciendo uso de la red de datos, para minimizar gastos en telefonía.

En conclusión, los autores con su tesis plantean reducir los costos por telefonía, diseñando un sistema *VoIP* que permita la comunicación entre usuarios conectados a una red de datos local. Ofreciendo servicios de voz y video e integrando la red telefónica pública existente, adaptándola para unificar en una sola red, tanto el antiguo, como el nuevo servicio de telefonía. Esto genera grandes beneficios, pues se reducen costos y se mejora la calidad de servicio. Para Llontop Gianmarco (Llontop, 2015), la comunicación es de vital importancia en una empresa, sea grande o pequeña. Contar con un sistema de comunicación por voz, ahorra tiempo y dinero. Por ello, mejorar este servicio eleva la productividad de sus empleados. Sin embargo, para pequeñas o medianas empresas, un sistema *VoIP* no siempre es accesible. De ese modo, el autor en su trabajo de

investigación, plantea diseñar e implementar un servidor *VoIP* con *Asterisk* y *Raspberry Pi*, para unificar los pisos de una oficina mediante los servicios de voz sobre IP, siendo estos dos elementos ideales para el trabajo, pues *Asterisk* es un potente *software* libre para comunicaciones en telefonía IP, y la *Raspberry Pi* es una pequeña computadora de bajo costo, que trabaja con sistemas operativos Linux, siendo adecuado para el proyecto.

Sobre el trabajo de Guerra (Guerra, 2013), plantea diseñar un sistema de control domótico y video vigilancia, por medio de una red IP, siendo supervisado desde un teléfono móvil. Esta propuesta surge debido al problema de la inseguridad y la delincuencia en aumento. Además, recalca en su trabajo, los altos costos por implementar un sistema de vigilancia automatizado, lo cual no es muy accesible para todos los ciudadanos. Por ende, el autor decide diseñar un sistema inteligente que proporcione seguridad supervisada por el mismo usuario. Para ello hace uso de tecnologías *VoIP* y hardware de bajo costo. Aprovecha las ventajas que brinda los servicios de Voz sobre IP para controlar dispositivos u observar en tiempo real, lo que suceda en su hogar, tema que es beneficioso para ciudadanos que desean implementar sistema de seguridad confiables a bajo costo.

2.2. MARCO TEORICO

2.2.1. Red Telefónica Publica Conmutada (PSTN)

La PSTN (*Public Switched Telephone Network*) es una red de conmutación de circuitos tradicional optimizada para comunicaciones de voz en tiempo real. Como se observa en la figura 1, los terminales (teléfonos) se comunican con una central de conmutación por medio de un solo canal compartido por la señal del micrófono y del auricular. Cuando se realiza una llamada, un conmutador se cierra al marcar, estableciendo un circuito con dicho receptor (ElastixTech, 2019).

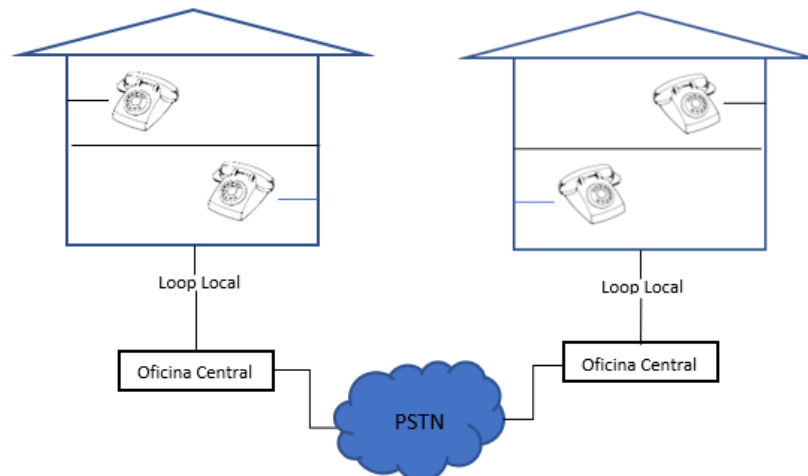


Figura 1: Esquema de una Red Telefónica Tradicional

Para las comunicaciones a distancias, las llamadas pasan de troncal a troncal, llegando a un nodo central, donde este enlaza un usuario con otro. Algunas características principales de la PSTN son las siguientes:

- Banda base de 4KHz para conversaciones entre dos usuarios.
- Capacidad de interconexión con las redes móviles.
- Única red con cobertura y capilaridad nacional.
- El costo por usuario depende del servicio, la distancia de extremo a extremo y la duración de la llamada.
- Consta de medios de transmisión y centrales de conmutación (troncales y líneas de acceso a la red).

2.2.2. Voz sobre protocolo de Internet (VoIP) y Telefonía IP

VoIP (Voice over Internet Protocol) hace posible que la señal de voz viaje a través protocolos de Internet. Divide el audio y video en paquetes IP, codificando la señal (convierte la señal analógica a señal digital) y reensamblando estos paquetes al receptor final por medio de protocolos de señalización, logrando la comunicación entre usuarios (Llontop, 2015). Por ello, es claro que, debido a su naturaleza, sea potencialmente integrable a sistemas informáticos y de tratamiento de datos.

Las comunicaciones de *VoIP*, facilitan servicios y procesos que para los sistemas de telefonía tradicional son muy complejos y costosos de implementar. Una de estas razones, es que *VoIP* puede ejecutarse desde computadoras, teléfonos IP o teléfonos convencionales (*Gateway*) y desde *smartphones* o teléfonos inteligentes (Bardales y Flores, 2015).

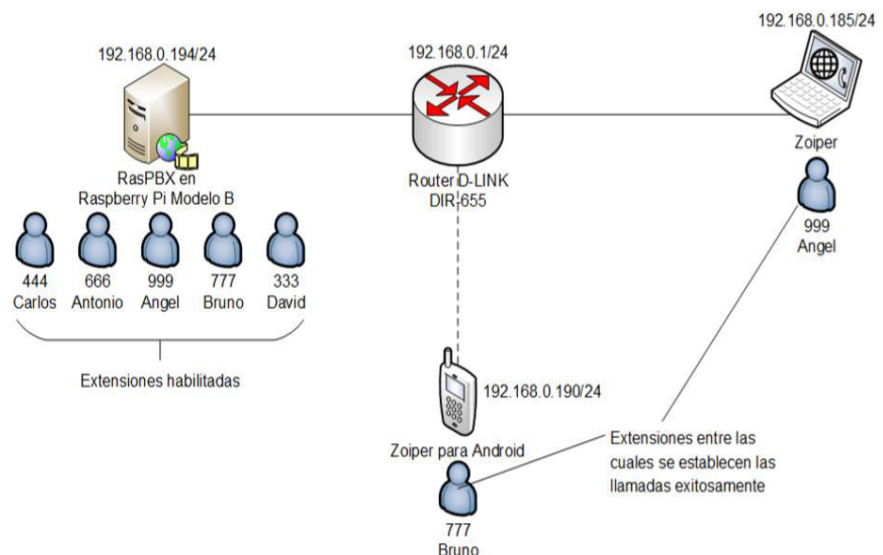


Figura 2: Escenario básico de una red de telefonía IP

Fuente: (Russoniello, Gil y Samaniego, 2015)

La comunicación entre usuarios es mediante teléfonos inteligentes, ordenadores, etc. Todos conectados a una red local, mediante sistemas *VoIP*

La Telefonía IP, básicamente es una aplicación directa de *VoIP*. Este término es utilizado para explicar la convergencia de *VoIP* con servicios como fax y multimedia sobre protocolos de internet (Murthy, 2010); por ello, permite la transmisión de voz por medio del protocolo TCP/IP, enviando la información mediante paquetes comprimidos a través de la red de datos. Sin embargo, el servicio de Telefonía IP añade a la *VoIP* prestaciones adicionales como la calidad de voz y las características avanzadas de comunicaciones (interoperabilidad con la red pública de telefonía, conectividad multi-proveedor, *hunting*, etc.).

Las ventajas que ofrece son:

- Reducción de costos: Transmisión de voz y datos mediante una sola plataforma y red local, esto implica menor inversión en implementación de diferentes servicios para cada aplicación, tanto en mantenimiento como administración.
- Movilidad: Los *softphones* son aplicaciones de telefonía IP (virtuales) para ordenadores y móviles. Si un trabajador necesita trasladar su extensión, no requiere nueva instalación o cambios drásticos de red. En teléfonos inteligentes, actúan de la misma forma que la red móvil, solo basta con estar conectada a una red local de internet, y podrá trasladarse a cualquier parte.
- Tecnología escalable: Sus aplicaciones y tecnologías de la telefonía IP, ofrecen arquitecturas flexibles y escalables, es decir, mejoran cada día, se expanden a nivel mundial. Esto genera desarrollo y crecimiento en el sector empresarial, puesto las ventajas que ofrece la telefonía IP.

2.2.3. Protocolos

En una transmisión, los datos mismos no conocen o no son capaces de hallar su propio camino de comunicación hacia su destino. Para ello, se hacen uso de diversos protocolos para el envío de paquetes de datos. En este caso, tres grupos de protocolos: protocolo de internet o IP, de transporte y señalización.

El *Internet Protocol (IP)* es la base de todo internet. Trabaja a nivel de red (modelo OSI), ofreciendo un servicio “*sin garantías*”. Es decir, no garantiza que los paquetes lleguen, pero tratara de enviar con todo lo posible para hacerlos llegar a su destino (Wikibooks, 2015). Con los protocolos de señalización, su misión es permitir el intercambio de información entre dos puntos o usuarios en la red, con el fin de establecer las llamadas y terminarlas sin ningún problema. Entre los más conocidos tenemos SIP (protocolo de inicio de sesión), H.323 (definido por la ITU-T) y el protocolo original para comunicaciones con *Asterisk*, IAX. A continuación, En la Tabla 1 se muestra los diversos protocolos relativos a la Telefonía IP que son

posibles de integrar en diversos escenarios. Se expone la tabla con muchos de estos protocolos de señalización (Matango, Protocolos de Señalización de la VoIP, 2016).

Tabla 1: Protocolos de Señalización usados en sistemas VoIP

Protocolo	Descripción
H.323	Protocolo definido por la ITU-T
SIP	Protocolo definido por la IETF
Megaco	Conocido también como H.248
SCCP	Skinny Client Control Protocol, propiedad de Cisco
MiNet	Propiedad de Mitel
CorNET-IP	Propiedad de Siemens
IAX	Protocolo original para las comunicaciones entre PBXs <i>Asterisk</i>
Skype	Usado en la aplicación Skype (peer to peer)
Jajah	Protocolo propietario usado en los teléfonos web
IAX 2	Reemplazo de IAX para las PBXs <i>Asterisk</i>
Jingle	Protocolo abierto utilizado en tecnologías Jabber
Telme	Protocolo usado en DeskCall
MGCP	Propiedad de Cisco
weSIP	Protocolo de licencia libre de VozTelecom

La señalización juega un papel importante en *VoIP*, permite establecer, mantener, administrar y finalizar una conversación entre dos usuarios (Universidad de Mendoza, 2015). Para el caso de Telefonía IP, la señalización SIP debe interactuar con los sistemas de señalización SS7 de telefonía fija y móvil.

En parte por los protocolos de transporte, se encargan obviamente de transportar la información (la voz) en tiempo real. Teniendo en cuenta, las técnicas necesarias para evitar problemas durante la transmisión, como retardos o *jitter*, etc. Entre los protocolos más utilizados se muestran en la tabla 2:

Tabla 2: Protocolos de Transporte más usados en sistemas VoIP

Protocolos	Descripción
UDP	Usado generalmente en transmisiones de voz y video
RTP	Protocolo de nivel aplicación para la transmisión de información en tiempo real
RTCP	Protocolo que proporciona información de control para flujos RTP
cRTP	Comprime cabeceras IP/UDP/RTP en enlaces de baja velocidad
SRTP	Seguridad para los datos RTP en aplicaciones unicast y multicast

Dado que la comunicación es tiempo real, UDP es el transporte adecuado para comunicaciones *VoIP*, a diferencia de TCP, que normalmente a nivel de internet global, es más confiable. Esto se debe a que TCP con su control de flujo y congestión, ocasionarían retardos durante las transmisiones (Llontop, 2015).

2.2.4. Codecs

La Telefonía IP es de naturaleza digital, por ende, el primer paso es la conversión de la señal análoga (la voz) a una señal digital mediante el módulo de conversión A/D. Posterior a ello continúa el proceso de codificación digital, cuya acción es reducir el ancho de banda requerido por la señal original en su transporte digital, manteniendo un nivel de calidad de señal determinado. Sin embargo, existen diversos tipos de *codecs* o codificadores, que pueden afectar directamente la calidad de la señal (Russoniello, Gil y Samaniego, 2015). Esto se debe a que el codificador comprime la secuencia de datos y proporciona la cancelación del eco para transmitirlo y luego convertirlo nuevamente, descomprimiendo la señal de voz o video para la comunicación en tiempo real. Siendo sumamente importante la elección del *codec* para la calidad de servicio (CISCO, 2016).

Cada *codec* ofrece características propias, las más relevantes entre estas son:

- Frecuencia de muestreo: Número de muestras que son tomadas en un segundo (normalmente este valor es de 8KHz).

- Tasa de compresión nativa: El valor de la velocidad binaria de la señal en la salida del *codec* (valor típico de 64 Kbps).
- Tamaño de la Trama y *Payload*: Longitud de los datos digitalizados de cada duración de la trama, expresado en milisegundos o en Bytes. El *Payload* es la zona de carga que transporta los datos enviados por los paquetes de protocolos superiores.
- Tasa de compresión del *codec*: Velocidad binaria que el algoritmo de compresión del *codec* es capaz de realizar.

Mientras que en *VoIP* existen diversos codificadores con diversas calidades de señal, para Telefonía IP éstos deben brindar una calidad telefónica que la UIT recomienda en la serie *G.XXX*. El desempeño y la calidad dependerá del algoritmo utilizado. La tabla 3 enlista algunos de estos *codecs*:

Tabla 3: Codificadores para *VoIP*

Codecs	Descripción
G.711	<i>Codec</i> principal de la PSTN estandarizado por la ITU
G.726	También conocido como ADPCM, sustituyo al G.721
G723.1	Opera 6.3 o 5.3 Kbps
G.729A	Requiere un ancho de banda de 8Kbps
GSM	Sistema global para comunicaciones móviles
iLBC	<i>Codec</i> Bajo del Índice Binario del Internet
Speex	<i>Codec open source</i> que busca reducir la barrera de la conversión hacia tecnologías <i>VoIP</i>

2.2.5. Calidad de Servicio

En telecomunicaciones, la provisión de los servicios públicos debe asegurar el indicador de calidad del mismo sobre un nivel regulado, el cual debe presentarse a nivel usuario y dentro de la infraestructura del proveedor. Así, la calidad de servicio, *QoS (Quality of Service (QoS))*, hace referencia a la capacidad que tiene un sistema de comunicaciones para asegurar que se cumplen los requisitos de

tráfico para un flujo de información determinado (Matango, ServerVoIP, 2016).

En *VoIP*, las comunicaciones se dan en tiempo real, por lo tanto, se debe asegurar que la voz (paquetes IP) llegue al receptor en el menor tiempo, en lo posible jamás lleguen tarde o nunca (Villa, 2007). Los mecanismos de garantía de *QoS* tratan de asegurar que esto no suceda, idealmente. El objetivo es proveer soluciones frente a los problemas que se han venido presentando en las redes IP o los sistemas *VoIP* (generalizando, en los sistemas de telecomunicaciones o TIC) ante el uso masivo de usuarios. La expansión exorbitante de aplicaciones multimedia, que incluyen servicios de voz y video entre otros, han generado la necesidad de establecer un nivel de *QoS* para el buen manejo y funcionamiento de las tecnologías y aplicaciones.

En cuanto a los factores para calidad de servicio de *VoIP*, se tiene: *Delay* o retardo, *jitter* (variación del retardo), caudal, pérdida de paquetes y el eco.

2.2.6. Asterisk

De forma resumida, *Asterisk* es un *framework* de código abierto para el desarrollo de aplicaciones de comunicaciones. *Asterisk* sobre un sistema de cómputo es un servidor de comunicaciones, potenciando sistemas IP, PBX, *VoIP*, servidores de conferencias, etc. (*Asterisk*, 2019).

Comenzando en 1999 cuando Mark Spencer libero el código origen bajo licencia GPL. A partir de ahí, mejoras y aplicaciones se realizan por la comunidad mundial. *Asterisk* es mantenido por Digium y los esfuerzos de la comunidad. Está escrito en C bajo Linux (y otros sistemas Unix), *Asterisk* permite el desarrollo de aplicaciones (multi-protocolo) de comunicación en tiempo real, soportando *VoIP* (principalmente SIP), PSTN y POTS (Gómez y Gil, 2008).

La arquitectura de *Asterisk* consta de principalmente 4 elementos:

- Núcleo PBX, que es el componente esencial que provee gran parte de la infraestructura.

- Módulos, que proporcionan funciones adicionales
- Llamadas y canales, en el caso de *Asterisk*, se refieren a uno más canales existentes.
- Dialplan, que es el único y principal método para controlar el comportamiento de *Asterisk*. Se aloja como archivos de texto en lenguaje AEL o LUA.

La figura 3 muestra la arquitectura de *Asterisk*

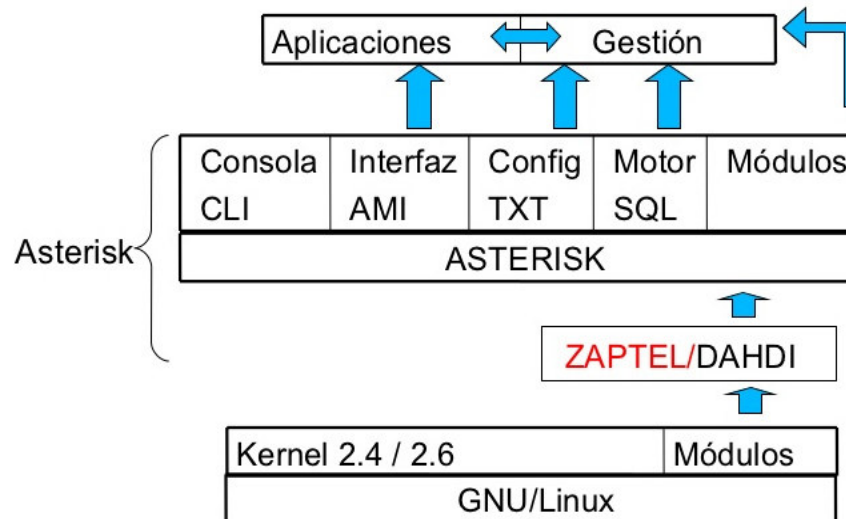


Figura 3: Arquitectura de *Asterisk*

Fuente: (Astudillo, 2009)

2.2.7. *FreePBX*

FreePBX es una GUI (interfaz gráfica de usuario) que administra y controla *Asterisk*, creada bajo licencia GPL. La flexibilidad de *FreePBX* proviene de las docenas de complementos y funciones que permite organizar una interfaz de gestión, de acuerdo a las necesidades que se tiene (Sangoma, 2016).

De la misma forma que *Asterisk*, la comunidad de desarrolladores de *FreePBX* dedican sus aportes para crear una interfaz fácil de usar para softwares de telefonía.

2.2.8. *Raspberry Pi 3*

Las *Raspberry Pi* son una serie de “microcomputadoras de una sola placa”, desarrollados por *Raspberry Pi Foundation*, originalmente creados para promover la enseñanza de la computación en las

escuelas. Actualmente se han lanzado 4 modelos de placa, cada una superior a la anterior. Para el proyecto, se ha emplea la penúltima versión, la *Raspberry Pi 3 B+*, con mejoras a sus antecesores. Este modelo incluye:

Tabla 4: Especificaciones técnicas de la *Raspberry Pi 3*

<i>Especificaciones</i>	<i>RASPBERRY PI 3 MODEL B+</i>
PROCESADOR	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
FRECUENCIA DE RELOJ	1,4 GHz
MEMORIA	1GB LPDDR2 SDRAM
CONECTIVIDAD INALÁMBRICA	2.4GHz / 5GHz IEEE 802.11.b/g/n/ac Bluetooth 4.2, BLE
CONECTIVIDAD DE RED	Gigabit Ethernet over USB 2.0 (300 Mbps de máximo teórico)
PUERTOS	GPIO 40 pines HDMI 4 x USB 2.0 CSI (cámara <i>Raspberry Pi</i>) DSI (pantalla tácil) Toma auriculares / vídeo compuesto Micro SD Micro USB (alimentación) Power-over-Ethernet (PoE)

Evaluar el rendimiento de las *Raspberry Pi* es un tanto más complejo que hacerlo en un móvil, PC o portátil con sistemas operativos de escritorio o móviles tradicionales: el catálogo *software* de esos sistemas suele incluir unas cuantas herramientas para la ejecución de pruebas sintéticas que como siempre son una buena referencia para poder determinar niveles de rendimiento para cada dispositivo. (Xataka, 2018)

Las *Raspberry Pi* se manejan por medio de su propio sistema operativo basado en *Debian GNU/Linux*, este es el *Raspberry Pi OS* (anteriormente *Raspbian*), lo cual se descarga desde su propia página web, además para ello la *Raspberry Pi* necesariamente requiere de una tarjeta de memoria micro SD para instalarse su

sistema operativo, mínimo de 8 Gb de espacio. La figura 4 muestra el escritorio estándar del sistema.

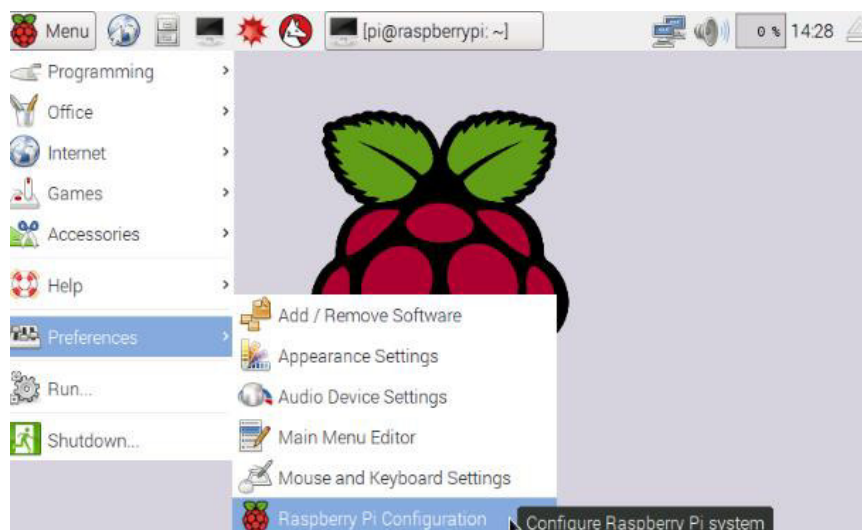


Figura 4: Escritorio de la *Raspberry Pi*

Los sistemas operativos para la *Raspberry Pi* son muy similares a los de GNU/Linux.

2.3. MARCO METODOLÓGICO

2.3.1. Tipo de investigación

Teniendo en cuenta el tipo y desarrollo del proyecto, el presente trabajo es de investigación aplicada tecnológica, ya que se emplea el desarrollo de la ciencia y la tecnología para resolver un problema (los costos excesivos por telefonía), tanto social (las comunicaciones y poca disponibilidad de información especializada en sistemas *VoIP*) y empresarial (reducción de gastos en telefonía).

Según la investigación, se divide en:

- Investigación orientada a conclusiones. Engloba la metodología cuantitativa.
- Investigación orientada a decisiones. No se centra en hacer aportes teóricos, sino en buscar soluciones a los problemas. La investigación-acción forma parte y se vale de algunas metodologías cualitativas. (Pérez, 2010)

Para el desarrollo del trabajo de investigación, se utiliza documentación oficial del hardware y software. De este modo, se

asegura que la información empleada en el diseño e implementación del proyecto sea confiable y actualizada.

2.3.2. Metodología de la investigación

De manera sintetizada, se resume la metodología desarrollada en tres etapas principales: Diseño de la Central Telefónica *VoIP*, Implementación de la Central Telefónica *VoIP*, y en último Análisis de Pruebas y Rendimiento, como se muestra en la figura 5.



Figura 5: Diagrama de bloques simplificado

A. Diseño de la central telefónica *VoIP*

Para el desarrollo de esta primera fase, se resaltan 3 puntos clave:

1. Identificación y descripción de las capacidades y requerimientos, tanto del hardware (*Raspberry Pi*) y del software (*Asterisk*). Esto permitirá conocer los límites y los alcances que tienen las plataformas utilizadas. De este modo, la elección del hardware y software para la central telefónica *VoIP*, tendrá justificación notable sobre del porque se emplea estas tecnologías y no otras.
2. Análisis teórico de la capacidad de los servicios o aplicaciones que podría ofrecer *Asterisk* al ejecutarse en una *Raspberry Pi*. Tanto este *framework* libre como esta pequeña computadora se basan en sistemas Linux, de tal manera, son compatibles. Sin embargo, aquello no significa que pueda tener las mismas capacidades de un ordenador promedio con *Asterisk*. El análisis trata sobre la capacidad y comparación de la *Raspberry Pi*, ejecutando *Asterisk* frente a otros sistemas o equipos.
3. Diseño y análisis de redes de datos donde se pueda implementar la central telefónica *VoIP*. Considerando la comunicación en una red local interna o externa.

B. Implementación de la central telefónica *VoIP*

Esto implica esencialmente la implementación de los servicios básicos proporcionados por una central telefónica *VoIP* por medio de *Asterisk* y *Raspberry Pi*. Principalmente los servicios de llamada por voz y video, mensajería y buzón de voz. Para la gestión de la central, el análisis del tráfico de voz o el historial de llamadas (duración y fecha), la creación de nuevos anexos (usuarios) o grupos de timbrado. Además de enlazar distintos servidores o centrales telefónicas (redes troncales).

C. Análisis de pruebas y rendimiento

En este proceso se realizan las pruebas de estrés sobre la central telefónica *VoIP*. Esto implica la implementación de una red que integra teléfonos IP y *softphone* (teléfonos IP virtuales) en ordenadores y teléfonos inteligentes. Con el fin de comparar los análisis teóricos con los análisis experimentales, medir el rendimiento que tiene la *Raspberry Pi* frente a un determinado número de usuarios y sobre todo definir su capacidad como central *VoIP*. Esto servirá para la identificación de servicios adicionales (o creación de aplicaciones) que se puedan incluir en la central telefónica y mejorar la calidad de servicio ofrecida.

2.4. MARCO LEGAL

2.4.1. Recomendaciones Y.1540 y Y.1541

Para regular los sistemas de telecomunicaciones y los niveles de calidad de servicio, organizaciones e instituciones exponen recomendaciones, que toda empresa dedicada a un determinado sector, debe presentar parámetros definidos por estas organizaciones, con el fin gestionar y regular estos sistemas.

La UIT (Unión Internacional de las Telecomunicaciones) en su recomendación Y.1540, para evaluar la calidad de funcionamiento en

la transferencia de paquetes o datos por medio del protocolo de internet IP, define parámetros en cuanto a velocidad, precisión, fiabilidad y disponibilidad (ITU, 2011). Para medir la calidad de servicio, se mencionan 4 principales:

- IPTD o *IP PACKet Transfer Delay*, que es el tiempo que tarda el paquete en llegar a su destino.
- IPLR o *IP PACKet Loss Ratio*, relación entre el total de paquetes perdidos y los que fueron transmitidos.
- IPDV o *IP PACKet Delay Variation*, referencia a la variación en el tiempo de llegada de los paquetes.
- IPER o *IP PACKet Error Ratio*, que se refiere a la tasa de paquetes con errores.

Para la recomendación Y.1541, define a los valores de calidad de funcionamiento para los parámetros anteriores, pero con definición de número de clase de QoS (Olave, Valencia y Cuéllar, 2014)

- Clase 0, definida para aplicaciones de tiempo, como *VoIP* y *VTC* (Video-Tele-Conferencias) con uso de técnicas de enrutamiento y distancias limitadas (ITU, 2011).
- Clase 1, igual que la anterior clase 0, pero con la diferencia de que sus técnicas son menos limitadas (enrutamiento y distancia)
- Clase 2, comprende aplicaciones de datos transaccionales y aplicaciones altamente interactivas. Técnicas de enrutamiento y distancia limitadas
- Clase 3, misma clase 2 pero con técnicas menos limitadas.
- Clase 4, para aplicaciones que soporten pérdidas bajas (transacciones cortas, *videostreaming*, etc.). Hace uso cualquier ruta o trayecto.
- Clase 5, definida para aplicaciones tradicionales de redes IP (ITU, 2011)

La relación de ambas recomendaciones se muestra en la tabla 5.

Tabla 5: Parámetros de QoS en una red

Parámetro	Clases de QoS					
	CLASE 0	CLASE 1	CLASE 2	CLASE 3	CLASE 4	CLASE 5
IPTD	100 ms	400 ms	100 ms	400 ms	1 s	U
IPDV	50 ms	50 ms	U	U	U	U
IPLR	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	U
IPER	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	U

Como se observa en la tabla, ambos elementos (parámetros) están relacionados, y es fundamental tenerlos en cuenta en la provisión de servicios.

2.5. DIAGRAMA DE BLOQUES Y ARQUITECTURA DEL SISTEMA

Para el presente trabajo de investigación, se desarrolla mediante el siguiente diagrama de bloques en la figura 6:



Figura 6: Diagrama de bloques general

El diagrama de bloques resume prácticamente todo el proyecto de investigación

Como se mencionó anteriormente:

- En la parte de Identificación de las capacidades del Hardware, se analiza a la *Raspberry Pi* y todas las funciones que presenta.
- En segundo, la descripción de los servicios ofrecidos por *Asterisk*, este *framework* libre está hecho para centrales telefónicas.
- En análisis de la capacidad de *Asterisk* en una *Raspberry Pi*, se estudia la compatibilidad y los servicios que podría proporcionar.
- Sobre la implementación de la central telefónica *VoIP*, se incluye la creación de los anexos, los servicios que ofrece una central *VoIP*, etc.

- Ejecución de pruebas de estrés, por medio de teléfonos físicos y virtuales.
- En el análisis de rendimiento y resultados, se mide la capacidad de la *Raspberry Pi*, como central *VoIP*.

Para la arquitectura del sistema, se basa en el sistema Usuario/Servidor. En este caso, los usuarios, para comunicarse, marcan el anexo que llega al servidor telefónico que es la *Raspberry Pi*, que a su vez los enlaza a su destino. Se define la arquitectura en la figura 7 del primer escenario:

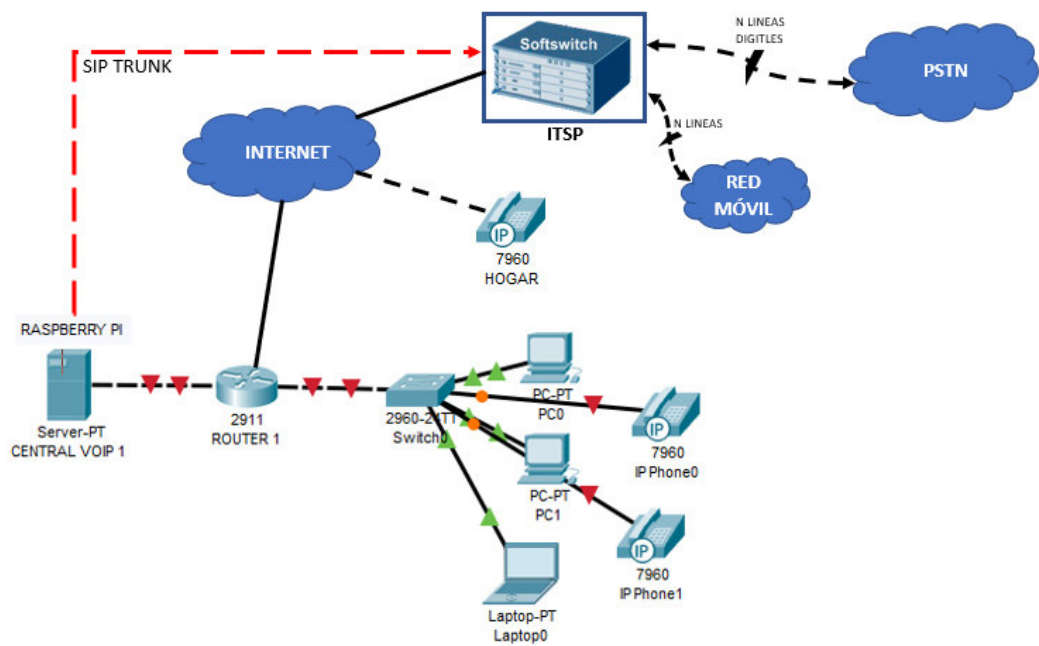


Figura 7: Arquitectura del sistema – Escenario 1

En este primer escenario, se tiene una red local. Conectada por medio del ITSP o proveedor de servicios de telefonía por internet, hacia la red móvil y la red de telefonía tradicional. Está a su vez tiene una troncal SIP (*trunk SIP*) hacia el ITSP para poder conectar a usuarios remotos que se encuentren en redes externas por internet. Además de tener una central *VoIP* alterna, para problemas futuros.

Para un segundo escenario, se define la siguiente arquitectura en la figura 8:

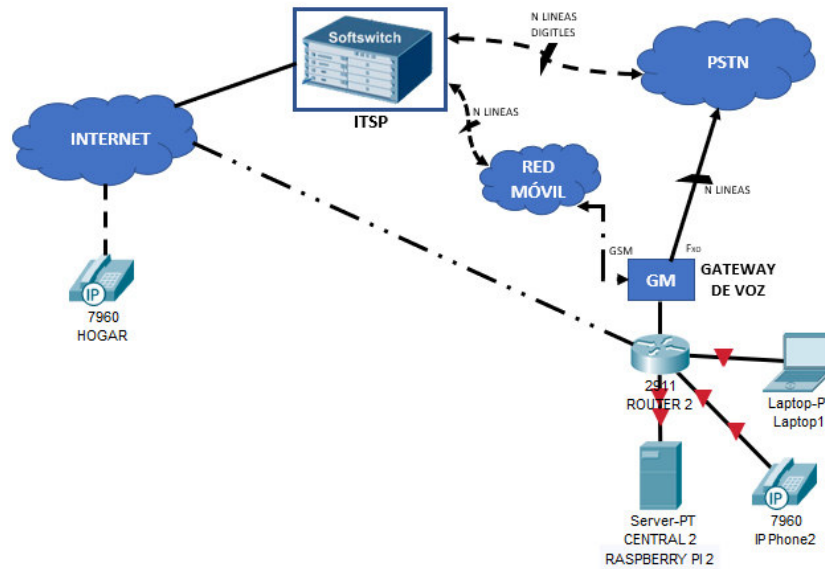


Figura 8: Arquitectura del sistema – Escenario 2

En este entorno de red, al igual que el escenario 1, se tiene otra red local que, mediante un *Gateway*, los usuarios puedan acceder a la red móvil y la red de telefonía fija. A su vez, por medio de una red de internet, algunos usuarios remotos pueden comunicarse desde su hogar, hacia la empresa.

CAPÍTULO III: DESARROLLO DE LA APLICACIÓN

3.1. DESARROLLO DEL OBJETIVO ESPECÍFICO 1: Evaluar la situación actual de los servicios de telefonía tradicional y los nuevos sistemas que han surgido hasta la actualidad.

La telefonía tradicional o fija en el país peruano a lo largo de su desarrollo e integración en los hogares, desde sus inicios hasta la actualidad, ha tenido una historia muy particular. Al recordar las décadas de los noventa, donde adquirir una línea telefónica era extremadamente costosa, y además de que tomaría mucho tiempo (meses o incluso años), al compararse a la actualidad, donde a solo una llamada cualquiera pueda adquirir una variedad de servicios telefónicos, costaría creer el increíble cambio que tuvieron las telecomunicaciones en el Perú (Gestión, 2018).

Actualmente, el servicio de comunicación más extendido a nivel mundial es la telefonía móvil, lo cual ha ido desplazando (no del todo) a la telefonía fija tradicional. Sea analógica o digital, la cero escalabilidad de esta red, y los costosos mantenimientos que conlleva la adquisición de estos servicios, generaron hasta el 2018 (y actualmente) un gran decrecimiento en el uso de minutos de la telefonía fija, como se observa en la figura 9. Inclusive, los teléfonos públicos han ido desapareciendo a nivel nacional (figura 10). Solo las líneas privadas, cuyo mayor uso se le dan en empresas o corporaciones grandes, permanecen intactas (aunque no del todo), debido a la ventaja de velocidad en comunicación (Gestión, 2019).

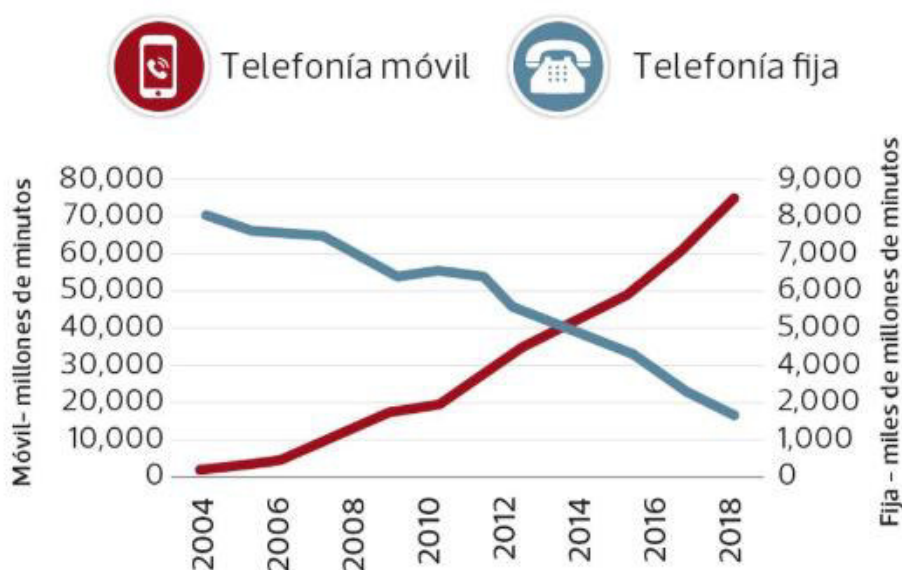


Figura 9: Comportamiento de la telefonía fija y móvil hasta el 2018

Fuente: (Osiptel y Telefónica, 2019)

El uso de la telefonía fija ha tenido un decrecimiento notable a través de los años hasta la actualidad.



Figura 10: Comportamiento del uso de los teléfonos públicos

Fuente: (Osiptel y Telefónica, 2019)

Es notable como el uso de los teléfonos públicos han decaído, hecho que fue ocasionado por comunicaciones móviles.

Tanto la Fig.09 y 10 muestran el claro decrecimiento del uso de la telefonía fija, esto no significa que se haya vuelto totalmente obsoleta. Esta red es muy importante y usada en empresas medias y grandes, sin embargo, la telefonía móvil con sus beneficios en comunicación y los menores costos por la contratación de sus servicios la relevan casi en su totalidad. A pesar de ello, la integración de una red de telefonía, es bien necesaria, sea en empresas u hogares, dada los beneficios que se trae. En consecuencia, nuevas tecnologías surgen para mejorar e integrar a la telefonía y poder ser una buena solución para sus comunicaciones fijas o móviles. Para ello, la telefonía IP surge, que es una aplicación directa de tecnologías *VoIP*.

La telefonía IP trajo consigo dos grandes beneficios, mejoras en la *QoS* y la reducción increíble de costos, tal cual se expuso párrafos atrás. Dada sus grandes ventajas, en Perú, un 25% de las líneas telefónicas ya operan con tecnología *VoIP* y se estima que esta cifra crecerá (Gestión, 2018).

VoIP viene ganando terreno, a nivel nacional y mundial, y es normal puesto la versatilidad y la escalabilidad que tienen estas tecnologías. De modo que es ideal para la realización de este proyecto de investigación.

3.2. DESARROLLO DEL OBJETIVO ESPECÍFICO 2: Determinar los indicadores principales que se debe tener en cuenta para un servicio de telefonía.

3.2.1. QoS (Calidad de Servicio)

Cuando se habla de comunicaciones, de los tres elementos básicos de dicho sistema (emisor, mensaje y receptor), se puede afirmar que el mensaje es el más importante de todos. Una comunicación se origina cuando un individuo quiere transmitir información hacia otra, por ende, se le presta mayor atención a este medio. De nada sirve si el mensaje que se quiere enviar, llega defectuoso, que en consecuencia no se logra una comunicación.

En comunicaciones por voz (servicios de telefonía y video), para garantizar que el mensaje llegue sin problemas, el medio o canal de transmisión debe clasificar y priorizar el tráfico de información cuando haya momentos de congestión. Quien se encarga de medir y lograr todo esto, es la *QoS* o Calidad de Servicio.

En toda red de comunicación convergente moderna, *QoS* debe estar presente, pues asegura que no solo el mensaje llegue a su destino, sino también que sea de buena calidad y se pueda entender lo que el emisor trata de comunicar, con un buen *QoE* (calidad de experiencia) (Salazar, 2016).

En resumen, *QoS* es un conjunto de mecanismos de control de tráfico con el fin de no degradar la experiencia del usuario. Por ello *QoS* abarca factores como la capacidad del medio para transmitir (ancho de banda), retardos, *jitter* (variación del retardo) y pérdida de paquetes cuando ocurra congestiones en la red, y por supuesto la latencia, que es el tiempo que le toma un paquete llegar a su destino.

3.2.2. Costos por telefonía

De nada sirve brindar servicios de telefonía si estos no están accesibles a todos los clientes, ya sea por infraestructura o por ser muy cara. Un claro ejemplo es la antigua situación que sostuvo el Perú a inicios de los 90's cuando empezó a ofrecer telefonía fija, donde solo 3% de la población total constaba con un teléfono en casa (Gestión, 2019). Para poder comunicar una persona con otra, nacional o internacional, la telefonía tradicional, cuyo funcionamiento estaba basado en conmutación por circuitos, tenían que conectar cables a lo largo de todo el recorrido para formar un camino entre ellos. Esto hacía que mientras más grande sea la distancia, mayor era el costo por llamada (Matango, 2016).

Actualmente para telefonía fija, se han reducido los costos por sus servicios, sin embargo, frente a la telefonía móvil, el gasto que requiere mantenerlos es alto, pues se paga mensualmente, sin importar si se ha llegado a usar alguna vez durante el periodo de contrato, a diferencia de la red móvil, donde se paga cada vez que se utiliza.

Los costos son indicadores que van de la mano en cualquier servicio que se brinda, meditar sobre las ventajas y beneficios que se obtiene de acuerdo a la cantidad que se paga por un servicio es de gran importancia. Saber a qué costo, se podrá implementar una red de comunicaciones por voz (telefonía) y a largo plazo como resultaran los gastos, genera desarrollo y crecimiento económicos que toda empresa debe conocer.

La telefonía IP aquí es donde resalta, ya que reduce los costos increíblemente por implementar un sistema de comunicaciones, sea por voz o video. Como se menciona atrás, estas nuevas tecnologías brindan excelentes herramientas y son muy abiertas y escalables, a diferencia de las redes telefónicas tradicionales. Prácticamente con telefonía IP, se paga mucho menos por el mismo servicio, pero aún mejor y desarrollada, tanto económica como técnica.

3.3. DESARROLLO DEL OBJETIVO ESPECÍFICO 3: Diseñar una central telefónica VoIP que permita reducir los gastos de telefonía y mejorar la calidad de comunicación por voz

Definido una vez que se va utilizar un sistema VoIP para implementar el sistema de comunicaciones. Lo siguiente es detallar (o en todo caso diseñar) la topología de la red, el hardware y *software* necesarios para el servidor y la red en general.

3.3.1. Diseño de la Red VoIP en General y/o Estándar

Generalmente, toda empresa mantiene en su infraestructura redes telefónicas (fijas y móviles) y de internet. La versatilidad de VoIP permite integrar un sistema de comunicación, en las redes locales (o globales) propias de la empresa sin necesidad de implementar una nueva red para este servicio. De este modo, la topología que toma, puede ir acorde a la infraestructura de la red que la empresa ya mantiene. Además de ello, al ser escalable, el sistema de comunicación, bien puede crear subredes telefónicas (grupos de timbrado) o sistemas de mensajería, voz y video entre sedes que estén a larga distancia. A continuación, se expone la topología física de una red básica en general, que toda empresa mantiene en su infraestructura en conjunto al sistema VoIP integrado.

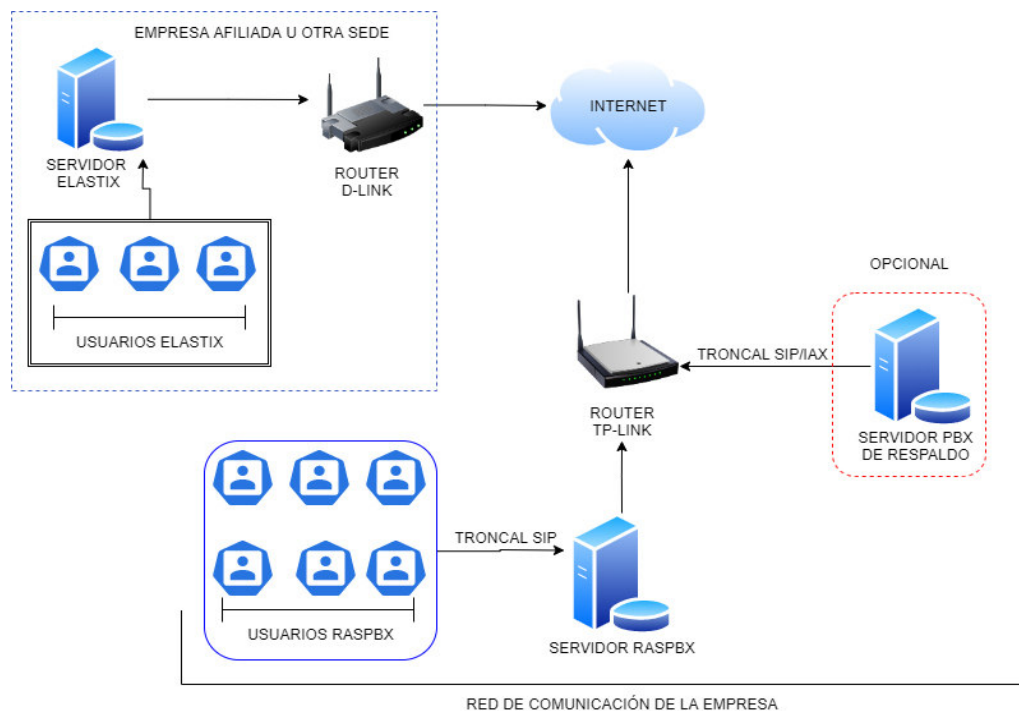


Figura 11: Topología Física de la red estándar de una empresa pequeña

La figura 11 muestra el esquema de una red general básica de comunicaciones. Esta contempla los anexos registrados en el servidor *RasPBX*, una PBX de respaldo y las conexiones a otras sedes con otros servidores. Es importante tener muy cuenta la parte física y lógica de la red, esto permitirá establecer un orden y seguimiento en los usuarios cuando estén haciendo uso de los teléfonos.

Centrándose solo en la red local, el diseño de la topología comienza de la siguiente forma:

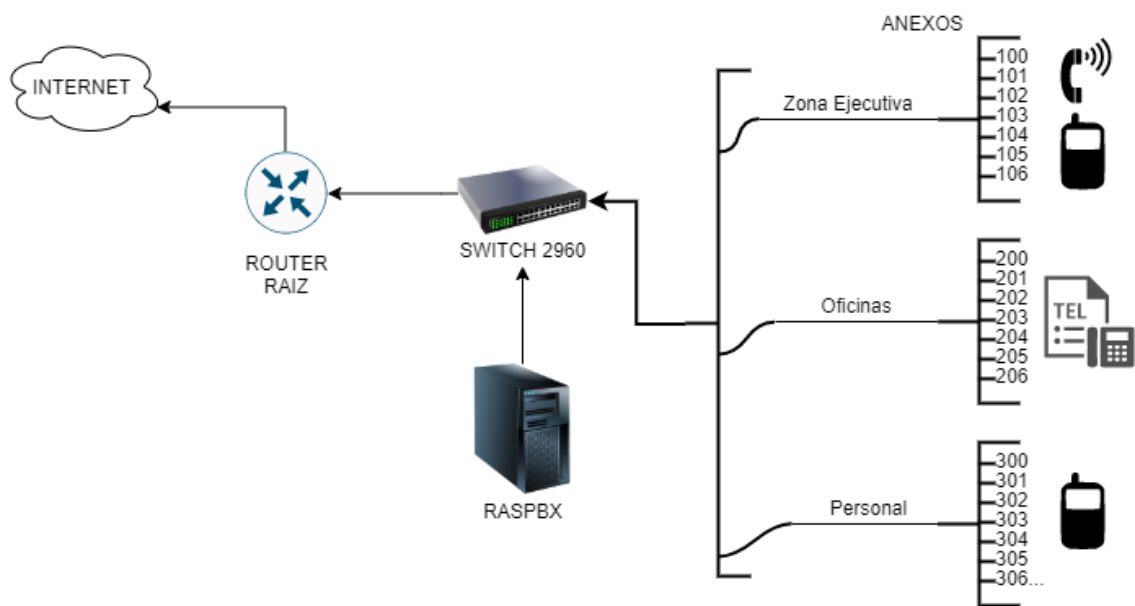


Figura 12: Topología de la red local de comunicaciones

En la figura 12 se muestran tres grupos distintos. Esta la zona de gerencia o ejecutiva que abarcan los anexos del 100 hacia adelante. El grupo de oficinas con los anexos a partir de los 200 y el ultimo que es personal que toman los 300. En todos los grupos, se pueden generar o quitar nuevos anexos y crear nuevos grupos de llamadas.

El servidor telefónico implementado en la *Raspberry Pi* permite la comunicación entre teléfonos IP, *softphones* y mediante *gateways*, teléfonos tradicionales conectados a la PSTN. Capaz de procesar llamadas de voz y video, crear extensiones (sean SIP o IAX) configuraciones para servicios de mensajería o buzón de voz, música

en espera, centro (ACD), registro (CDR), grabación e interrupción de llamadas, etc. Comunicación entre sedes o servidores, temas que serán detallados más adelante.

3.3.2. Instalación y Configuración de *RasPBX*

El hardware de la central *VoIP* es la *Raspberry Pi* (3b+ en este caso). Este tipo de minicomputadoras trabajan con *software* basados en Linux (*Debian GNU/Linux*) y son muy potentes a pesar del tamaño. Asimismo, *Asterisk* que es un *software* especializado para centrales telefónicas IP, puede ser instalado en cualquier plataforma que contenga el *kernel* de Linux. En modo que la *Raspberry Pi* puede abordar *Asterisk*. Para convertir una *Raspberry* en una centralita *VoIP*, actualmente existen dos métodos: Instalación por medio de *Raspbian* o instalación de *RasPBX*.

Raspbian (actualmente *RaspberryOS*) es el sistema operativo principal de la *Raspberry*, para la instalación de *Asterisk*, los métodos son iguales que en una distribución Linux. Actualizando y extrayendo los repositorios, como si se instalara Elastix (otro *software* que contiene *Asterisk*) en una CentOS (otra distribución de Linux). Se instalan de manera manual *Asterisk* y *FreePBX*.

En cuanto a *RasPBX*, es una distribución para la *Raspberry* dedicada especialmente para *VoIP*, pues solo contiene a *Asterisk* y *FreePBX*. Es una plataforma que no consta de una GUI (interfaz gráfica) y las configuraciones se dan por líneas de comando (CLI).

Para el proyecto, se utiliza el segundo método por dos obvias razones: Es una plataforma echa como servidor *VoIP* y el rendimiento como central telefónica es mucho más alto que la *Raspbian*. La *Raspbian* contiene una GUI (además de los muchos otros procesos que realiza en segundo plano), esto disminuye la capacidad de la *Raspberry* como centralita.

La instalación de *RasPBX* es igual que la *Raspbian*, se quema la imagen del sistema en una tarjeta micro SD (véase el anexo 1)

Antes de realizar cualquier tipo de cambios, se le daba dar una preconfiguración como actualizar el sistema (sudo apt-get update), la zona horaria, cambiar el lenguaje, etc. (véase anexo 2)

3.3.3. Configuración de Asterisk

Actualmente para poder configurar *Asterisk*, se han creado ciertas GUIs amigables que cualquier usuario sin conocimiento sobre *VoIP* puede gestionar. Sin embargo, es muy importante conocer los archivos base que contiene *Asterisk* y configurar sus funcionalidades. Esto ayuda cuando se quiera desarrollar interfaces gráficas propias o generar nuevas funciones para la central telefónica.

En especial hay dos archivos importantes a configurar: ***sip.conf***, donde se configuran a los clientes SIP y ***extensions.conf***, que se desarrolla el plan de marcado, encargado de gestionar las llamadas y todas las demás funciones y privilegios que dispones los usuarios. Para un escenario sencillo de *VoIP*, basta con realizar mínimos cambios en estos ficheros. Sin embargo, para los propósitos del proyecto, son necesarios la modificación de otros archivos de configuración. Dentro de estos se encuentran la gestión de las extensiones, como SIP o IAX, los sistemas de buzón de voz, mensajería, dialplan, etc.

A. *SIP.CONF*

Este archivo permite configurar el canal para el protocolo SIP y la creación de las extensiones (usuarios). Lo conforman dos bloques: el primer bloque **[general]** que establece los parámetros predeterminados para las extensiones, como el puerto de comunicación, los *codecs* permitidos, la IP del servidor, etc. Y el segundo bloque **[extensión]** donde se crean y asignan a los nuevos usuarios para la central. Cabe mencionar que para esta ocasión dentro de los corchetes **[---]** se escribe el nombre o número con el que se desea identificar el usuario (ejem. [7001]).

Además, se describen los datos únicos para este cliente, como contraseña, nombre, etc.

En la Tabla 6 se muestra los valores que normalmente se configuran de manera básica.

Tabla 6: Configuración [general] SIP

[general]	
Función	Descripción
updbindaddr=0.0.0.0:5060	Define la interfaz de red y el puerto UDP. En este caso el "0.0.0.0" es para escuchar a todas las interfaces y el 5060 es el puerto UDP usado las extensiones SIP
allowguest=no	Define a que las llamadas de extensiones SIP remotas sin contraseña son rechazadas. Parámetro de seguridad
srvloOKup=yes	Realiza búsquedas DNS para el envío de correos, debe estar activo cuando se usa proveedores de afuera o externos
alwaysauthreject=yes	Rechaza usuarios al intentar autenticarse con información o credenciales erróneas
language=sp	Se define el idioma

Para la configuración de una extensión se procede como se muestra en la Tabla 7:

Tabla 7: Configuración de [extensión] SIP

[extension]	
Función	Descripción
username=200	Nombre de usuario para la extensión
secret=1234	Clave del usuario
callerid=200	Identificador de llamada, muestra el nombre y la extensión si este no está agendado
mailbox=100@default	Buzón de correo definido en el voicemail.
type=friend	Define el tipo interacción entre el usuario y el sistema (como <i>user</i> y <i>peer</i>)
nat=no	No permite la traducción de direcciones IP, en ocasiones SIP tiene conflictos con el uso NAT por lo que se desactiva

host=dynamic	Define el host en el que el usuario se comunica con el servidor. El modo <i>Dynamic</i> permite al usuario conectarse desde varias direcciones IP
context=internal-extensions	Contexto del dialplan
canreINVITE=no	Define si el audio de la llamada pase directamente a <i>Asterisk</i> o a los terminales. Con "No" se evita los problemas de tener el audio en una dirección o no se obtenga nada forzando al streaming pasar directamente a <i>Asterisk</i>
Qualify=yes	Determina cuando el dispositivo será alcanzado. Por defecto <i>Asterisk</i> considera presente al dispositivo al tener latencias menores a 2000 ms, por lo que poner <i>yes</i> o <i>2000ms</i> es igual.
Disallow=all	Desactiva todos los <i>codecs</i> de audio
Allow=alaw	Habilita un <i>codec</i> en específico (se pueden habilitar varios <i>codecs</i> para un mismo usuario)

B. IAX.CONF

Este fichero permite configurar el canal las extensiones iax (IAX2 en este caso), la configuración es similar al fichero *sip.conf* mostrado anteriormente como se muestra en la tabla 8.

Tabla 8: Configuración [general] IAX

[general]	
Función	Descripción
updbindaddr=0.0.0.0:4569	Define la interfaz de red y el puerto UDP. En este caso el "0.0.0.0" es para escuchar a todas las interfaces y el 4569 es el puerto UDP usado las extensiones SIP
srvloOKup=yes	Realiza búsquedas DNS para el envío de correos, debe estar activo cuando se usa proveedores de afuera o externos
delayreject=yes	Retrasa los mensajes de autenticación. Parámetro de seguridad frente a ataques de fuerza bruta
alwaysauthreject=yes	Rechaza usuarios al intentar autenticarse con información o credenciales erróneas
autOKill=yes	Cancela las conexiones si no recibe un <i>ACK</i> en 2000ms. Usado para verificar el estado de las

	conexiones o para terminar llamadas que no tienen respuesta
--	---

Para las configuraciones de las extensiones IAX, es lo mismo que SIP.

C. Buzón de Voz [VOICEMAIL.CONF]

Antes de exponer la configuración de este fichero, se debe definir la función de esta característica de *Asterisk*.

La definición del buzón de voz es simplemente el almacenamiento de los mensajes voz dejados cuando una llamada entrante no era contestada. Del mismo que las antiguas contestadoras, aunque en la actualidad este servicio ha ido mejorando. Las centrales modernas tienen esta característica y son prescindibles en su sistema. Resumiendo, el buzón de voz tiene la función de almacenar los mensajes y avisar a los usuarios del evento, ya sea por correo o simples notificaciones.

Obviamente *Asterisk* integra este sistema y es uno muy flexible. Sus principales características son: saludos personalizados, organización de los mensajes por carpeta, asociación de varias extensiones con un buzón y viceversa. Notificaciones de nuevos mensajes por correo electrónico, entre otras. Lo interesante y bueno de *Asterisk*, permite acceder al buzón de voz de varias maneras, tales como la típica de las centrales mediante un menú implícito con funciones de lectura y eliminación de mensajes a través de números o letras (es decir, presiona tal tecla para tal función), otros redireccionan los mensajes hacia el correo y acceder a ellos mediante aplicaciones o también el uso de servidores especiales (*SQL*, *IMAP*, *POP*, etc.).

El archivo ***voicemail.conf*** permite configurar el buzón de voz para la central telefónica. La función es simple, si una llamada no puede ser contestada o la línea está ocupada, esta es redireccionada al buzón donde se almacenará el mensaje y finalizar la conexión, como se observa en la figura 13.

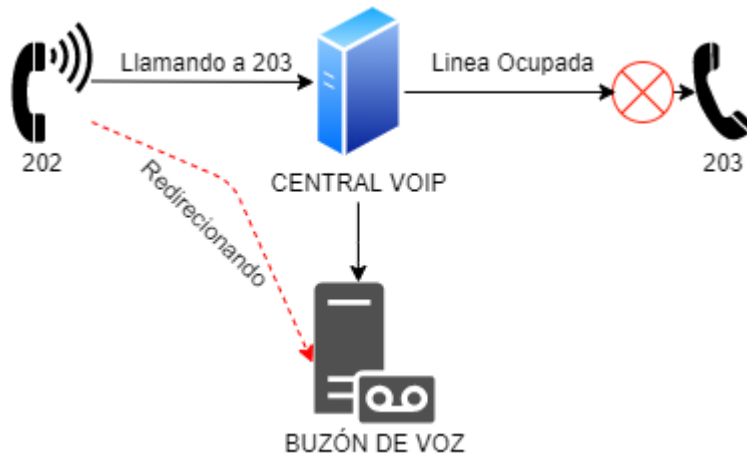


Figura 13: Sistema de buzón de voz

Lo característico de *Asterisk* es que es muy flexible, permitiendo organizar al modo en que sea más útil. Esto es importante en centrales con muchos clientes que abordan una gran cantidad de mensajes.

Para que *Asterisk* ejecute este sistema, se conecta a un servidor SMTP instalado en la central que se encarga de enviar los mensajes (también es posible utilizar servidores remotos). Uno conocido es *SendMail* pero con el paso del tiempo, debido a la poca fiabilidad y la complejidad, surgen nuevos servidores SMTP ligeros, aunque poco seguros. El principal problema era el SPAM excesivo, dada la ligereza del protocolo (UDP que solo envía y recibe sin confirmar nada), la verificación de los mensajes entrantes, etc. Para este caso, el proyecto no se enfocará mucho en esta parte, pero es recomendable utilizar cuentas de servidores verificados y seguros.

Se hace uso del servidor por defecto que viene en Debian que es *exim*. El fichero ***voicemail.conf*** ocupa tres segmentos: **[general]**, **[zonemessages]** y **[default]**.

Para el bloque [general] se configuran los parámetros principales como el tiempo de grabación, la cantidad de mensajes, el número de reintentos, etc. Como se expone en la tabla 9

Tabla 9: Configuración [general] del voicemail.conf

[general]	
Función	Descripción
format=wav	Formato en que se guardan los mensajes de voz: .gsm o .wav
serveremail=Asterisk@boxvoice.org	Nombre o dirección por el cual se identifica por correo ante el remitente
attach=yes	Indica si el mensaje se envía por correo electrónico
maxmsg=50	Máximo número de mensajes
maxsecs=200	Duración máxima (segundos) para cada mensaje de voz
minsecs=2	Mínima cantidad de segundos para que el mensaje sea reconocido y enviado
maxgreet=100	Define el máximo de segundos del mensaje de bienvenida
maxsilence = 5 silencethreshold = 128	Si el silencio dura más de 5 segundos, el servidor corta la llamada al buzón de voz. Para un tiempo sin límites, debe escribirse 0
maxlogins=5	Cantidad de intentos permitidos para ingresar al buzón
moveheard=yes	Todos los mensajes ya escuchados se mueven a la carpeta <i>hold</i>
userscontext=default	Contexto por defecto para los usuarios
pbxskip=yes	Permite omitir la abreviación <i>pbx</i> en el remitente
fromstring=boxvoice	Nombre del remitente
mailcmd=/usr/sbin/sendmail-t	Programa por defecto para el envío de mensajes
tz=Peru	Definición de la zona horaria para indicar la hora y fecha en que se recibió el correo de voz
attachfmt=wav	Define el formato de audio con el que se enviarán el mensaje de voz adjunto al correo electrónico
saycid=yes	Permite escuchar los datos de quien dejó el mensaje

sayduration=no	Permite decir o no la duración del mensaje de voz
saydurationm=1	Indica la cantidad mínima de la duración del correo de voz para ser anunciada
listen-control-forward-key=*	Tecla para adelantar el mensaje que se escucha
listen-control-reverse-key=#	Tecla para retroceder el mensaje que es escucha
listen-control-pause-key=0	Tecla para pausar
listen-control-restart-key=1	Tecla para escuchar el mensaje otra vez
listen-control-stop-key=23456789	Tecla para detener el mensaje y regresar al menú
bACKupdeleted = 200	Cantidad de mensajes que se almacenan en la carpeta <i>deleted</i>

El bloque [zonemessages] define las zonas horarias. Es necesario fijar distintas zonas horarias, ya que esto ayuda si hay muchos usuarios con horas distintas, permitiendo poder indicar el tiempo en que el mensaje fue recibido. En la tabla 10 se muestra dicha configuración.

Tabla 10: Configuración [zonemessages] del voicemail.conf

[zonemessages]	
Función	Descripción
Peru=America/Bogota 'vmreceived' aebY 'digits/at' HM eastern=America/New_York 'vmreceived' Q 'digits/at' IMp central=America/Chicago 'vmreceived' Q 'digits/at' Imp central24=America/Chicago 'vmreceived' q 'digits/at' H N 'hours' military=Zulu 'vmreceived' q 'digits/at' H N 'hours' 'phonetic/z_p' european=Europe/Copenhagen 'v m-received' a d b 'digits/at' HM	Definición de las zonas horarias

El contexto [default] define a los usuarios que usaran el buzón de voz, donde todos pueden estar en el mismo contexto o pertenecer a otros. Se tiene en la tabla 11:

Tabla 11: Configuración [default] del voicemail.conf

[default]	
Función	Descripción
202 => 1234,GOKu,kakaroto@gmail.com	Extensión, contraseña y correo de usuario

D. FEATURES.CONF

Este fichero provee a la centralita configuraciones personalizadas para darle opciones especiales a ciertas funciones, como la grabación, parqueo y transferencia de llamadas, sala de conferencias, etc. Normalmente todos los ficheros de *Asterisk* contienen configuraciones por defecto, si solo se desean funciones predeterminadas, no es necesario configurar demasiado. Por otro lado, para funciones únicas e ideales al requerimiento propio, la flexibilidad de *Asterisk* es de gran ayuda. En este caso, solo se muestra la transferencia y parqueo de llamadas, que son las más usadas.

Sobre la función de transferencia de llamadas se tienen dos tipos: **ciegas** y **atendidas**. Las transferencias ciegas ocurren cuando la llamada desviada no se anuncia hacia la extensión destino. Es decir, no revisa si el destino puede o no atender la llamada. Las transferencias atendidas son todo lo contrario, se anuncian para determinar si la llamada desviada será atendida o no.

El **parqueo de llamadas** permite dejar en espera (estacionar) llamadas recibidas para no perderlas y continuarlas luego. En términos simples, si hay escenarios como llamadas con mayor prioridad o realizar pausas breves, la llamada actual se pone en espera y se atiende la de mayor prioridad, al finalizar, se continua con la llamada anterior.

Todas estas funciones se configuran en el **features.conf** que lo conforman tres bloques: **[general]** que define los parámetros

generales, **[featuremap]** encargado de establecer la combinación de teclas a pulsar para realizar las funciones dichas. Cabe mencionar que, para esta función, es necesario un puente entre las llamadas (debe estar definido en el fichero *extensión.conf* y en la aplicación Dial). Por último **[applicationmap]** donde se puede agregar funciones utilizables en el transcurso de una llamada. Un ejemplo básico se muestra en la tabla 12 y 13:

Tabla 12: Configuración [general] del features.conf

[general]	
Función	Descripción
parkext => 800	Extensión donde se parquea la llamada
parkpos => 701-710	Extensiones reservadas para el parqueo de llamadas
context =>parkedcalls	Contexto para el <i>extensions.conf</i>
parkingtime => 60	Tiempo en segundos que una llamada puede estar parqueada
comebACKtoorigin = yes	La afirmación define que después finalizar el <i>parkingtime</i> , la llamada vuelve a la extensión que la parqueo
parkedplay = both	Define a quien enviar el <i>beep</i> cuando se llama a la extensión parqueada: <i>parked</i> (canal parqueado), <i>caller</i> (llamante) y <i>both</i> (ambos).
transferdigittimeout =>5	Define el máximo de segundos de espera entre dígitos cuando se transfiere una llamada
xfersound = beep	Define el sonido que indicara que la llamada con transferencia atendida se completó
xferfailsound = beeper	Define el sonido que indicara que la transferencia falló
featuredigittimeout= 2000	Tiempo en segundos entre dígitos para activar las funcionalidades del [featuremap]
atxfernoanswertimeout=15	Tiempo máximo para que una llamada sea contestada con el método atendido
atxferdropcall = no	Definir con "No" o "Si" hace que <i>Asterisk</i> devuelva una llamada transferida que en el método atendido fue colgada

atxferloopdelay = 15	Es el tiempo en segundos de espera antes de devolver la llamada con <i>atxferdropcall = no</i>
atxfercallbACKretries = 3	Define el número de veces que se intentara devolver la llamada

Tabla 13: Configuración [featuremap] del feature.conf

[featuremap]	
Función	Descripción
blindxfer => #	Tecla que se debe digitar seguido del número de la extensión para realizar una transferencia ciega
disconnect => *	Tecla para terminar la llamada
atxfer => *2	Tecla que se debe digitar seguido del número de la extensión para realizar una transferencia atendida
parkcall => *7	Secuencia de teclas para parquear una llamada, se puede digitar de esta forma o directamente la extensión 700 con #700

E. Distribución Automática de Llamadas

Esta parte se centra en la gestión que se debe realizar sobre las llamadas. Para ello se administra un **call center** (o centro de llamadas). Este es un término ya conocido, por lo que, resumiendo, el propósito de este sistema es atender grandes cantidades de llamadas entrantes y salientes con fines distintos (soporte, atención al cliente, comercio, etc.). Los encargados de realizar dichas funciones son los llamados **agentes**, capacitados rigurosamente en cumplir todos los requerimientos en la atención y disponibilidad del servicio.

El servicial multitarea *Asterisk* contiene un flexible sistema de distribución de llamadas. Esto permite la configuración de distintos algoritmos para una gestión correcta de las llamadas, basado en la necesidad del cliente. Para implementar un call center en *Asterisk*, se deben configurar tres ficheros: **agents.conf**, **queues.conf** y **extensions.conf**. Por supuesto, se

debe configurar un grupo de extensiones para los agentes en el *sip.conf*.

1. AGENTS.CONF: Para que el *call center* pueda atender las llamadas, se debe registrar un grupo de agentes, lo cual puede ser estático (solo puede ingresar al sistema desde la misma extensión o equipo terminal) o dinámico (se puede conectar desde diferentes extensiones o equipos terminales). En caso de solo configurar agentes del tipo estático, no se debe realizar ningún cambio en el fichero. Si se desea contar con agentes dinámicos, se deben configurar los parámetros generales y el contexto *agents*. Como se observa en la tabla 14 y 15.

Tabla 14: Configuración [general] del agents.conf

[general]	
Función	Descripción
multiplelogin=yes	Permite conectarse como agente múltiple a una misma extensión

Tabla 15: Configuración [agents] del agents.conf

[agents]	
Función	Descripción
maxlogintries=3	Número de intentos máximo que el agente tiene para acceder
autologoff=20	Tiempo máximo que tiene un agente para contestar, sino automáticamente se desconectara
autologoffunavail=no	Si el agente se vuelve "no disponible", el agente se desconecta automáticamente
endcall=yes	los agentes terminan la llamada con *
ACKall=yes	Los agentes se registran con #.
acceptdmf=#	<i>tonotdmf</i> de # para el registro
endtdmf=*	<i>tonotdmf</i> de * para finalizar
updatecdr=no	Define si se registra en el cdr

savecallsin=/var/calls	Directorio donde se guardarán las grabaciones de las llamadas
recordedagentcalls=yes	permite grabar las llamadas de un agente
agent => 501,1234,KasugaArata	Define los agentes que se van a utilizar en la cola, seguidos de la contraseña y su nombre

2. QUEUES.CONF: La principal función de las colas es poner en espera las llamadas entrantes que no puedan ser atendidas por los agentes. Por supuesto, para agentes dinámicos y estáticos contemplan distintas configuraciones. A continuación, se expone los parámetros generales y la configuración para ambos agentes en la tabla 16, 17 y 18.

Tabla 16: Configuración [general] del queues.conf

[general]	
Función	Descripción
persistentmembers=yes	Se coloca "yes" para guardar la configuración cuando se trabaja con <i>agents</i> dinámicos, es decir cuando <i>Asterisk</i> arranque nuevamente, colocara a cada agente en su respectiva cola
autofill = yes	Este parámetro es importante. Permite que los usuarios sean atendidos de forma paralela hasta que haya agentes disponibles, si se define como "no" la forma de trabajar será que un usuario debe esperar estar primero en la cola para ser atendido aun así existan agentes disponibles

Tabla 17: Configuración [queues] para agentes dinámicos

[queues] dinámico	
Función	Descripción
strategy=random	Se define el tipo de estrategia que se va utilizar para atender las colas, entre las opciones se tiene; 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, fewestcalls : asigna la siguiente llamada al agente que menos llamadas haya atendido, random : asigna la siguiente llamada aleatoriamente a cualquier agente disponible, rrmemory : distribuye las llamadas por turnos entre los agentes disponibles y recuerda el último agente que le intentó llamar, wrandom : asigna la llamada aleatoriamente usando una métrica basada en penalidades
timeout=20	Tiempo en segundos en el cual permanece timbrando antes de considerar que la llamada no va a ser contestada
retry=5	Tiempo en segundos que se debe esperar antes de volver a timbrar a otro agente
wrapuptime = 1	Tiempo en segundos que debe expirar después de terminar una llamada y pasar otra al agente
maxlen = 0	Número máximo de llamadas que se puede colocar en una cola, el 0 implica que no tiene ningún límite
announcefrequency=0	Tiempo en el cual se le anunciará al usuario en qué posición se encuentra en la cola
announce-holdtime=no	Anuncia junto con la posición el tiempo estimado de espera
ringinuse=no	El "No" indica que los agentes que estén ocupados, no serán timbrados.
member=>Agent/501 member=>Agent/502	Configuración de agentes dinámicos, que formarán parte de esta cola

Tabla 18: Configuración [queues1] para agentes estáticos

[queues1]	
Función	Descripción
member=>SIP/601 member=>SIP/602	Contexto donde se definen agentes estáticos. La diferencia entre queues1 y queues es que aquí se definen las extensiones que formarán parte de la cola

El número de colas que se desee configurar será propio de los fines de la aplicación para el *call center*. En base a los agentes dinámicos, para que sus extensiones se conviertan en agentes, deben acceder con su número y contraseña del agente. Para los del tipo estático, la configuración de la cola define que estas extensiones son parte de ella.

3. MUSICONHOLD.CONF: Un archivo de configuración muy útil. Cuando un cliente se encuentra esperando a ser atendido, colocar música en espera creará un ambiente agradable. *Asterisk* usa archivos del tipo *.mp3* y *.wav*. Normalmente la central usa una gran capacidad de procesamiento para las llamadas, por lo que es recomendable usar archivos más livianos para esta función. Sin embargo, para modo de prueba, no será necesario la conversión de los formatos. Se tiene la siguiente configuración básica en la tabla 19

Tabla 19: Configuración [default] del musiconhold.conf

[default]	
Función	Descripción
mode=files directory= /var/lib/Asterisk/moh	Solo es necesario definir la dirección a la carpeta donde se encuentra la música

F. Enlaces de comunicación entre Centrales *Asterisk*

Con *Asterisk* es posible interconectar dos centrales que se encuentran ubicadas remotamente, en cualquier parte del mundo. Sin necesidad de implementar una nueva red para este enlace o poseer una línea especial dedicada. Mientras se tenga una conexión a internet, estas dos (o más) estaciones pueden establecer una comunicación. Ya sea mediante enlaces dedicados por proveedores de *VoIP* o redes privadas virtuales (VPN) para un monitoreo o control del tráfico de datos.

Se establecen mediante dos tipos de enlace troncal, IAX o SIP. La elección dependerá de las necesidades o ventajas que

presenta cada protocolo. Por ejemplo, SIP normalmente presenta problemas frente a NAT, por lo que usar IAX es conveniente. Sin embargo, para poder conectarse hacia la PSTN, los *gateways* de voz admiten solamente SIP, de modo que es favorable el protocolo de inicio de sesión. El uso o aplicación que se da, será elección del cliente. Definido esto, se presenta el ejemplo de una configuración de enlace troncal mediante SIP. Este proceso se realiza en el fichero *sip.conf*:

Tabla 20: Configuración de un enlace troncal SIP

IP de Central Principal: 192.168.4.4 IP de Central Sucursal: 192.168.5.5	
Central Principal	Central Sucursal
[CORE] username=sede type=friend secret=1234 host=192.168.5.5 permit= 192.168.5.5 auth=md5 disallow=all allow= <i>ulaw</i> context= internal-extensions peercontext= internal-extensions <i>trunk=yes</i>	[SEDE] username=core type=friend secret=1234 host=192.168.4.4 permit= 192.168.4.4 auth=md5 disallow=all allow= <i>ulaw</i> context= internal-extensions peercontext= internal-extensions <i>trunk=yes</i>

Como se muestra en la tabla 20, la configuración de los parámetros es similar a la de una extensión normal, la diferencia es que la IP debe ser de la central contraria, la contraseña debe ser igual en ambos lados. El parámetro ***trunk*** se establece en “yes” para permitir el enlace troncal y definir los contextos. El nombre de usuario no necesariamente debe ser el de la central contraria, esto depende de quien configure ambos ficheros.

G. Enlace con la PSTN

Esta es otra de las ventajas de *Asterisk*, permitir la comunicación con líneas de la red de telefonía tradicional, sin necesidad de

instalar nuevas líneas especiales. Solo basta una red de internet y un *Gateway* de voz, los más usados y conocidos son de la marca Grandstream.

La configuración en *Asterisk* se realiza en el fichero *sip.conf* debido a que el *Gateway* trabaja sobre este protocolo. Como si se creara una nueva extensión, como se muestra en la tabla 21:

Tabla 21: Configuración de una extensión para la PSTN

En <i>sip.conf</i>	
Función	Descripción
[100]	Nombre de usuario
username = 100	
context= internal-extensions	Contexto de la extensión
secret = 1234	Contraseña que se configuro el <i>fxoport</i> del <i>Gateway</i> de voz
type = peer	
host = 192.168.4.10	Host del <i>Gateway</i> de voz
port=5060	Puerto SIP
dtmfmode = rfc2833	Modo de los tonos dtmf que se define el <i>Gateway</i> de voz
canreINVITE = no	
disallow=all	
allow = alaw	

H. CDR (Registro de llamadas)

El *Call Detail Record* o CDR son registros generados automáticamente por la central y pueden ser almacenados en distintos formatos. La información que contienen son principalmente el número de llamadas realizadas, la fecha y duración, el origen y destino, así como también el costo de las mismas. Además, se pueden almacenar ficheros con el total de llamadas hechas en la central, o individuales por extensión.

Estos ficheros son muy útiles, pues permiten realizar un monitoreo de las llamadas realizadas y darles aplicaciones como informes y recibos de gastos hechos por los usuarios, con el fin mostrarle transparencia frente al servicio que contratan.

La configuración que se realiza puede depender del formato que se desea. Por ejemplo, si se desea registrar el CDR separado por

comas, es decir un fichero del tipo CSV, se debe configurar un módulo en el archivo **modules.conf** (al final del fichero se agrega *load=>cdr_csv.so*) además de configurar el fichero *cdr.conf*. Esta configuración del fichero en formato .csv es ideal cuando no se quiere utilizar muchos recursos de la central cuando se desea obtener la información del CDR a través de una base de datos, puesto que es muy fácil transformar los archivos .csv y no se requieren instalación de módulos innecesarios.

Los parámetros a configurar en el *cdr.conf* se muestran en la tabla 22:

Tabla 22: Parámetros generales del cdr.conf

[general]	
Función	Descripción
enable=yes	Activa o desactiva la utilización del CDR
unanswered=yes	Registrar las llamadas no atendidas
batch=no	Permite almacenar los datos de las llamadas en un buffer, para luego enviarlos a donde se quiera almacenar. Cabe mencionar que, activando esta opción, si bien se liberaría un poco de procesamiento a <i>Asterisk</i> , puede ocasionar posibles pérdidas de datos, en caso que <i>Asterisk</i> se detenga.
size=200	En caso que se active la opción <i>Batch</i> , se define la cantidad de registros de CDR que se acumularán en el buffer, antes de ser enviados al medio de almacenamiento
time=200	En caso que se active la opción <i>Batch</i> , se define la cantidad de tiempo máximo que deberán estar los registros almacenados en el buffer
scheduleroonly=no	<i>Asterisk</i> 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 <i>Batch</i> 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 <i>Identificación Única del CDR</i> , El valor predeterminado es No.
loguserfield=no	Variable <i>Campo de usuario del CDR</i> , El valor predeterminado es No.
accountlogs=yes	Crear un archivo CSV separado para cada valor distinto de la variable CDR <i>accountcode</i> , El valor predeterminado es Yes

I. *EXTENSIONS.CONF*

Como se menciona anteriormente, este es un archivo muy importante para hacer funcionar a la central. Se configura el *dialplan* o plan de marcado que realizara el servidor telefónico para cada usuario y contexto.

En la actualidad, mayor parte de la configuración viene por defecto con los servicios o aplicaciones que se requieren, que en consecuencia no es necesario realizar muchos cambios. Sin embargo, es necesario explicar el concepto de este fichero y como se configura.

En *Asterisk*, para definir extensiones, servicios o aplicaciones viene de la siguiente forma:

Exten=>patrón, índice; acción(parámetros)

Donde:

- exten: hace referencia a la definición de una extensión seguido de la acción que realizara, todas las líneas del plan de marcado comienzan con esta palabra.
- patrón: se define la extensión, donde se puede escribir de forma numérica o combinaciones complejas (“700” o “_x.,2” normalmente, se usan combinaciones complejas cuando se quiere abarcar varias extensiones).
- índice: orden en que se realizan las acciones.
- acción: se define las acciones a ejecutar

Definido el formato, se procede a configurar el *extensions.conf*, que, como todos los ficheros, hay parámetros generales o por defecto. A continuación, se presenta la configuración de algunos de estos:

- [default]
 exten => _Y.,2,Hangup(21)
 exten => s,2,Hangup(21)

Esto indica si hay un usuario desconocido sin contexto definido, la llamada será colgada, este contexto es uso dado para la seguridad.

- [parqueo]
 exten => 501,hint,park:501@parkedcalls
 exten => 502,hint,park:502@parkedcalls
 exten => 503,hint,park:503@parkedcalls

Como su nombre lo indica, este contexto define las extensiones que servirán para dejar en espera o parquear una llamada. Nótese la relación de la extensión con el contexto definido

parkedcalls en el fichero *features.conf* para alcanzar la acción del parqueo de llamadas.

- *[internal-extensions]*

En este contexto se definen las extensiones SIP e IAX configuradas anteriormente en sus ficheros correspondientes, la comunicación hacia la PSTN y los enlaces entre centrales. Cada línea sigue el formato antes mencionado, tanto las letras y los números indican patrón, prioridad, etc. Para poder entender a profundo cada línea, se recomienda ver el anexo al final.

Plan de marcado desde y hacia la PSTN

```
exten => 200,1,Goto(ivr,s,1)
exten => _9X.,1,Dial(SIP/200/${EXTEN:1})
```

Plan de marcado entre centrales.

```
exten => _5XXXX,1,Dial(SIP/sede/${EXTEN},90,tr) exten
=> _5XXXX,2,Hangup
```

Para la comunicación interna en la central.

```
exten => 4001,1,Dial(SIP/3001,20,Tt)
exten => 4001,2,Followme(3001,san)
exten => 4001,n,VoiceMail(4001@default)
```

Para la comunicación entre centrales

```
exten => 43001,1,Dial(SIP/3001,20,Tt)
exten => 43001,2,Followme(3001,san)
exten => 43001,n,VoiceMail(4001@default)
```

Extensiones iax

```
exten => 1001,1,Dial(IAX2/1001,20,Tt)
exten => 1001,n,VoiceMail(5001@default)
exten => 42001,1,Dial(IAX2/1001,20,Tt)
exten => 42001,n,VoiceMail(5001@default)
```

Inclusión de los contextos de las aplicaciones

include => Aplicaciones

include => call-center

include => parkedcalls

include => ivr

- [call-center] (va dentro de *internal-extensions*)

Para agentes estáticos

exten => 33333,1,Answer()

exten => 33333,n,*Ringin*g

exten => 33333,n,SetMusicOnHold(default)

exten => 33333,n,Queue(colas1)

exten => 33333,n,Hangup

Para agentes dinámicos

exten => 2000,1,Agentlogin(\${EXTEN})

exten => 2000,n,Hangu

exten => 4081,1,Answer

exten => 4081,n,*Ringin*g

exten => 4081,n,SetMusicOnHold(default)

exten => 4081,n,Queue(colas)

exten => 4081,n,Hangup

- [Aplicaciones]

Acceso para escuchar los mensajes del Buzón de Voz

exten => 8500,1,Answer()

exten => 8500,n,VoiceMailMain(@default)

Aplicación para grabar IVR

exten => 8991,1,Answer()

exten => 8991,n,Wait(1)

exten => 8991,n,Record(welcome.gsm)

exten => 8991,n,Wait(1)

exten => 8991,n,PlaybACK(welcome)

```
exten => 8991,n,Hangup()
```

- [IVR]

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

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

```
exten => s,n,BACKground(welcome-menu)
```

```
exten => s,n,WaitExten(5)
```

```
exten => 1,1,Goto(internal-extensions,5003,1)
```

```
exten => 2,1,Goto(internal-extensions,1002,1)
```

```
exten => 3,1,Goto(Aplicaciones,9999,1)
```

```
exten => 4,1,Goto(call-center,33333,1)
```

```
exten => *,1,Goto(s,1)
```

```
exten => t,1,PlaybACK(vm-goodBYE)
```

```
exten => t,n,Hangup()
```

```
exten => i,1,PlaybACK(pbx-invalid)
```

```
exten => i,n,Goto(s,1)
```

Para el objetivo del proyecto, que es implementar una central *VoIP* que brinde servicios de generales de telefonía. Estas vendrían a ser la configuración básica del *extensions.conf* luego de haber terminado con los demás ficheros como el *sip.conf*. Se ha puesto solo las líneas de configuración y se ha resumido con lo más relevante e importante. Para poder entender cada línea, como se dijo al principio, se recomienda revisar el anexo.

Se debe mencionar que estos no son los únicos archivos que contiene *Asterisk*. Solo se han mencionado los más relevantes y principales para la gestión de la central telefónica. Existen muchos otros ficheros que aportan (como el *followme.conf*). Todo dependerá de la necesidad o requerimientos que se desea obtener con la central *VoIP*.

3.3.4. Configuración de *Asterisk* por *FreePBX*

Asterisk es una poderosa herramienta para servicios *VoIP*. Sin embargo, la gestión o configuración no es tan simple. Para que los usuarios puedan implementar centrales telefónicas con *Asterisk*, deben conocer de manera básica el manejo de sistemas Linux y entender los ficheros o lenguaje que usa *Asterisk*, donde realmente es conocimiento que no todos poseen. Por ello, varias comunidades desarrolladoras han generado y compartido formas de gestionar *Asterisk* en manera muy sencilla. La mayoría de estos softwares o aplicaciones son libres, es decir, gratis. *FreePBX* es una de ellas y una de las más usadas. Este *software* presenta una interfaz gráfica para poder manejar la central *VoIP* de forma práctica y sencilla. Muestra las herramientas o aplicaciones de *Asterisk*, permitiendo gestionarlas gráficamente y no a través de códigos. El único problema que presenta es la limitación de configuración y el consumo de recursos en la *Raspberry Pi*. Hay ciertos archivos (o cambios) que no se pueden configurar desde *FreePBX*, por lo que se debe realizar de manera directa en el CLI de la central. Como toda aplicación que se instala, esta ocupa memoria o recursos, y para el caso de la *Raspberry Pi*, es lo que menos se quiere, gastar recursos y no disminuir el potencial de la central. Esto no es un problema serio, ya que no afecta tanto en el funcionamiento de la central, pero siempre se debe tener en cuenta las aplicaciones y recursos que contiene la central para la resolución de problemas.

Para emplear *FreePBX*, el acceso es remoto. En cualquier navegador, se escribe la dirección IP del servidor y automáticamente se redirige a la plataforma de *FreePBX* como se observa en la figura 14.



Figura 14: Acceso remoto de *FreePBX*

La primera función que se encuentra es el modo de operar: administrador o usuario. En el modo *admin* se encuentran todas las funciones para la gestión de la central telefónica. El panel del usuario es una interfaz sencilla donde el cliente accede a sus servicios, como el buzón de voz, mensajería e historial de llamadas.

Para acceder a ambos paneles, se requiere una contraseña y nombre de usuario. Cuando se ingresa por primera vez, el usuario y contraseña por defecto es *admin*. De ahí en adelante, se puede cambiar estos valores, lo cual es recomendable para ambos paneles (figura 15).

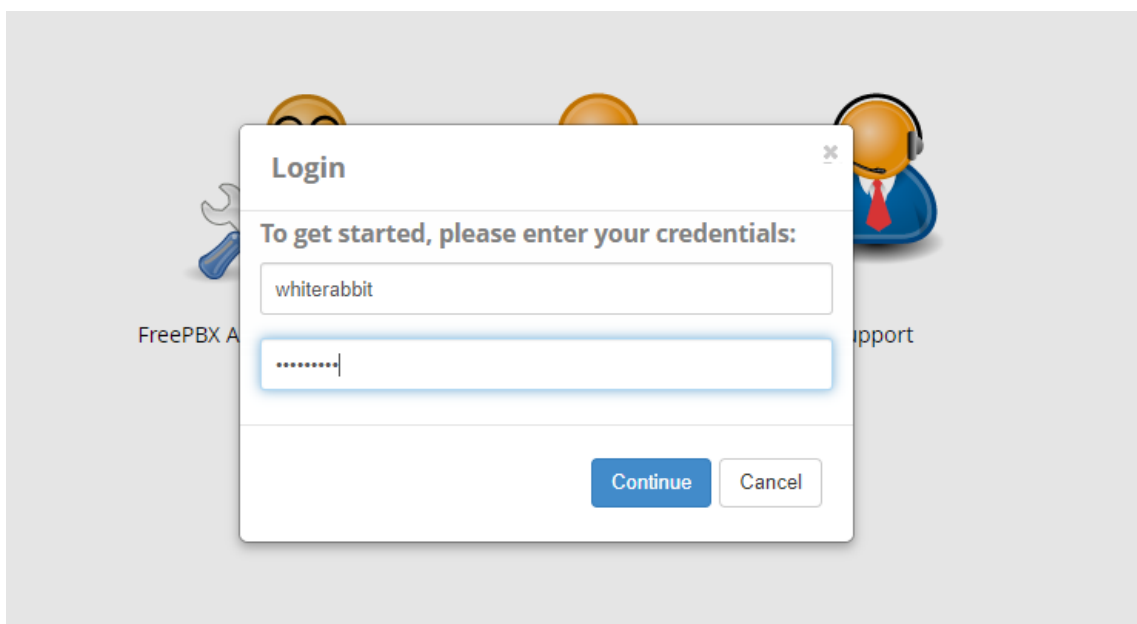


Figura 15: Acceso al panel de administración

Al acceder al panel de administración, la primera imagen es la portada con la información reciente y los cambios dados en la central telefónica, como se observa en la figura 16.

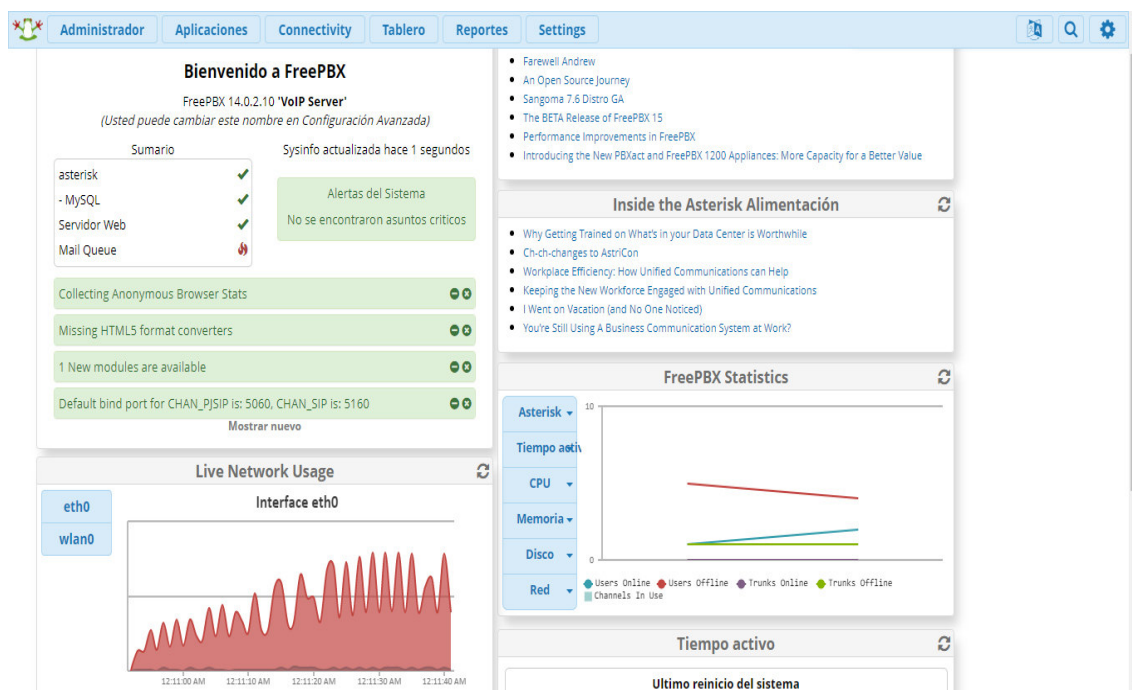


Figura 16: Portada de administración de FreePBX

En este panel se observa principalmente información sobre el estado de la central. Como el uso de la red, estadísticas sobre el uso de *Asterisk*, el tiempo activo, memoria, disco, CPU, etc. En la parte superior se encuentran las funciones que gestionan a la central y es ahí donde se realizan los cambios. A continuación, se muestra algunas configuraciones hechas anteriormente en el método directo de *Asterisk*, como la creación de extensiones, troncales, entre otras.

A. Pestaña Configuraciones

En la pestaña “*configuraciones*” de la parte superior, se puede observar la configuración de ficheros ya antes vistos, como la de IAX, SIP, música en espera, correo de voz, etc. Cada una lleva a su correspondiente panel para su administración (figura 17).

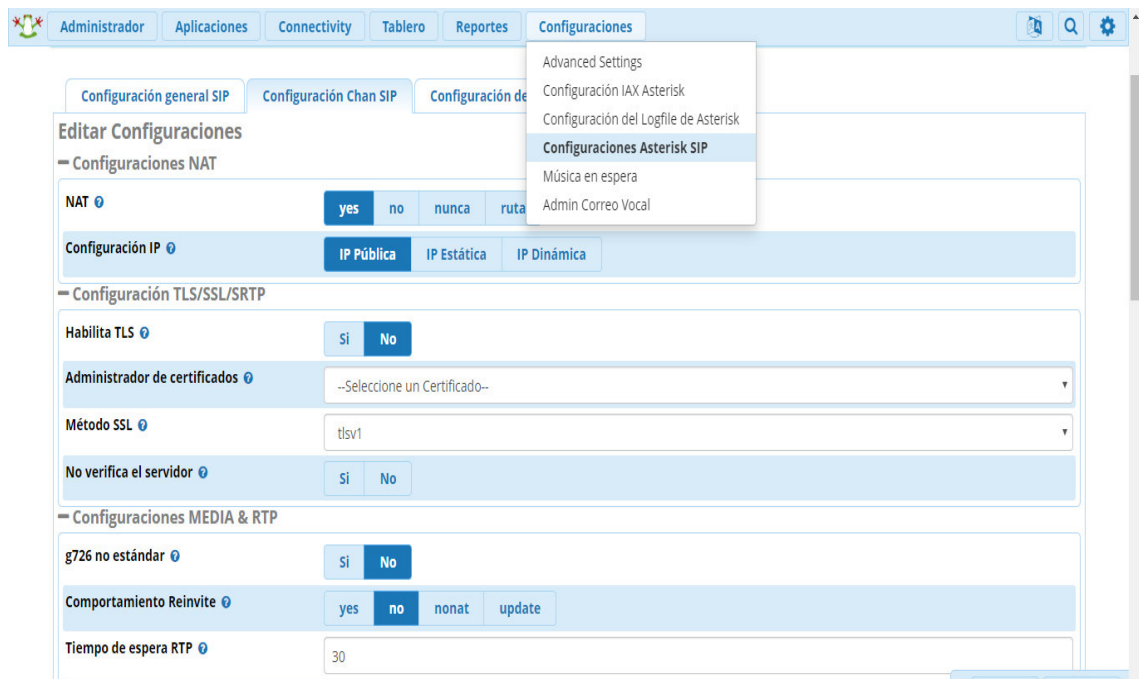


Figura 17: Panel general de SIP

Al seleccionar la configuración de SIP, se encuentran pestañas, cada una con su función respectiva. Aquí se puede establecer la dirección IP, NAT, *ReINVITE*, *codecs* de audio y video, el puerto de escucha, entre otros. Tal y como se vio anteriormente en el fichero de *sip.conf*.

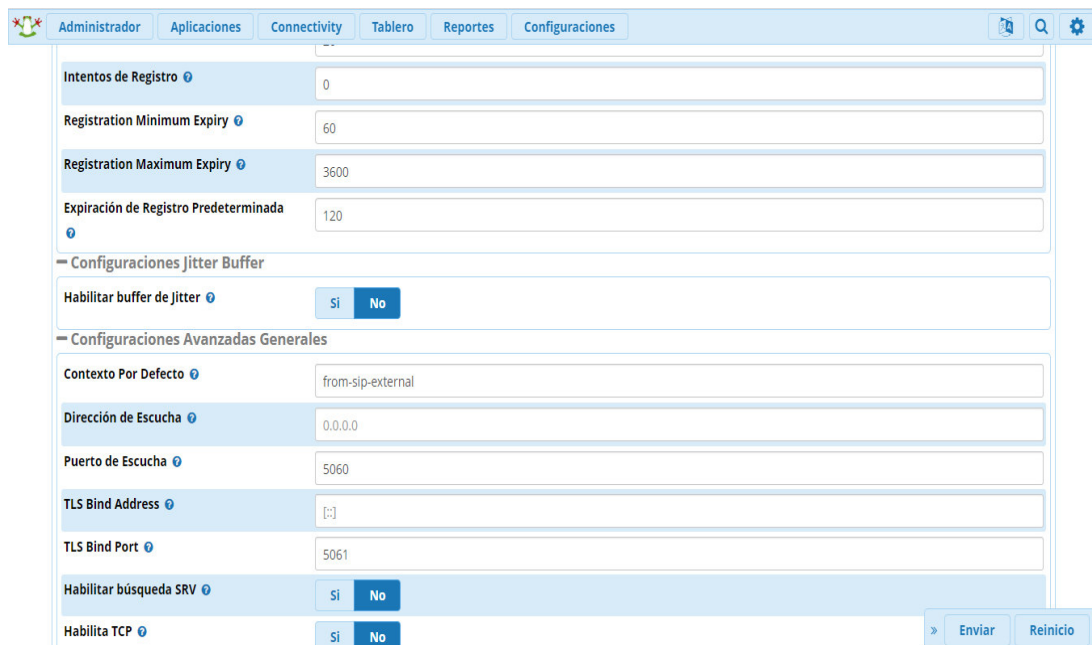


Figura 18: Panel de configuración general de SIP

En la figura 18 se pueden observar valores que se expuso anteriormente, la configuración es similar, lo único que cambia es su interfaz.

La mayoría de las configuraciones generales se encuentran en esta pestaña. Mostrarlas todas sería innecesario pues ya se describió anteriormente. Aquí los valores son iguales, dependiendo de lo que se quiera conseguir.

B. Pestaña Aplicaciones

Como se observa en la figura 19, en esta pestaña se configuran algunos servicios para la central, como las extensiones, grabación de llamadas, IVR, anuncios, etc.

Extension	Name	CW	DND	FM/FM	CF	CFB	CFU	Type	Actions
	rabbit/08	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	iax2	
<input type="checkbox"/>	200	Pan	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	sip	
<input type="checkbox"/>	201	Goku	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	sip	
<input type="checkbox"/>	202	Bell	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	sip	
<input type="checkbox"/>	203	Naruto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	sip	
<input type="checkbox"/>	204	Neji	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	sip	

Figura 19: Pestaña Aplicaciones

Este apartado permite configurar las extensiones o clientes que pertenecerán a la central. Además de los otros servicios que se realiza en este panel.

Esta es una manera muy sencilla de administrar los anexos o extensiones que se crean para la central telefónica. Una observación bien clara sobre este modo de configuración es la combinación de los ficheros. En el método anterior, para configurar ciertos valores, habría de redirigirse a su respectivo

fichero y reescribir los cambios. Esto generara ciertas pérdidas de tiempo y confusión al momento de configurar. En *FreePBX* no sucede de esa manera. La interfaz gráfica combina estos archivos y se configuran desde un solo panel. Esto significa que, si desea configurar extensiones, se accede al panel de aplicaciones e internamente *FreePBX* redirecciona y abre el fichero correspondiente para su configuración. Del mismo modo con todas las pestañas y paneles. Con un solo clic se accede a todos los archivos de *Asterisk*, lo cual es la esencia de las interfaces gráficas, a diferencia de los terminales, donde para configurar el archivo *sip.conf* y *extensions.conf* era por separado.

Para la creación de una nueva extensión, en “*aplicaciones*” acceder al panel de “*extensions*” y crear el tipo de extensión que se desea. Configurando los valores correspondientes como anteriormente se ha realizado.

Add SIP Extension 5001

General Voicemail Advanced Pin Sets

— Add Extension

This device uses **CHAN_SIP** technology listening on Port 5060 (UDP)

User Extension ⓘ	5001
Display Name ⓘ	EXAMPLE
Outbound CID ⓘ	5001
Secret ⓘ	1234

Really Weak

Figura 20: Creación de una extensión SIP

Add IAX2 Extension **3002**

General Voicemail Advanced Pin Sets

— Add Extension

User Extension ? 3002

Display Name ? exAMPLE

Outbound CID ? 3002

Secret ? 1234

Figura 21: Creación de una extensión IAX2

En la fig. 20 y 21, la similitud es notable, se pueden crear extensiones y activar el correo de voz y algunas otras funciones, tal y como se desarrollaba en el fichero *sip.conf* y *extensions.conf*. Del mismo modo, (en la figura 22) con la creación de colas y los agentes en los ficheros *agents.conf* y *queues.conf*.

Administrador Aplicaciones Connectivity Tablero Reportes Settings Apply Config

Colas Add Queue

General Settings Queue Agents Opciones de Agentes & Tiempo Opciones de Capacidad Caller Announcements Advanced Options Reiniciar es

Agentes fijos

Dynamic Agents

Agent Quick Select

- 07 (rabbit/08)
- 200 (Pan)
- 201 (Goku)
- 202 (Bell)
- 203 (Naruto)
- 204 (Néji)

Figura 22: Adición de colas y agentes

Puede verse la configuración para ambos tipos de agentes y seleccionar la extensión que actuara como uno.

C. Pestaña Conectividad

Como su nombre lo indica, esta pestaña se encarga de la conectividad. Aquí es donde se configuran los enlaces troncales, las rutas entrantes y salientes, etc. Como se muestra en la figura

23, la configuración es similar a la que se escriba en la terminal de Asterisk.

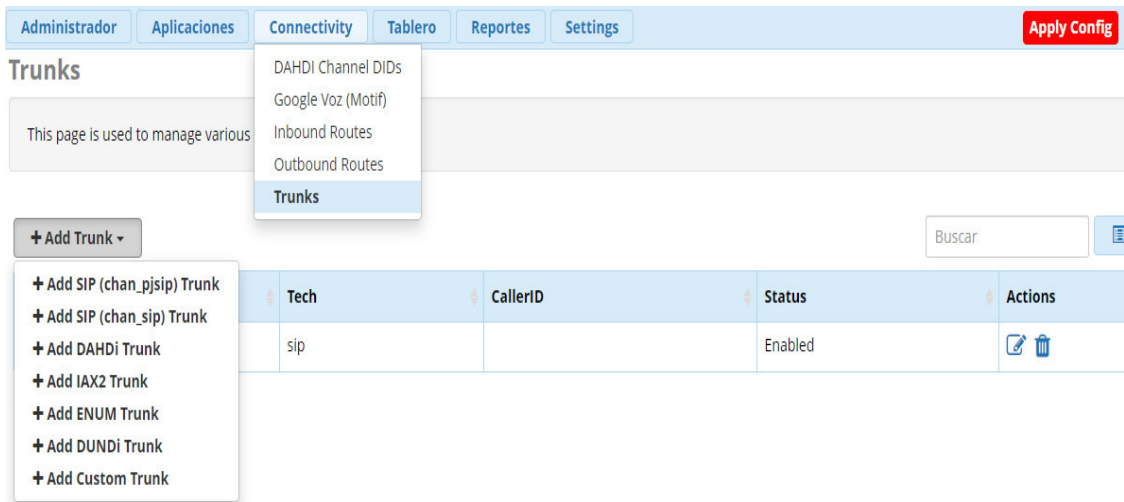


Figura 23: Pestaña Conectividad

Para agregar un enlace troncal, se accede al panel “trunk” y se escoge el tipo de troncal a establecer. Ya seleccionado el tipo, se redirige al cuadro de configuración general.

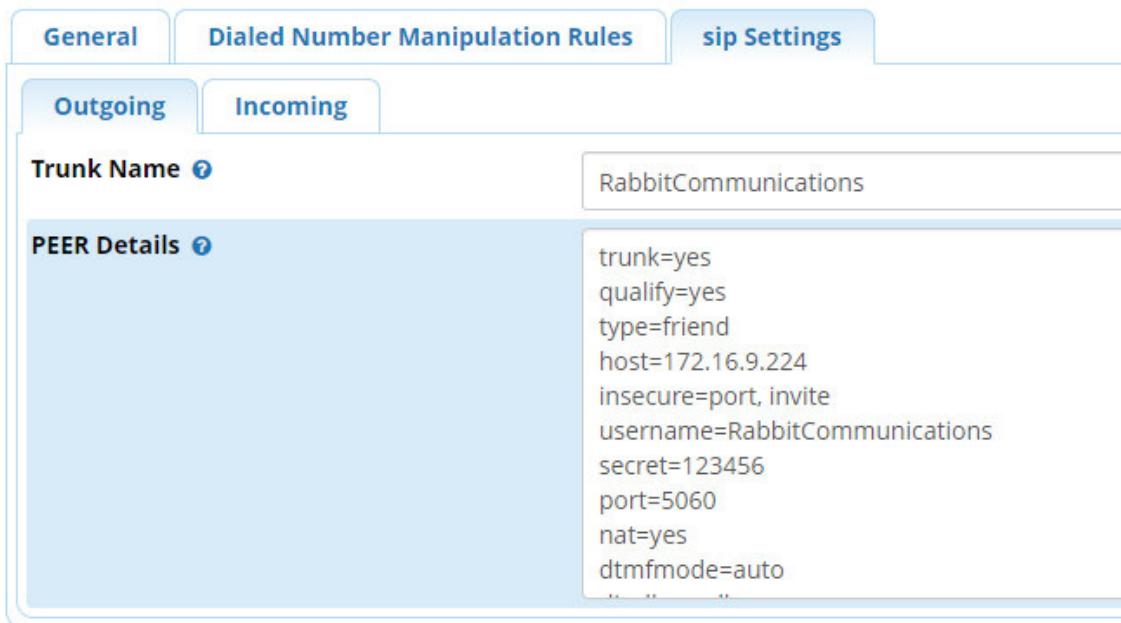


Figura 24: Configuración Outgoing de troncal SIP

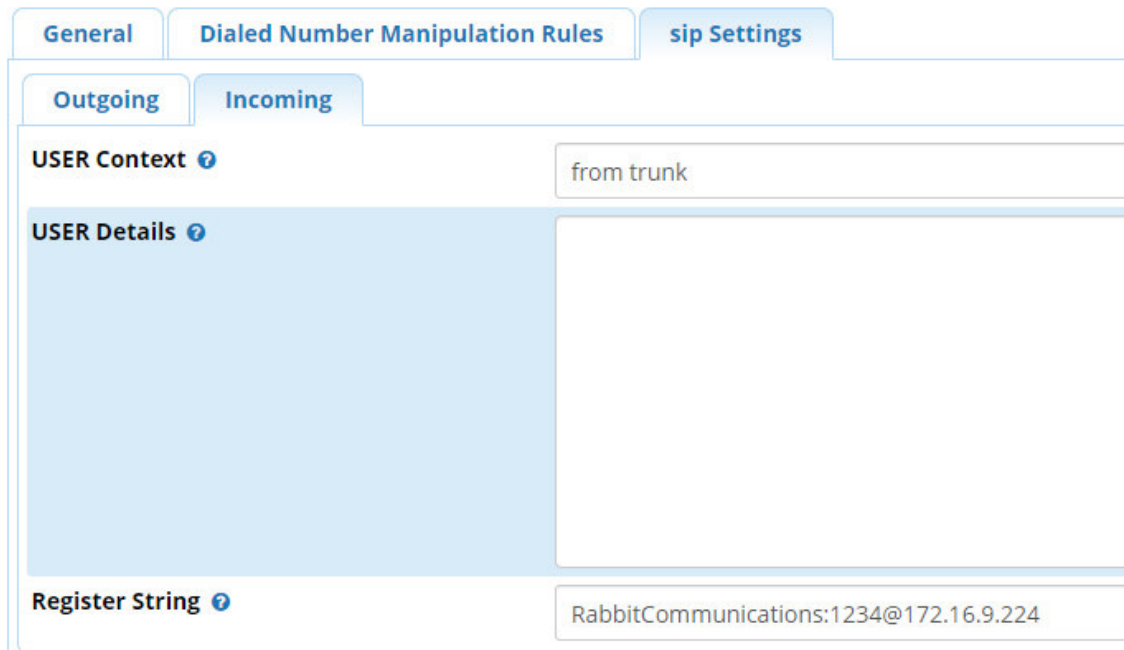


Figura 25: Configuración Incoming de troncal SIP

Como se muestra en la Fig.24 y 25, de esta manera, se puede enlazar sedes o centrales ubicadas remotamente y establecer una red de comunicación amplia, teniendo en cuenta los estados confiabilidad y seguridad.

D. Pestaña Reportes

Aquí es donde se almacena toda la información correspondiente al estado de la central. En este panel no solo contiene un resumen actualizado de *Asterisk*, también en donde se registra el CDR y toda la información en general, como se muestran en las figuras 26 y 27.

Administrador Aplicaciones Connectivity Tablero Reportes Settings Apply Config

Información Asterisk

Esta página provee informaciones varias acerca de Asterisk
Version Asterisk Corriente: 13.20.0

Información Asterisk
 Archivos de bitácora de Asterisk
 Reporte de Llamadas
 Call Event Logging (CEL)

Resumen

Asterisk System uptime: 5 hours, 54 minutes, 57 seconds
 Last reload: 5 hours, 54 minutes, 57 seconds

Canal(es) SIP activos: 2	Canal(es) Activos PJSIP: 0	Canal(es) IAX2 activos: 0
Registro SIP: 2	Registros PJSIP: 0	Registro IAX2: 1
Pares SIP:		
Conectados: 1	Endpoints PJSIP Disponible: 0	Pares IAX2: Conectados: 1
En Línea-No Monitoreado 0	Indisponible: 0	Desconectados: 0
Desconectados: 5	Desconocido: 2	Sin monitorizar: 0
Fuera de Línea-No monitoreado: 0		

Resumen

Registros

Canales

Pares

Información Chan_Sip

Información Chan_PJSip

Información IAX

Conferencias

Suscripciones

Usuarios del buzón de voz

Colas

DAHDI

Figura 26: Pestaña Reportes

Para ver el registro de llamadas, se debe acceder al panel de “Reporte de Llamadas”. Aquí es donde se almacena toda esta información y lo interesante es su motor de búsqueda interactiva y dinámica, ya que contiene un orden y permite buscar llamadas entrantes y salientes en específico, sean por fechas o números. Además de indicarle que genere un archivo del formato .csv.

Buscar CDR

Ordenar por	Search Conditions	Extra Options
<input checked="" type="radio"/> Fecha Llamado	Desde: 01 Octubre 2019 00:00 Hasta: 31 Octubre 2019 23:59	<input checked="" type="checkbox"/> Buscar CDR Report Type: <input type="checkbox"/> CSV File <input type="checkbox"/> Gráfico Llamadas Result Limit: 100
<input type="radio"/> Número Llamante	<input type="text"/> No: <input type="checkbox"/> Comienza con: <input type="radio"/> Contiene: <input type="radio"/> Termina con: <input type="radio"/> Exacto:	
<input type="radio"/> Nombre Llamante	<input type="text"/> No: <input type="checkbox"/> Comienza con: <input type="radio"/> Contiene: <input type="radio"/> Termina con: <input type="radio"/> Exacto:	
<input type="radio"/> Número Llamante Saliente	<input type="text"/> No: <input type="checkbox"/> Comienza con: <input type="radio"/> Contiene: <input type="radio"/> Termina con: <input type="radio"/> Exacto:	
<input type="radio"/> DID	<input type="text"/> No: <input type="checkbox"/> Comienza con: <input type="radio"/> Contiene: <input type="radio"/> Termina con: <input type="radio"/> Exacto:	
<input type="radio"/> Número Destino	<input type="text"/> No: <input type="checkbox"/> Comienza con: <input type="radio"/> Contiene: <input type="radio"/> Termina con: <input type="radio"/> Exacto:	
<input type="radio"/> Nombre Destino	<input type="text"/> No: <input type="checkbox"/> Comienza con: <input type="radio"/> Contiene: <input type="radio"/> Termina con: <input type="radio"/> Exacto:	
<input type="radio"/> Campo Usuario	<input type="text"/> No: <input type="checkbox"/> Comienza con: <input type="radio"/> Contiene: <input type="radio"/> Termina con: <input type="radio"/> Exacto:	
<input type="radio"/> Código Cuenta	<input type="text"/> No: <input type="checkbox"/> Comienza con: <input type="radio"/> Contiene: <input type="radio"/> Termina con: <input type="radio"/> Exacto:	
<input type="radio"/> Duración	Entre: <input type="text"/> Y: <input type="text"/> Segundos	
<input type="radio"/> Disposición	Disposición Todos <input type="checkbox"/> No: <input type="checkbox"/>	
<input type="button" value="Recientes Primero"/>	<input type="button" value="Por Grupo: Día"/>	<input type="button" value="Buscar"/>

Figura 27: Motor de búsqueda de CDR en FreePBX

La búsqueda se puede realizar por categorías, indicarle que genere un archivo .csv o un formato para Excel con el fin de realizar un control o monitoreo de todas las llamadas entrantes y salientes.

Se ha mostrado las pestañas y paneles principales para la configuración de la central telefónica. Claro, *FreePBX* abarca muchas más funciones que mediante su interfaz gráfica, hace fácil y rápido el manejo de *Asterisk* y moldear a la central conforme a los requerimientos dados.

Hay gran similitud en la configuración por *Asterisk* entre *FreePBX* y es claro que deben ser iguales. Lo que se hace en *FreePBX*, se reescribe en *Asterisk*. Usar la GUI de *FreePBX* no significa una configuración distinta a la de *Asterisk*, esta plataforma no es más que un medio de fácil acceso y muy vistoso para que muchos usuarios o administradores tengan facilidades de implementar centrales telefónicas de servicios *VoIP*. Así como sucede en muchas otras aplicaciones que usan interfaces graficas. Al configurar en *FreePBX*, se cree que se está manejando una central con *FreePBX*, pero en realidad se está administrando con *Asterisk* internamente.

3.4. DESARROLLO DEL OBJETIVO ESPECÍFICO 4: Definir los *codecs* y protocolos adecuados para la central telefónica *VoIP*

Cuando la información viaja de un lugar a otro por medio de una red de internet, sea local o global, para que los datos lleguen a su destino, se hacen uso de protocolos y *codecs*. Sin ellos no sería posible entablar una comunicación.

La elección del tipo de *codec* o protocolo dependerá de la aplicación o servicio que se planea usar.

3.4.1. Elección del *Codec*

La definición del *codec* es sencilla. La voz es una señal analógica, y para enviarla por medio de una red digital, es necesaria convertirla a señales digitales. A este proceso o módulo de conversión análogo/digital se le denomina como codificación. Actualmente

existen varios tipos de codificadores o *codecs*, su elección afectará directamente a la calidad de la señal y la velocidad, dado que estos aplican procesos de comprensión y presentan atributos como velocidad binaria, complejidad, retardo y ancho de banda. Cada uno de ellos están estrechamente relacionados, por lo que es muy importante prestarle atención al seleccionar el *codec*.

Para telefonía, los *codecs* más usados son el G711 y el G729, por lo tanto, se centrará en estos dos.

El G711 es un *codec* creado para calidad de servicio en PSTN, ofrece una alta calidad de voz, pero también consume anchos de banda mayor. Maneja los datos a 64 kbps y viene en dos versiones: G711u (*ulaw*) utilizado en USA, y el G711a (*alaw*) para Europa y otros países. Este *codec* es soportado por casi todos los equipos de telefonía y SP (proveedores de servicio). A pesar de alta calidad de voz, no se recomienda usar este codificador si hay limitaciones en el ancho de banda.

Por otro lado, el G729 ofrece una calidad de servicio menor frente a la G711. Por ello, este *codec* es usado en casos donde el ancho de banda es limitado, pues comprime los datos a 8 kbps, teniendo un menor consumo de ancho de banda. Este también presenta dos versiones que es la G729 y G729a, La diferencia en ambos es el ancho de banda y la complejidad (MIPS, millones de instrucciones por segundo). El único inconveniente con este *codec* es que requiere un considerable poder de procesamiento, lo cual no podría ser soportado por todos los equipos (Russoniello, Gil y Samaniego, 2015), además de que a diferencia del G711, el G729 es un *codec* licenciado, lo cual algunos *softphone* de uso libre no incluyen este *codec*.

Cabe mencionar que otro *codec* de los más usados es el iLBC (*Internet Low Bitrate Codec*) pero se descarta para este proyecto. A pesar de que ofrece un nivel de calidad de voz superior y un reducido consumo del ancho de banda frente a G729, para realizar estas

prestaciones consume una gran cantidad de recursos del CPU (aplica algoritmos muy complejos cuando codifica y decodifica) que a larga presentaría problemas al momento de mantener llamadas simultaneas, estresando muy rápido al procesador. En la tabla 23, se muestra entre estos codificadores más empleados.

Tabla 23: Comparación entre Codecs G711 VS G729

Codec	Data rate (kbps)	Bandwidth (kbps)	MOS (Mean Opinion Score)
G.711 a-law	64	87.2	4.1
G.711 u-law	64	87.2	4.1
G.729	8	31.2	3.92
G.729a	8	20	3.7

El valor MOS es un sistema para evaluar las conexiones de voz bajo un enfoque cualitativo.

Habiendo definido ambos *codecs* y visto sus atributos, para el caso del proyecto se emplea el G729, como primera opción, a pesar de que ofrezca calidad menor frente al G711, ocupa un menor ancho de banda, lo cual ayuda en la velocidad y sin comprometer el servicio. Este sería para el caso donde algunas de las empresas pequeñas o medianas presentan limitaciones con el ancho de banda que manejan. Frente a este contexto, se aplica a usar *codecs* que no requieren anchos de banda altos para servicios de comunicación por voz. Sin embargo, si el escenario es manejable y presenta una red dedicada a la comunicación (normalmente a partir de empresas medianas), se utiliza la codificación G711, dando una mejor calidad de voz, mitigando algunos problemas con la pérdida de paquetes.

3.4.2. Elección del protocolo de Señalización

Para *VoIP*, los protocolos de señalización cumplen funciones similares a sus homólogos en la telefonía tradicional, es decir, permiten el intercambio de información, establecimiento de sesión, progreso de las llamadas, etc. Se encuentran en la capa 5 del modelo

OSI (Capa de Sesión). Los más conocidos y usados son SIP e IAX (o IAX2).

Anteriormente ya se ha hablado de estos protocolos, y para entender sus diferencias principales, se resume en la tabla 24:

Tabla 24: Diferencias entre IAX y SIP

Diferencias	SIP	IAX
Ancho de banda	Mayor ancho de banda pues envía mensajes de texto	Codifica los mensajes de forma binaria usando un menor ancho de banda
NAT	En SIP la señalización y los datos viajan por separado, dando origen a problemas frente la conversión de direcciones incompatibles que requieran NAT para hacer que el flujo de audio supere a los <i>Routers</i> o firewalls (SIP suele necesitar un servidor STUN para este problema)	En IAX los datos y la señalización viajan conjuntamente, evitando los problemas con NAT concurrentes en SIP
Estandarización	SIP es un protocolo estandarizado por la IETF, siendo ampliamente implementado por todos los fabricantes de equipos y software	IAX no está aún estandarizado, por lo que no se encuentra mucho en los equipos o dispositivos del mercado
Uso de Puerto	SIP usa un puerto (5060) para señalización y dos puertos RTP por cada conexión de audio	IAX utiliza un mecanismo de multiplexación, por ello usa un solo puerto (4569) para mandar la información de señalización y los datos

Al observar la tabla, se puede deducir que es más ventajoso usar IAX que SIP. Sin embargo, no todo es completamente así. El mayor problema viene a ser la estandarización, IAX al no estar estandarizado totalmente, muchos de los fabricantes no integran este protocolo, por lo que de nada serviría si los dispositivos finales no puedan soportarlo. Por otra parte, la seguridad también se

discute. IAX usa un solo puerto para la señalización y los datos, de esta manera se eliminan los problemas ocurridos con NAT, haciendo a IAX casi transparente a los cortafuegos, por lo que se debe tener en cuenta en caso de tener usuarios con ideas malintencionadas. Para este proyecto, se trabaja en base a SIP, pero no en su totalidad. Si existen ocasiones donde es más ventajoso usar IAX, se aplica este protocolo, puesto que IAX busca minimizar el ancho de banda requerido en la transmisión de voz y video a través de la red.

3.4.3. Elección del Protocolo de Transporte

Cuando se refiere al protocolo de transporte, normalmente se relaciona con UDP y TCP, lo cual es cierto, pero no totalmente. Queda más que claro que para servicios o aplicaciones de voz y video, el rey es UDP, ya que está orientado a la velocidad, que es lo que se requiere en estos servicios. Usar TCP generaría retardos en la transmisión pues este protocolo se orienta a la seguridad y confianza, tomando más tiempo poder transmitir en tiempo real, generando congestiones. Sin embargo, cuando se refiere a protocolos de transporte para voz, aquí se refieren a los protocolos que transportan la voz propiamente dicha o lo que normalmente se denomina carga útil, además de proveer técnicas para evitar problemas como *jitter* o retardo. Los más usados son RTP, RTCP o cRTP, SRTP, etc. Definiciones que se hicieron anteriormente.

Para este caso, se aplica RTP o Protocolo de transporte en tiempo real, ideal para servicios de entrega como voz y video. Es el estándar usado para aquellos datos muy sensibles a retardos que comprometen a la *QoS*.

Asterisk implementa a RTP sobre UDP para la transmisión de voz y sigue sus dichas características. Es decir, no provee mecanismos para asegurar la entrega, pero si técnicas para la velocidad de transmisión, con el fin de evitar retardos, *jitters* y congestión.

3.5. DESARROLLO DEL OBJETIVO ESPECÍFICO 5: Realizar pruebas de estrés a la central telefónica VoIP y medir la eficiencia y el rendimiento

Para medir el rendimiento y potencial al servidor VoIP (*Raspberry Pi*) como central telefónica, se diseña una red para simular el sistema de comunicaciones en una empresa. Se realizó las configuraciones antes descritas, por ambos métodos, estableciendo servicios principalmente como llamadas de voz y video. En la figura 28, se tiene la siguiente topología de red:

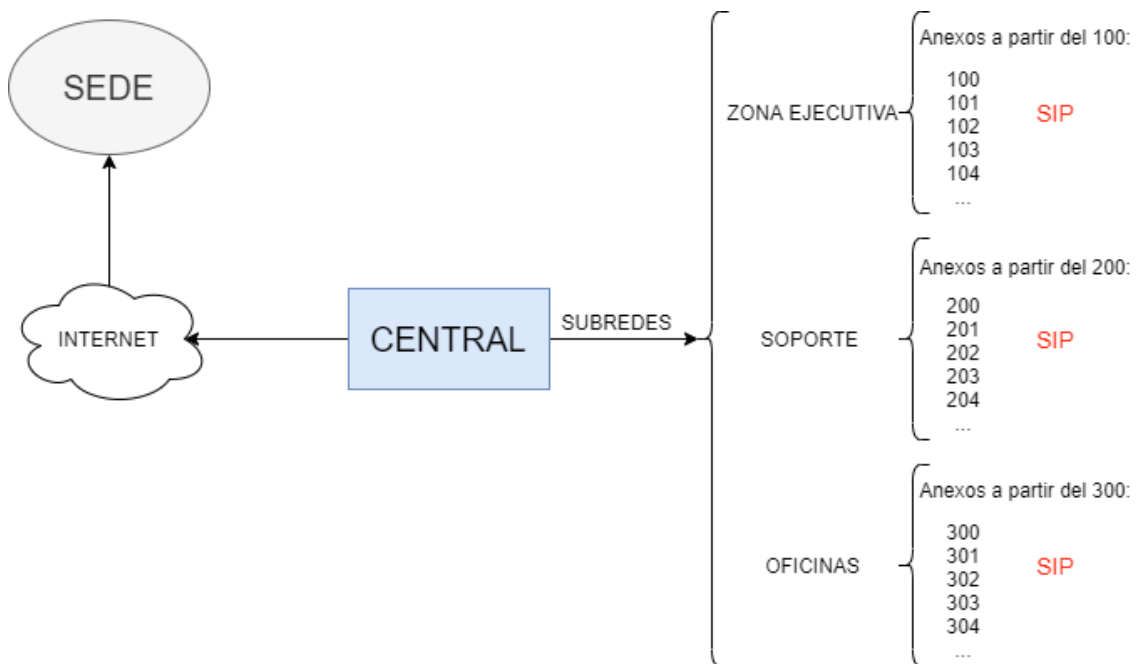


Figura 28: Topología de red física para el sistema de comunicaciones

Para entender la división de la red, se muestra la topología lógica de la red en la fig.29:

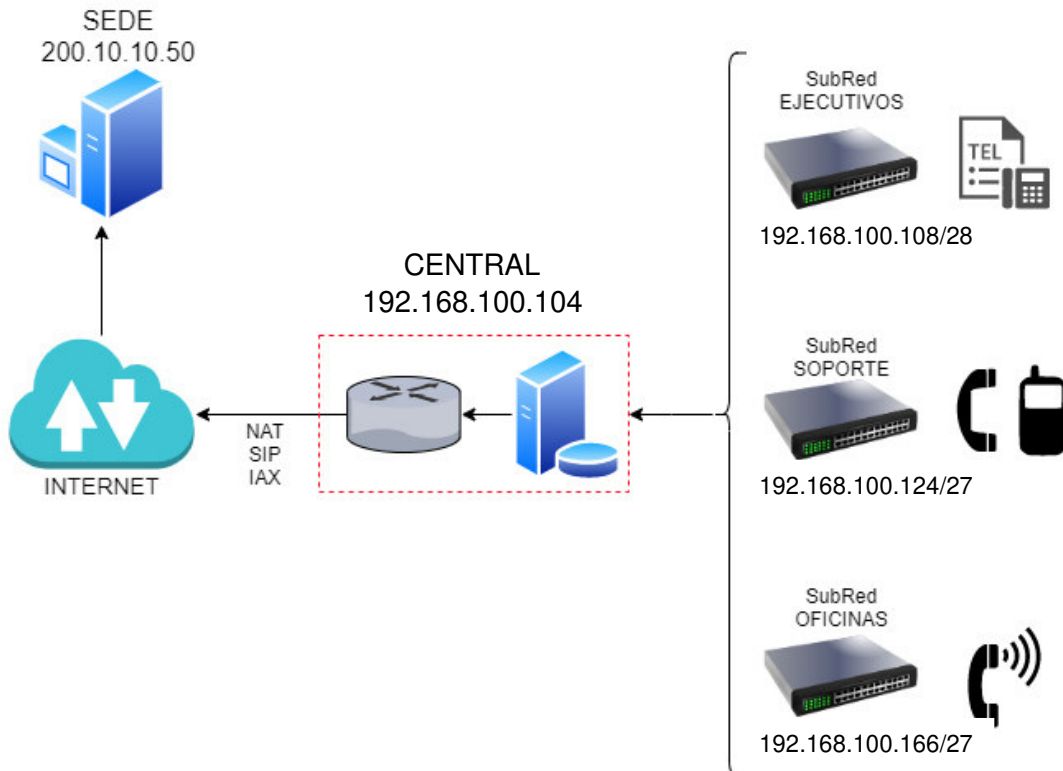


Figura 29: Topología Lógica de la red de comunicaciones

Se crearon tres subredes de la red general, abarcando 30, 20 y 10 usuarios mínimos respectivamente. Para poder enlazar con la sede remotamente, ha de ser necesario NAT para traducir las direcciones privadas a públicas, o en todo caso que la central posea una dirección pública para viajar hacia afuera.

Con la red implementada, se procede a evaluar la central *VoIP*. Como se menciona, se tendrán en cuenta parámetros para la *QoS*, como *jitter* o variación de retardo, latencia, pérdida de paquetes, ancho de banda, número de llamadas simultáneas, etc. El MOS (*Mean Opinion Score*) también es un valor importante, este es una medida de la calidad de voz subjetiva (según la ITU-T P.800) donde el usuario asigna valor de 1 a 5 que define la calificación de la llamada (Ermez, 2014).

Antes de evaluar a la central, se debe definir el número de llamadas simultáneas que puede soportar *Asterisk* en base a los equipos que se tiene. Como la velocidad y el ancho de banda del *Router*. Para ello, el *codec* es importante, y como se mencionó anteriormente, se abarca el G.729.

En la tabla 25, se presenta información útil y necesaria para la determinación de los *codecs*.

Tabla 25: Información de Codecs

Codec y BitRate (Kbps)	Tamaño (Bytes)	Intervalo (ms)	MOS	Payload (Bytes)	Payload (ms)	Paquetes por segundo (PPS)	Ancho de banda MP o FRF.12 (Kbps)	Ancho de banda w/cRTP MP o FRF.12 (Kbps)	Ancho de banda Ethernet (Kbps)
G.711 (64)	80	10	4.1	160	20	50	82.8	67.6	87.2
G.729 (8)	10	10	3.92	20	20	50	26.8	11.6	31.2

Se debe conocer el tamaño de la trama que es igual:

$$TRAMA = Payload + Capa 2 + Capa 3 + Capa 4 + encabezado de capa de enlace = 86 \text{ bytes}$$

Un byte equivale a 8 Bits, por tanto, se tiene 866 bits por trama. El G.729 tiene 50 tramas por segundo, por ello:

$$BW \text{ requerido} = 866 \frac{\text{bits}}{\text{trama}} * \frac{50 \text{ tramas}}{\text{segundos}} = 34400 \frac{\text{bits}}{\text{seg}} \approx 34 \text{ Kbps}$$

Con las características del *Router* TP-Link 10/100 Mbps

$$\# \text{ máximo de llamadas simultáneas} = \frac{100 \text{ Mbps}}{34 \text{ Kbps}} = 2941$$

Teóricamente, si se usara la capacidad total del *Router* y del ordenador (central *VoIP*), con un alto rendimiento, 2941 llamadas simultaneas pueden ser realizadas. En un caso real, alcanzaría un total 2900 aproximadamente, teniendo en cuenta los factores que afectan a la *QoS*. Cabe mencionar que si se empleara el *codec* G.711, el número de llamadas llegaría alrededor de 1200 o más. Debido que este *codec* consume un mayor ancho de banda. Sin embargo, la cuestión es si la pequeña *Raspberry Pi* soportaría dicha cantidad. Lo cual es poco probable.

Esto no significa que la *Raspberry Pi* no sea útil para una central. De hecho, computadoras de rendimiento decente tampoco llegan a procesar tal

cantidad, además de que no es muy probable que en una empresa pequeña o mediana lleguen a necesitar realizar más 1000 llamadas simultaneas. Aquí lo importante es definir cuantas llamadas simultaneas puede soportar la *Raspberry Pi*.

3.5.1. Eficiencia y Rendimiento de la *Raspberry Pi* como central VoIP

Se debe aclarar que la cantidad de llamadas simultaneas que soporta un servidor o equipo, no determina el número de usuarios que debe tener (Por ejemplo, la central puede tener 100 usuarios y soportar 35 llamadas simultaneas). Sabiendo esto, a continuación, se describe en la tabla 26 de *Asterisk* con los requerimientos mínimos de un equipo para una cierta cantidad de llamadas simultaneas. Estos valores ideales mas no definitivos.

Tabla 26: Requerimientos para ciertas cantidades de llamadas simultaneas

Área	Número de Llamadas o Canales	Mínimo Recomendado
Solo pruebas	Máximo 5	400 Mhz x86, 256 MB de RAM
SOHO (pequeña oficina u oficina en casa)	5 a 10	1 Ghz x86 512 MB de RAM
Pequeña o mediana empresa	No más de 25	1,2 Ghz x86 1 GB de RAM
Sistema mediano o grande	Mas de 25	Dual Cores

Comparando la tabla con las características de la *Raspberry Pi* utilizada para la central (ARMv8 y 1 GB RAM), se puede determinar que es más que suficiente para escenarios de pequeña o mediana empresa. Estos valores de la tabla (tabla 26) no definen la cantidad máxima de llamadas que soporta la *Raspberry*, solo son escenarios estándar.

Para determinar esta cantidad soportada de un modo práctico, se hace uso de un *software* llamado *SIPP*.

SIPP es un *software* libre cuyo funcionamiento es de generar tráfico SIP, con dos modos de trabajo: *UAC* y *UAS* (Cliente y Servidor). Este *software* ayudara a generar el tráfico necesario para medir el rendimiento de la *Raspberry Pi* para simular los escenarios que se desean.

Existen varias maneras de instalar *SIPP*, la más sencilla es desde la terminal de Linux: ***apt-get install sip-tester***

Lo siguiente es crear una extensión en el *sip.conf* para hacer pruebas de llamadas en el servidor *Asterisk*.

```
[SIPP]
Type=friend
Context=SIPP
Host=Dynamic
Insecure=INVITE, port
User=SIPP
Canreinvite=no
Disallow=all
Allow=alaw
Allow=ulaw
Allow=g729
```

Lo siguiente es configurar el *extensions.conf* para agregar la extensión (1234) para el tráfico *SIPP*. La llamada recibida es contestada y el usuario que llamo escucha la música por defecto mientras espera 30 segundos (*SetMusicHoldOn(default)*).

```
[SIPP]
exten => 1234,1, Answer
exten => 1234, n,SetMusicOnHold(default)
exten => 1234, n,WaitMusicOnHold(30)
exten => 1234, n, Hangup
```

Finalmente se ejecutan los escenarios, donde bien pueden ejecutarse archivos *.yml* o las líneas de código, como el siguiente:

SIPP -s 1234 -sn UAC -d 1000 -m 30 -timeout 60 -r 10 -l 192.168.0.101 192.168.0.103 -trace_screen

Donde:

- **-s** indica la extensión a la que se va a llamar (1234)
- **-sn** indica el escenario que se va a utilizar (*UAC*, User Agent Client)
- **-d** indica la duración de la llamada en milisegundos
- **-m** es el número máximo de llamadas que se van a realizar
- **-timeout** indica el tiempo total de la prueba de tráfico
- **-r** determina el número de intentos de llamadas por segundo
- **-i** indica la IP local, del servidor o por defecto del cliente *SIPP*
- **-trace_screen** establece guardar los logs que corresponden a la información en pantalla

Para medir el rendimiento y hacer comparaciones de la *Raspberry Pi*, se simularon varios escenarios, sobre todo se cambiaron los valores de los intentos de llamadas (-r) y de la cantidad máxima de llamadas (-m).

3.5.2. Análisis de rendimiento con *SIPP*

Lo primero será analizar la variación del uso del CPU con *Asterisk* frente a cierta cantidad de llamadas simultáneas. Para ello se ejecutan escenarios variando este valor. La tabla 27 sintetiza el uso frente a los escenarios.

Tabla 27: Uso del CPU y memoria frente a distintas cantidades de llamadas simultáneas

Llamadas Simultáneas	Rendimiento CPU
5	6.5%~8%
20	24.6%~27%
40	44.2%~47%
50	53.3%~55%
60	63.9%~74%
80	81.2%~93%
100	100%~117%

Al observar la tabla, puede notarse cierta proporcionalidad del número de llamadas frente al porcentaje del uso del CPU mientras se ejecuta la simulación. Desde 5 hasta 50 llamadas simultaneas, la variación del rango es aproximadamente de 2 a 3 por ciento. Durante este proceso, la cantidad de llamadas exitosas han sido el 100% total, para estos 4 primeros escenarios (5, 20, 40 y 50 llamadas simultaneas).

A partir de 60 a 100 llamadas, es cuando la *Raspberry* comienza a tener algunas dificultades para procesar tal cantidad. Se refleja en la tabla un mayor rango de variación frente a los escenarios de 60, 80 y 100 llamadas. Con más del 10% debido al número de llamadas exitosas o fallidas que aumentaban o disminuían, haciendo uso variable del CPU, inclusive llegar a obtener un 110% o más del uso de *Raspberry* (CPU o *Asterisk*).

Tal y como se observa en las siguientes figuras:

```

Incoming call created |          0          |          0
OutGoing call created |          0          |          5
Total Call created   |          0          |          5
Current Call         |          0          |
-----+-----+-----
Successful call      |          0          |          5
Failed call          |          0          |          0
1 [|                0.6%] Tasks: 65, 148 thr; 1 running
2 [|                0.0%] Load average: 0.19 0.08 0.03
3 [|                0.7%] Uptime: 02:20:02
4 [|                3.9%]
Mem[|||||||         171M/873M]
Swp[                0K/100.0M]
PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
439 asterisk  -11   0  172M 53340 27452 S  6.5  6.0  2:20.62 /usr/sbin/as

```

Figura 30: Escenario 5 llamadas simultaneas

Escenario con 5 llamadas exitosas y un uso de CPU de 6.5 por ciento respectivos

```

Incoming call created |          0          |          0
OutGoing call created |          0          |         20
Total Call created   |          0          |         20
Current Call         |          0          |          0
-----+-----+-----
Successful call      |          0          |         20
Failed call          |          0          |          0

1  [|||]          1.4%]   Tasks: 65, 162 thr; 1 running
2  [||]          1.4%]   Load average: 0.12 0.09 0.03
3  [||]          2.7%]   Uptime: 02:32:02
4  [||]          4.0%]
Mem[|||||]      171M/873M]
Swp[            0K/100.0M]

PID USER      PRI  NI  VIRT   RES   SHR S  CPU% MEM%   TIME+  Command
439 asterisk -11   0  172M 53340 27452 S 25.9  6.0  2:29.79 /usr/sbin/asterisk

```

Figura 31: Escenario 20 llamadas simultaneas

Escenario con 20 llamadas exitosas y un uso de CPU de 25.9 porciento respectivos

```

Incoming call created |          0          |          0
OutGoing call created |          0          |         40
Total Call created   |          0          |         40
Current Call         |          0          |          0
-----+-----+-----
Successful call      |          0          |         40
Failed call          |          0          |          0

1  [||||]        5.6%]   Tasks: 65, 175 thr; 1 running
2  [||]         1.6%]   Load average: 0.10 0.13 0.05
3  [||]         0.7%]   Uptime: 01:20:33
4  [||]         1.4%]
Mem[|||||]      168M/873M]
Swp[            0K/100.0M]

PID USER      PRI  NI  VIRT   RES   SHR S  CPU% MEM%   TIME+  Command
436 asterisk -11   0  173M 55100 27404 S 45.0  6.2  2:02.84 /usr/sbin/asterisk

```

Figura 32: Escenario 40 llamadas simultaneas

Escenario con 40 llamadas exitosas y un uso de CPU del 45% respectivos

```

Incoming call created |           0           |           0
OutGoing call created |           0           |           50
Total Call created   |           0           |           50
Current Call         |           0           |
-----+-----+-----
Successful call      |           0           |           50
Failed call          |           0           |           0

 1  [|||]           7.2%]   Tasks: 65, 186 thr; 1 running
 2  [||]           3.0%]   Load average: 0.59 0.17 0.07
 3  [|]            2.1%]   Uptime: 01:32:33
 4  [|]            0.0%]
Mem[|||||]         171M/873M]
Swp[|]             0K/100.0M]

PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
436 asterisk  -11   0  177M 57944 27404 S 53.1  6.5  2:58.37 /usr/sbin/asteri

```

Figura 33: Escenario 50 llamadas simultaneas

Escenario con 50 llamadas exitosas y un uso de CPU del 53.1% respectivos

```

Incoming call created |           0           |           0
OutGoing call created |           0           |           60
Total Call created   |           0           |           60
Current Call         |           0           |
-----+-----+-----
Successful call      |           0           |           57
Failed call          |           0           |           3

 1  [|]            0.8%]   Tasks: 67, 198 thr; 1 running
 2  [|||]          6.6%]   Load average: 0.02 0.05 0.07
 3  [|]            0.8%]   Uptime: 02:19:07
 4  [|]            0.7%]
Mem[|||||]         179M/873M]
Swp[|]             0K/100.0M]

PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
436 asterisk  -11   0  191M 62008 27408 S 72.7  6.9  5:42.45 /usr/sbin/asteri

```

Figura 34: Escenario 60 llamadas exitosas

Escenarios con 57 llamadas exitosas y 3 fallidas, con un uso de CPU de 72.7% respectivos

```

Incoming call created |          0          |          0
OutGoing call created |          0          |          80
Total Call created   |          0          |          80
Current Call         |          0          |          0
-----+-----+-----+
Successful call      |          0          |          79
Failed call         |          0          |          1
1 [|||||]           10.5%] Tasks: 66, 197 thr; 1 running
2 [|||||]           9.2%] Load average: 0.50 0.18 0.11
3 [|||||]           6.9%] Uptime: 00:47:40
4 [|||||]           9.0%]
Mem[|||||||]       175M/873M]
Swp[                0K/100.0M]
PID USER      PRI  NI  VIRT  RES  SHR  S  CPU% MEM%  TIME+  Command
435 asterisk -11   0  179M 57356 27304 S 85.4 6.4  1:04.32 /usr/sbin/asteri

```

Figura 35: Escenario 80 llamadas simultaneas

Escenarios con 79 llamadas exitosas y 1 fallida, con un uso de CPU de 85.4% respectivos

```

Incoming call created |          0          |          0
OutGoing call created |          0          |          100
Total Call created   |          0          |          100
Current Call         |          0          |          0
-----+-----+-----+
Successful call      |          0          |          98
Failed call         |          0          |          2
1 [||]             2.4%] Tasks: 67, 211 thr; 1 running
2 [||]             2.4%] Load average: 0.10 0.18 0.27
3 [|||||]          9.4%] Uptime: 00:40:01
4 [||]             3.8%]
Mem[|||||||]       198M/873M]
Swp[                0K/100.0M]
PID USER      PRI  NI  VIRT  RES  SHR  S  CPU% MEM%  TIME+  Command
431 asterisk -11   0  194M 62832 27344 S 102. 7.0  5:52.55 /usr/sbin/asteri

```

Figura 36: Escenario 100 llamadas simultaneas

Escenario con 98 llamadas exitosas y 2 fallidas, con un uso de CPU de 102% respectivos

Como se observa, el uso del CPU es en cierta forma proporcional a la cantidad de llamadas que se ejecutan en cada escenario. Al comparar los resultados, se puede asegurar que la *Raspberry Pi* puede manejar de 50 a 60 llamadas en simultaneo sin ninguna perdida y un uso de memoria y sistema estable. Al incrementar el número, el servidor telefónico empezara a estresarse poco a poco, que en consecuencia algunas llamadas se pierden y el uso del CPU puede aumentar muy significativamente, afectando otros procesos del sistema, que, a su vez, generarían fallas a la central *VoIP*. Claro está que 100 llamadas no es limite que se puedan simular. Con *SIPP* no hay restricción alguna. Dependerá de los escenarios que se quieran establecer.

Para este caso, se simularon escenarios de incluso 300, 500, 1000 o 3000 llamadas en simultaneo. Por supuesto la cantidad de llamadas fallidas incrementaba grandemente, tal y como se observa en la figura 37 y 38.

```

Incoming call created |          0          |          0
OutGoing call created |          0          |         500
Total Call created   |          0          |         500
Current Call         |          0          |
-----+-----+-----
Successful call      |          0          |         406
Failed call          |          0          |          94

1 [|||||]          30.8%] Tasks: 68, 244 thr; 2 running
2 [|||||]          31.8%] Load average: 0.34 0.43 0.29
3 [|||||]          26.8%] Uptime: 00:15:14
4 [|||||]          21.4%]
Mem [|||||]        194M/873M]
Swp [|||||]         0K/100.0M]

PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
433 asterisk -11   0  208M 71732 27492 S 204.  8.0  4:59.78 /usr/sbin/asteri

```

Figura 37: Escenario 500 llamadas simultaneas

Escenario con 406 llamadas exitosas y 94 fallidas, con un uso de CPU de 204% respectivos

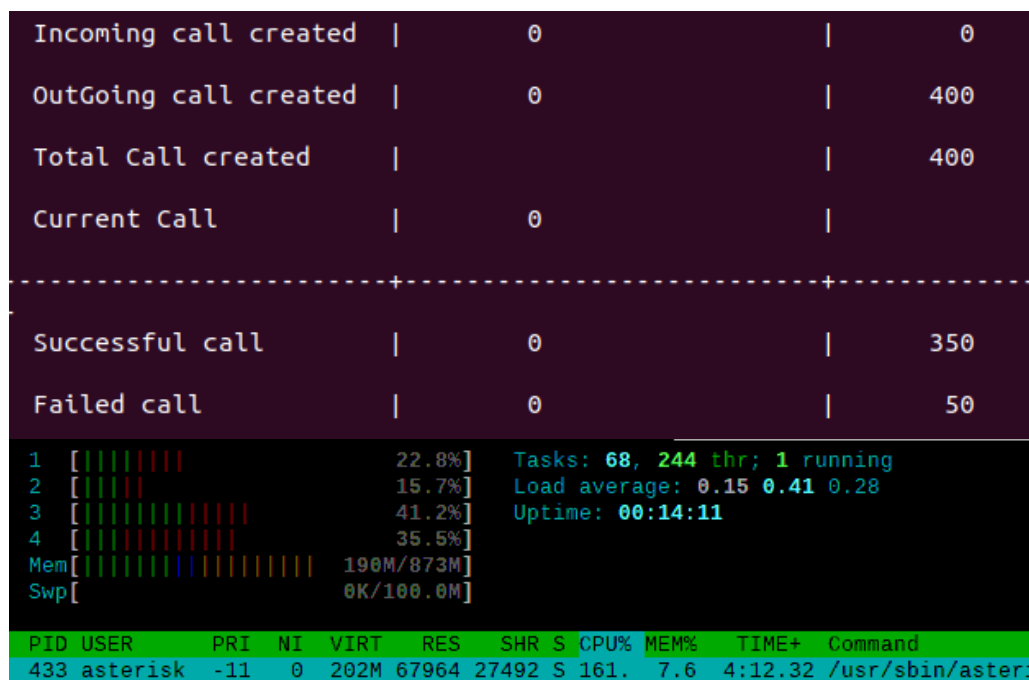


Figura 38: Escenario 400 llamadas simultaneas

Escenario con 350 llamadas exitosas 350 llamadas exitosas y 50 fallidas, con un uso de CPU de 161% respectivos. Se puede notar el aumento del uso de los 4 núcleos del procesador de la central.

A partir de 60 a 100 llamadas en simultaneo, la cantidad de perdidas es relativamente baja (de uno a cinco). Así como al ejecutar un escenario de 100 llamadas, se obtiene el uso límite del CPU de la central *VoIP*. Al pasar el 100%, la cantidad de llamadas fallidas o perdidas aumentara exponencialmente.

3.5.3. Análisis de una llamada real

Para obtener un análisis detallado, la siguiente prueba se realiza con una llamada real. Para capturar los paquetes, se emplea *Wireshark* en la computadora con el *softphone* conectado. Para ello se utilizan los siguientes usuarios:

En el ***sip.conf***:

[100]	[101]
type=friend	type=friend
context=sipp	context=sipp
host=dynamic	host=dynamic
user=100	user=101
secret=1234	secret=1234
disallow=all	disallow=all
allow=ulaw (o g729)	allow=ulaw (o g729)

En ***extensions.conf*** se agrega:

```
exten => 100,1,Dial(SIP/100)
exten => 100,n,Hangup
exten => 101,1,Dial(SIP/101)
exten => 101,n,Hangup
```

Se analiza una llamada real entre la computadora con *Wireshark* y un *softphone* en un móvil Android, con el objetivo de diferenciar y comprender el trabajo de los *codecs* y los protocolos.

En primera instancia, capturando solo los paquetes SIP, que es el protocolo de inicio de sesión. La secuencia que se realiza es la iniciación de la llamada, la respuesta del destino, y la finalización del evento.

Time	Source	Destination	Protocol	Length	Info
2262	148.957740	192.168.100.104	192.168.100.96	SIP/SDP	919 Request: INVITE sip:100@192.168.100.96:9205
2263	148.973119	192.168.100.96	192.168.100.104	SIP	394 Status: 180 Ringing
2297	160.045397	192.168.100.96	192.168.100.104	SIP/SDP	741 Status: 200 OK
2298	160.053750	192.168.100.104	192.168.100.96	SIP	454 Request: ACK sip:100@192.168.100.96:9205
2299	160.056397	192.168.100.104	192.168.100.96	SIP/SDP	894 Request: INVITE sip:100@192.168.100.96:9205, in-dialog
2301	160.057890	192.168.100.96	192.168.100.104	SIP/SDP	741 Status: 200 OK
2302	160.063523	192.168.100.104	192.168.100.96	SIP	454 Request: ACK sip:100@192.168.100.96:9205
3308	174.341056	192.168.100.104	192.168.100.96	SIP/SDP	895 Request: INVITE sip:100@192.168.100.96:9205, in-dialog
3310	174.342269	192.168.100.96	192.168.100.104	SIP/SDP	741 Status: 200 OK
3311	174.352730	192.168.100.104	192.168.100.96	SIP	454 Request: ACK sip:100@192.168.100.96:9205
3312	174.352730	192.168.100.104	192.168.100.96	SIP	486 Request: BYE sip:100@192.168.100.96:9205
3314	174.354058	192.168.100.96	192.168.100.104	SIP	386 Status: 200 OK

```

Message Header
  > Via: SIP/2.0/UDP 192.168.100.104:5060;branch=z9hG4bK28dd8f2e
    Max-Forwards: 70
  > From: "101" <sip:101@192.168.100.104>;tag=as082b43b4
  > To: <sip:100@192.168.100.96:9205>
  > Contact: <sip:101@192.168.100.104:5060>
    Call-ID: 70ec16230df2869110853b3565647c48@192.168.100.104:5060
    [Generated Call-ID: 70ec16230df2869110853b3565647c48@192.168.100.104:5060]
  > CSeq: 102 INVITE
    User-Agent: Asterisk PBX 16.2.1~dfsg-1+deb10u2
    Date: Thu, 13 May 2021 16:11:22 GMT
    Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE
    Supported: replaces, timer
    Content-Type: application/sdp
    Content-Length: 282
Message Body
  > Session Description Protocol
    Session Description Protocol Version (v): 0
    > Owner/Creator, Session Id (o): root 922283291 922283291 IN IP4 192.168.100.104
    Session Name (s): Asterisk PBX 16.2.1~dfsg-1+deb10u2
    > Connection Information (c): IN IP4 192.168.100.104

```

Figura 39: Captura de tráfico SIP en una llamada con Wireshark

La figura 39 muestra el proceso de la llamada, de inicio a fin de la sesión. Sin embargo, para analizar el tráfico, es necesario explicar los términos que se muestran en los paquetes capturados.

En el apartado de protocolo, en distintos paquetes, se observan de dos tipos: SIP y SDP. SDP es el protocolo de descripción de sesión y viaja en conjunto de SIP, que es el protocolo de inicio de sesión.

SIP contiene la información principal y se encarga de entablar la comunicación entre usuarios, SDP transporta información parcialmente secundaria (como lo indica su nombre, descripción del cuerpo del paquete como usuario, ID, versión y nombre de la sesión, tipo de red y dirección, IP, etc.), dependientemente de SIP. Sintetizando, ambos protocolos funcionan en combinación para establecer una llamada. Para ello SIP utiliza mensajes y términos numéricos que describen el estado de la comunicación, a estos se les da el nombre de solicitud y respuesta. Hay catorce métodos de peticiones (o solicitudes) SIP: **INVITE**, que establece una sesión; **ACK**, confirma una solicitud; **CANCEL**, cancela la sesión establecida; **OPTIONS**, solicita sobre las capacidades de envío y recepción del servidor o teléfono SIP; **BYE**, que finaliza la sesión; **REFER**, para

solicitar una transferencia de llamada; **SUBSCRIBE**, para recibir notificaciones; **NOTIFY**, da avisos acerca de nuevos eventos al suscriptor; **INFO**, comunica información de la sesión; **PUBLISH**, publica algún evento en el servidor; **MESSAGE**, envía mensajes instantáneos; **REGISTER**, registra la dirección, equipo o IP de un usuario; **PRACK**, acuse de recibo temporal y **UPDATE**, que actualiza o modifica el estado de la sesión.

Estas solicitudes reciben de forma numérica sus respuestas. De 100 a 199 significa una respuesta informativa (como **100 Trying** o **180 Ringing**), de 200 a 299 son respuestas de éxito (**200 ACK**), para 300 a 399 respuestas de redireccionamiento. De 400 a 499, 500 a 599 y 600 a 699, son errores de solicitud, servidor y globales correspondientemente. Ambos son un conjunto que SIP utiliza para establecer la llamada, a su vez SDP incluye descripción del cuerpo.

Luego de iniciada la sesión, el protocolo RTP se encarga de transportar los datos o media (voz o video) en tiempo real sobre UDP, que es usado para mayor velocidad como se menciona anteriormente, como se observa las figuras 40 y 41.

Time	Source	Destination	Protocol	Length	Info
2985	192.168.100.110	192.168.100.96	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x778AB2F9, Seq=1644, Time=4166067190
2986	192.168.100.96	192.168.100.110	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x22CE, Seq=1310, Time=304640
2987	192.168.100.110	192.168.100.96	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x778AB2F9, Seq=1645, Time=4166067350
2988	192.168.100.96	192.168.100.110	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x22CE, Seq=1311, Time=304800
2989	192.168.100.96	192.168.100.110	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x22CE, Seq=1312, Time=304960
2990	192.168.100.110	192.168.100.96	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x778AB2F9, Seq=1646, Time=4166067510
2991	192.168.100.110	192.168.100.96	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x778AB2F9, Seq=1647, Time=4166067670
2992	192.168.100.96	192.168.100.110	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x22CE, Seq=1313, Time=305120
2993	192.168.100.110	192.168.100.96	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x778AB2F9, Seq=1648, Time=4166067830
2997	192.168.100.96	192.168.100.104	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x22CE, Seq=1314, Time=305280

```

User Datagram Protocol, Src Port: 9206, Dst Port: 7078
  Source Port: 9206
  Destination Port: 7078
  Length: 180
  Checksum: 0x52f0 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 2]
  > [Timestamps]
  UDP payload (172 bytes)
Real-Time Transport Protocol
  > [Stream setup by SDP (frame 14)]
  10.. .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0. .... = Extension: False
  ...0000 = Contributing source identifiers count: 0
  0... .... = Marker: False
  Payload type: ITU-T G.711 PCMU (0)
  Sequence number: 46
  [Extended sequence number: 65582]
  Timestamp: 36960
  Synchronization Source identifier: 0x000022ce (8910)
  Payload: bdc6cfd3667f24e4551e4727cdfd7c9d5dfc7bdc9e7d9e64e413f47453a43e3cfc7b9b8..
  
```

Figura 40: Captura de tráfico RTP (G711u) en una llamada con Wireshark

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

rtsp

No.	Time	Source	Destination	Protocol	Length	Info
23068	780.946202	192.168.100.110	192.168.100.96	RTP	74	PT=ITU-T G.729, SSRC=0x546744D3, Seq=3105, Time=3124071587
23069	780.961260	192.168.100.96	192.168.100.110	RTP	74	PT=ITU-T G.729, SSRC=0x53A7, Seq=9465, Time=539360
23070	780.976519	192.168.100.110	192.168.100.96	RTP	74	PT=ITU-T G.729, SSRC=0x546744D3, Seq=3106, Time=3124071747
23071	780.976519	192.168.100.110	192.168.100.96	RTP	74	PT=ITU-T G.729, SSRC=0x546744D3, Seq=3107, Time=3124071907
23072	780.982456	192.168.100.96	192.168.100.110	RTP	74	PT=ITU-T G.729, SSRC=0x53A7, Seq=9466, Time=539520
23073	780.994370	192.168.100.110	192.168.100.96	RTP	74	PT=ITU-T G.729, SSRC=0x546744D3, Seq=3108, Time=3124072067
23074	781.001044	192.168.100.96	192.168.100.110	RTP	74	PT=ITU-T G.729, SSRC=0x53A7, Seq=9467, Time=539680
23075	781.016820	192.168.100.110	192.168.100.96	RTP	74	PT=ITU-T G.729, SSRC=0x546744D3, Seq=3109, Time=3124072227
23076	781.020011	192.168.100.96	192.168.100.110	RTP	74	PT=ITU-T G.729, SSRC=0x53A7, Seq=9468, Time=539840
23080	781.041109	192.168.100.96	192.168.100.104	RTP	74	PT=ITU-T G.729, SSRC=0x53A7, Seq=9469, Time=540000

User Datagram Protocol, Src Port: 9206, Dst Port: 7078
 Source Port: 9206
 Destination Port: 7078
 Length: 40
 Checksum: 0xb390 [unverified]
 [Checksum Status: Unverified]
 [Stream index: 108]
 [Timestamps]
 UDP payload (32 bytes)

Real-Time Transport Protocol
 [Stream setup by SDP (frame 16733)]
 10.. = Version: RFC 1889 Version (2)
 ..0. = Padding: False
 ...0 = Extension: False
 0000 = Contributing source identifiers count: 0
 0... = Marker: False
 Payload type: ITU-T G.729 (18)
 Sequence number: 9468
 [Extended sequence number: 75004]
 Timestamp: 539840
 Synchronization Source identifier: 0x000053a7 (21415)
 Payload: 649446261e5fb22cad6e4491d9d127ad94a48969

Figura 41: Captura de tráfico RTP (G729) en una llamada con Wireshark

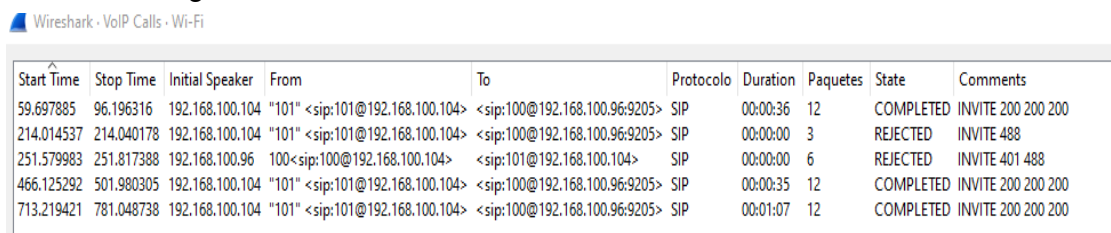
Dentro de estos paquetes o campos, la información que muestra son las propiedades del *codec* utilizado y los puertos primordialmente. En el caso de la fig. 40, el tipo de carga útil (*Payload type*) mostrado es: ITU-T G.711 PCMU (0), en términos sencillos, utiliza el *codec* G.711, estandarizado por la ITU y PCMU (o *ulaw*) como algoritmo para la compresión y descompresión. A diferencia de la fig.41, el tipo de carga útil muestra ITU-T G.729 (16), que utiliza el *codec* G.729 con 18 KHz de frecuencia de muestreo.

Es importante resaltar el detalle de las direcciones de origen y destino. Cuando se observa la captura de SIP (fig.39), las direcciones de origen y destino (y viceversa) de los paquetes al viajar son 192.168.100.104 (IP del servidor) y 192.168.100.96 (PC con *softphone* y *Wireshark*). Indicando que, para establecer la llamada, la PC se comunica con el servidor y este le responde para averiguar si el cliente destino está disponible. De la misma forma, el servidor contacta al otro *softphone*, enviando mensajes entre usuarios y servidor. Esto se analiza a detalle más adelante.

Por otra parte, en la captura de RTP, expone solo las direcciones IP de los *softphones*. (192.168.100.110, IP del otro *softphone*). Como se menciona, RTP transporta los datos o media (voz, video o texto) y la comunicación se realiza directamente entre los usuarios, sin necesidad de pasar por el servidor, pues este ya estableció la sesión. Solo se comunicaría con la central, si quiere acabar o transferir la llamada.

Esto son solo las capturas que entra y sale por la PC con *Wireshark*, para este caso, no hay necesidad de mostrar todo el tráfico que viaje por la red local.

Por supuesto, estas capturas solo muestran información descriptiva (y realmente útil). Para un análisis más detallado, *Wireshark* contiene herramientas dedicadas al tráfico *VoIP*. Se puede registrar o capturar las llamadas y capturar la voz o sonido que viaja por la red. Comparar retardos y analizar el flujo de llamada. A continuación, se presentan algunas llamadas, fallidas y exitosas, con *codec* G711 y G729 en la figura 42.



Start Time	Stop Time	Initial Speaker	From	To	Protocolo	Duration	Paquetes	State	Comments
59.697885	96.196316	192.168.100.104	"101" <sip:101@192.168.100.104>	<sip:100@192.168.100.96:9205>	SIP	00:00:36	12	COMPLETED	INVITE 200 200 200
214.014537	214.040178	192.168.100.104	"101" <sip:101@192.168.100.104>	<sip:100@192.168.100.96:9205>	SIP	00:00:00	3	REJECTED	INVITE 488
251.579983	251.817388	192.168.100.96	100<sip:100@192.168.100.104>	<sip:101@192.168.100.104>	SIP	00:00:00	6	REJECTED	INVITE 401 488
466.125292	501.980305	192.168.100.104	"101" <sip:101@192.168.100.104>	<sip:100@192.168.100.96:9205>	SIP	00:00:35	12	COMPLETED	INVITE 200 200 200
713.219421	781.048738	192.168.100.104	"101" <sip:101@192.168.100.104>	<sip:100@192.168.100.96:9205>	SIP	00:01:07	12	COMPLETED	INVITE 200 200 200

Figura 42: Llamadas fallidas y exitosas

En el apartado se observa las llamadas y su protocolo (SIP), junto a sus mensajes de *INVITE*, donde 3 llamadas no tuvieron problema alguno (200, que es éxito) y dos fallidas, en este caso rechazadas (de 400 a 499 son fallos de solicitud).

En las figuras 43 y 44 se muestran las capturas de espectro del audio.

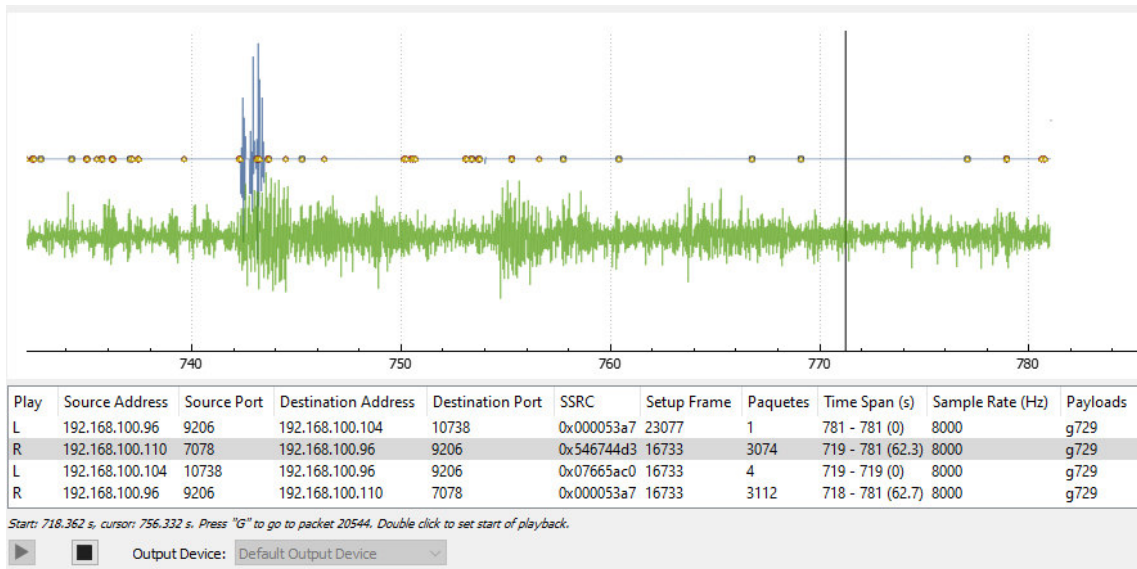


Figura 43: Audio capturado con codec G729

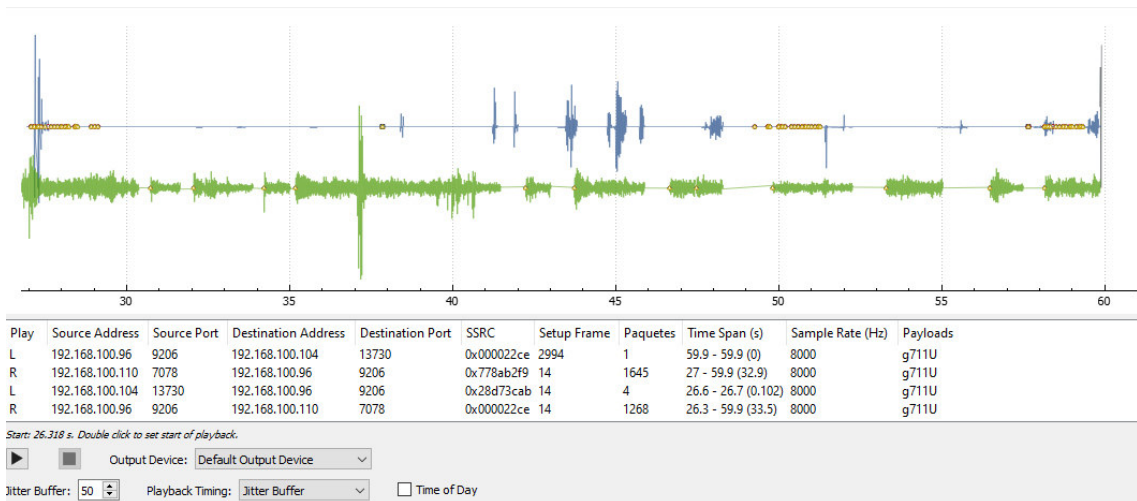


Figura 44: Audio capturado con codec G711 ulaw

Si comparamos ambas imágenes, la clara diferencia se observa en la gráfica o el espectro de audio. La de G711 se ve más limpia y ordenada que la G729, además de la cantidad de paquetes que viajaron en la duración de la llamada. Al reproducir el audio, la calidad del G711 es superior a la G729. Confirmando lo mencionado anteriormente en las diferencias de *codec*.

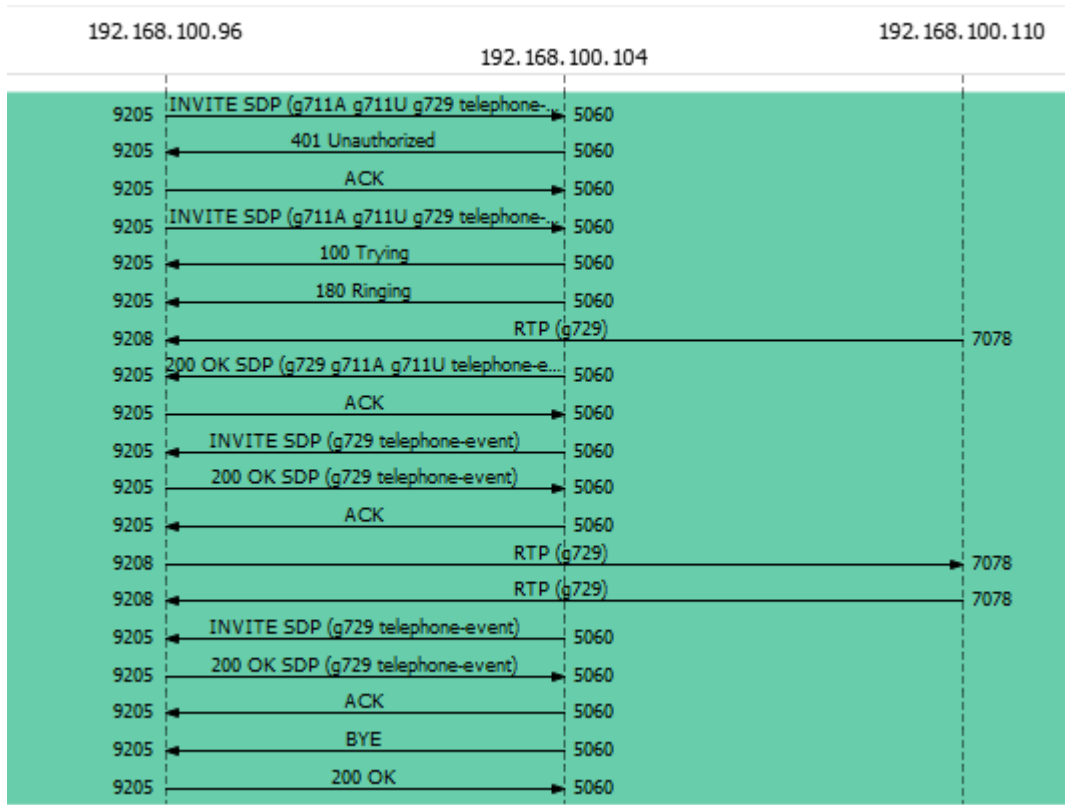


Figura 45: Flujo de llamada

En la figura 45 se muestra el flujo de llamada entre los dos *softphones*. 192.168.100.96 (PC con *Wireshark*) inicia la llamada y se dirige a 192.168.100.104 (central) con puerto 5060 (puerto SIP). Envía un *INVITE* con la información del usuario y los *codecs* que está usando. La central le responde con un 401 no autorizado, probablemente a que el servidor no recibió respuesta por parte de 192.168.100.110 (móvil). 192.168.100.96 le envía un *ACK* confirmando la respuesta. Inicia otra vez con un *INVITE* y la central le comunica que lo intenta (100 *Trying*) y que está timbrando (180 *Ringing*). 192.168.100.110 responde y empieza enviar sus paquetes RTP (media). El servidor le comunica a 192.168.100.96 que 192.168.100.110 acepto con éxito su invitación (200 *OK*) y este le confirma (*ACK*). La central le indica con un *INVITE* que la llamada se realiza con el *codec* g729, 192.168.100.96 acepta con éxito el *codec*, y central confirma la respuesta (*ACK*). Comienza el intercambio de datos o media entre los usuarios, con g729 como algoritmo de comprensión y descompresión. Finalmente, la central le envía un

INVITE, probablemente 192.168.100.110 quiere finalizar la llamada. 192.168.100.96, acepta con éxito y el servidor confirma (*ACK*), finalizando la sesión (*BYE*), a lo que 192.168.100.96 acepta y confirma (200 *OK*).

Como siguiente ejemplo, se configuraron a los usuarios con un distinto *codec*, donde como escenario el *softphone* móvil solo aceptara llamadas con *codec* G711 (figura 46).

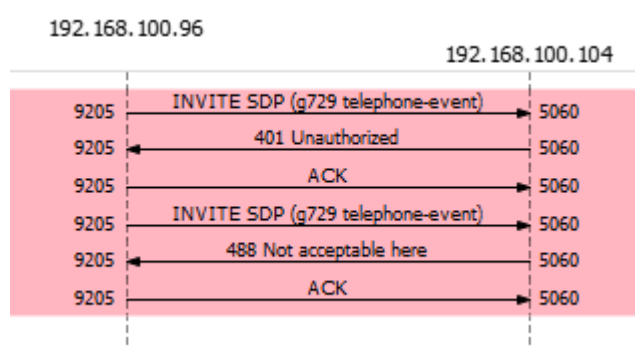


Figura 46: Flujo de llamada fallida

Para este escenario, 192.168.100.96 inicia la llamada enviando un *INVITE* con la descripción de la sesión e indicando que está usando g729 como *codec*. La central responde que no tiene autorización para iniciar la sesión (401). 192.168.100.96 confirma (*ACK*) e inicia otra vez la sesión con un *INVITE*. El servidor le responde finalmente que no es aceptable y no puede establecer la llamada (488). Sin éxito, 196.168.100.96 confirma y termina su sesión.

Así mismo, se puede analizar la cantidad de paquetes perdidos y el *jitter* como se observa en la figura 47.

Source Address	Source Port	Destination Address	Destination Port	SSRC	Payload	Paquetes	Lost	Max Delta (ms)	Max Jitter	Mean Jitter	Estado
192.168.100.96	9206	192.168.100.104	10738	0x53a7	g729	1	0 (0.0%)	0.000	0.000	0.000	
192.168.100.96	9206	192.168.100.110	7078	0x53a7	g729	3112	0 (0.0%)	24.225	1.425	0.818	
192.168.100.96	9206	192.168.100.110	7078	0x6c3	g729	555	0 (0.0%)	26.194	10.860	1.130	
192.168.100.96	9206	192.168.100.104	10926	0x3aad	g711U	2	0 (0.0%)	19.136	0.054	0.432	
192.168.100.96	9206	192.168.100.110	7078	0x3aad	g711U	1408	0 (0.0%)	38.989	2.955	0.743	
192.168.100.104	10738	192.168.100.96	9206	0x7665ac0	g729	4	0 (0.0%)	0.000	3.521	15.000	
192.168.100.104	16172	192.168.100.96	9206	0xb25cf37	g729	4	0 (0.0%)	0.000	3.521	15.000	
192.168.100.104	10926	192.168.100.96	9206	0xb9ad43c	g711U	4	0 (0.0%)	19.993	0.052	0.221	
192.168.100.110	7078	192.168.100.96	9206	0x546744d3	g729	3074	32 (1.0%)	253.682	28.355	6.974	
192.168.100.110	7078	192.168.100.96	9206	0xc0244936	g729	1338	12 (0.9%)	424.866	48.920	13.487	
192.168.100.110	7078	192.168.100.96	9206	0xa057379a	g711U	1397	1 (0.1%)	134.539	25.196	3.506	

Figura 47: Jitter y paquetes perdidos

En la imagen se muestra los porcentajes o valores de la pérdida y *jitter* de los paquetes. Al comparar las estadísticas de los *codecs*, G729 presenta valores más altos que los de G711, se puede diferenciar entre las medidas de la pérdida de paquetes y *jitter*. Cabe mencionar que los valores recomendados para no degradar una comunicación, el *jitter*, desde el inicio y fin de la llamada tiene que ser inferior a 100 milisegundos y la pérdida de paquetes debe ser menor al 1%. Sin embargo, este último depende del *codec* que se utiliza. Debido a que cuanto mayor sea la comprensión del *codec*, más perjudicial será el efecto de la pérdida de paquetes. Y un claro ejemplo es la imagen 47 y la captura del audio de las llamadas. Un 1% de paquetes perdidos degrada más la comunicación si se utiliza G729 que G711.

Forward		Envío	Retorno	Gráfica					
192.168.100.96:9206 → 192.168.100.110:7078		Paquete	Sequence	Delta (ms)	Jitter (ms)	Skew	Ancho de banda	Marker	Estado
SSRC	0x00003aad	2991	6258	20.49	0.68	21.00	81.60		✓
Max Delta	38.99 ms @ 2827	2993	6259	19.44	0.68	21.56	81.60		✓
Max Jitter	2.96 ms	2995	6260	20.09	0.64	21.47	81.60		✓
Mean Jitter	0.74 ms	2997	6261	19.88	0.61	21.59	80.00		✓
Max Skew	22.18 ms	2999	6262	20.74	0.62	20.85	80.00		✓
RTP Packets	1408	3000	6263	19.89	0.58	20.95	80.00		✓
Expected	1408	3003	6264	20.03	0.55	20.93	81.60		✓
Lost	0 (0.00 %)	3005	6265	19.41	0.55	21.51	81.60		✓
Seq Errs	0	3007	6266	20.64	0.56	20.88	81.60		✓
Start at	67.454742 s @ 163	3009	6267	20.15	0.53	20.73	80.00		✓
Duration	28.70 s	3011	6268	19.65	0.52	21.08	81.60		✓
Clock Drift	2 ms	3013	6269	19.46	0.52	21.61	81.60		✓
Freq Drift	8000 Hz (0.01 %)	3015	6270	20.71	0.53	20.91	80.00		✓

Forward		Envío	Retorno	Gráfica					
192.168.100.110:7078 → 192.168.100.96:9206		Paquete	Sequence	Delta (ms)	Jitter (ms)	Skew	Ancho de banda	Marker	Estado
SSRC	0xc0244936	15396	573	41.82	8.77	-26.82	23.52		✓
Max Delta	424.87 ms @ 15705	15407	575	192.27	17.74	-179.09	19.68		Wrong
Max Jitter	48.92 ms	15408	576	0.00	17.88	-159.09	20.16		✓
Mean Jitter	13.49 ms	15409	577	0.00	18.01	-139.09	20.64		✓
Max Skew	-383.35 ms	15410	578	0.00	18.14	-119.09	21.12		✓
RTP Packets	1338	15411	579	0.00	18.25	-99.09	21.60		✓
Expected	1350	15412	580	0.00	18.36	-79.09	22.08		✓
Lost	12 (0.89 %)	15413	581	0.00	18.47	-59.09	22.56		✓
Seq Errs	6	15414	582	0.00	18.56	-39.09	23.04		✓
Start at	474.809283 s @ 14335	15415	583	0.00	18.65	-19.09	23.52		✓
Duration	27.06 s	15421	584	107.95	22.98	-107.04	21.60		✓
Clock Drift	-49 ms	15422	585	0.00	22.80	-87.04	22.08		✓
Freq Drift	7985 Hz (-0.18 %)	15423	586	0.00	22.62	-67.04	22.56		✓

Figura 48: Análisis de los paquetes de llamadas con G729 y G711

En la figura 48, la llamada con G729 presenta más pérdida de paquetes (12-0.89%) que con el *codec* G711 (0-0%). Además del

jitter que es 48.92 milisegundos frente a 2.96 de G711. Sin embargo, el ancho de banda usado es mayor para llamadas de este último (80 Kbps aproximados), a diferencia de los 24 Kbps o menor utilizados por G729.

Ambos *codecs* se encuentran dentro del rango recomendado para establecer una comunicación muy buena. Sin embargo, para más calidad, es preferible usar *codecs* G711, pero en escenarios de menor ancho de banda, es útil usar el *codec* G729.

3.5.4. Análisis de QoS con SIPP y llamadas reales

En primer lugar, se realizaron escenarios con *SIPP* y una llamada real, con el objetivo de analizar y determinar la cantidad de llamadas simultaneas sin que la llamada real estuviese afectada.

Se tiene la figura 49:

Source Address	Source Port	Destination Address	Destination Port	SSRC	Payload	Paquetes	Lost	Max Delta (ms)	Max Jitter	Mean Jitter
192.168.100.104	15626	192.168.100.116	6000	0x4a9477f	g711U	427	0 (0.0%)	308.679	41.089	19.193
192.168.100.104	15342	192.168.100.116	6000	0x7e1af224	g711U	426	0 (0.0%)	310.479	40.833	19.269
192.168.100.104	10476	192.168.100.116	6000	0x6492da88	g711U	426	0 (0.0%)	310.479	40.804	19.156
192.168.100.104	13310	192.168.100.116	6000	0x6783ba59	g711U	427	0 (0.0%)	310.807	41.038	19.972
192.168.100.104	15238	192.168.100.116	6000	0x7dcd54b	g711U	426	0 (0.0%)	308.679	41.105	20.057
192.168.100.104	17942	192.168.100.116	6000	0x820bb29	g711U	427	0 (0.0%)	308.679	41.032	20.437
192.168.100.104	11398	192.168.100.116	6000	0x773fc5c	g711U	427	0 (0.0%)	308.679	41.110	20.555
192.168.100.104	11894	192.168.100.116	6000	0x77669558	g711U	428	0 (0.0%)	311.419	41.190	21.108
192.168.100.104	13888	192.168.100.116	6000	0x3bf89213	g711U	426	0 (0.0%)	310.479	40.838	21.833
192.168.100.104	17388	192.168.100.116	6000	0xeb68852	g711U	428	0 (0.0%)	310.163	40.861	21.876
192.168.100.104	13898	192.168.100.116	6000	0x67ac174c	g711U	428	0 (0.0%)	310.479	40.841	21.741
192.168.100.104	13690	192.168.100.116	6000	0x68a71103	g711U	433	0 (0.0%)	308.679	40.934	21.900
192.168.100.104	10540	192.168.100.116	6000	0x634b1180	g711U	440	0 (0.0%)	311.419	41.143	22.285
192.168.100.104	16028	192.168.100.116	6000	0x6cfe25c4	g711U	437	0 (0.0%)	310.807	41.114	22.189
192.168.100.104	13098	192.168.100.116	6000	0x493fe840	g711U	438	0 (0.0%)	310.479	40.837	22.367
192.168.100.104	18424	192.168.100.116	6000	0x3974ad1d	g711U	427	0 (0.0%)	308.679	41.023	21.783
192.168.100.104	17306	192.168.100.116	6000	0x3bd6ed2c	g711U	434	0 (0.0%)	308.679	41.085	22.306
192.168.100.104	15664	192.168.100.116	6000	0x31bc4563	g711U	431	0 (0.0%)	310.807	41.039	22.005
192.168.100.104	17048	192.168.100.116	6000	0x29c81990	g711U	436	0 (0.0%)	308.679	41.092	22.039
192.168.100.104	18842	192.168.100.116	6000	0x60f1d8c6	g711U	440	1 (0.2%)	310.479	40.794	22.236
192.168.100.110	7078	192.168.100.96	9206	0x99b3d60d	g729	2890	0 (0.0%)	350.570	37.378	7.459
192.168.100.110	7078	192.168.100.96	9206	0x99b3d60d	g729	2890	0 (0.0%)	350.570	37.378	7.459

Forward		Envío	Retorno	Gráfica					
		Paquete	Sequence	Delta (ms)	Jitter (ms)	Skew	Ancho de banda	Marker	Estado
192.168.100.110:7078 → 192.168.100.96:9206		24186	188	19.08	2.38	-0.81	24.00		✓
SSRC	0x99b3d60d	24222	189	21.53	2.33	-2.34	24.00		✓
Max Delta	350.57 ms @ 41765	24259	190	21.68	2.29	-4.01	24.48		✓
Max Jitter	37.38 ms	24296	191	16.62	2.36	-0.63	24.48		✓
Mean Jitter	7.46 ms	24333	192	21.45	2.30	-2.08	24.00		✓
Max Skew	-380.16 ms	24367	193	18.62	2.24	-0.70	24.48		✓
RTP Packets	2890	24407	194	21.02	2.16	-1.72	24.48		✓
Expected	2890	24443	195	18.21	2.14	0.07	24.48		✓
Lost	0 (0.00 %)	24483	196	19.93	2.01	0.13	24.48		✓
Seq Errs	0	24520	197	23.71	2.12	-3.58	24.00		✓
Start at	32.515644 s @ 16878	24557	198	16.07	2.23	0.35	24.48		✓
Duration	57.78 s	24595	199	20.74	2.14	-0.39	24.00		✓
Clock Drift	15 ms	24647	200	20.69	2.05	-1.08	24.00		✓
Freq Drift	8002 Hz (0.03 %)								

Figura 49: Escenario 50 llamadas y una real

Para este escenario, se simularon 50 llamadas con 50 intentos por segundo, generando tráfico SIP y RTP (gt11u):

**SIPP -sn UAC -s 1002 -l 50 -r 50 -d 10000 -mi 192.168.100.116
192.168.100.104**

De la fig.49 se observa que la llamada real se ha mantenido con la calidad que ofrece g729, no hubo perdidas en la sesión mantenida, el *jitter* dentro del rango recomendado (37.378 ms) y un consumo de 24 Kbps aproximado de ancho de banda. Inclusive el flujo RTP generado por el simulador no presenta perdidas casi en su totalidad. Lo siguiente fue simular el escenario de las 100 llamadas simultaneas. De acuerdo a los resultados obtenidos anteriormente, a partir de esta cantidad (CPU a más del 100%) el número de llamadas fallidas empieza a crecer exponencialmente, como se observa en la figura 50.

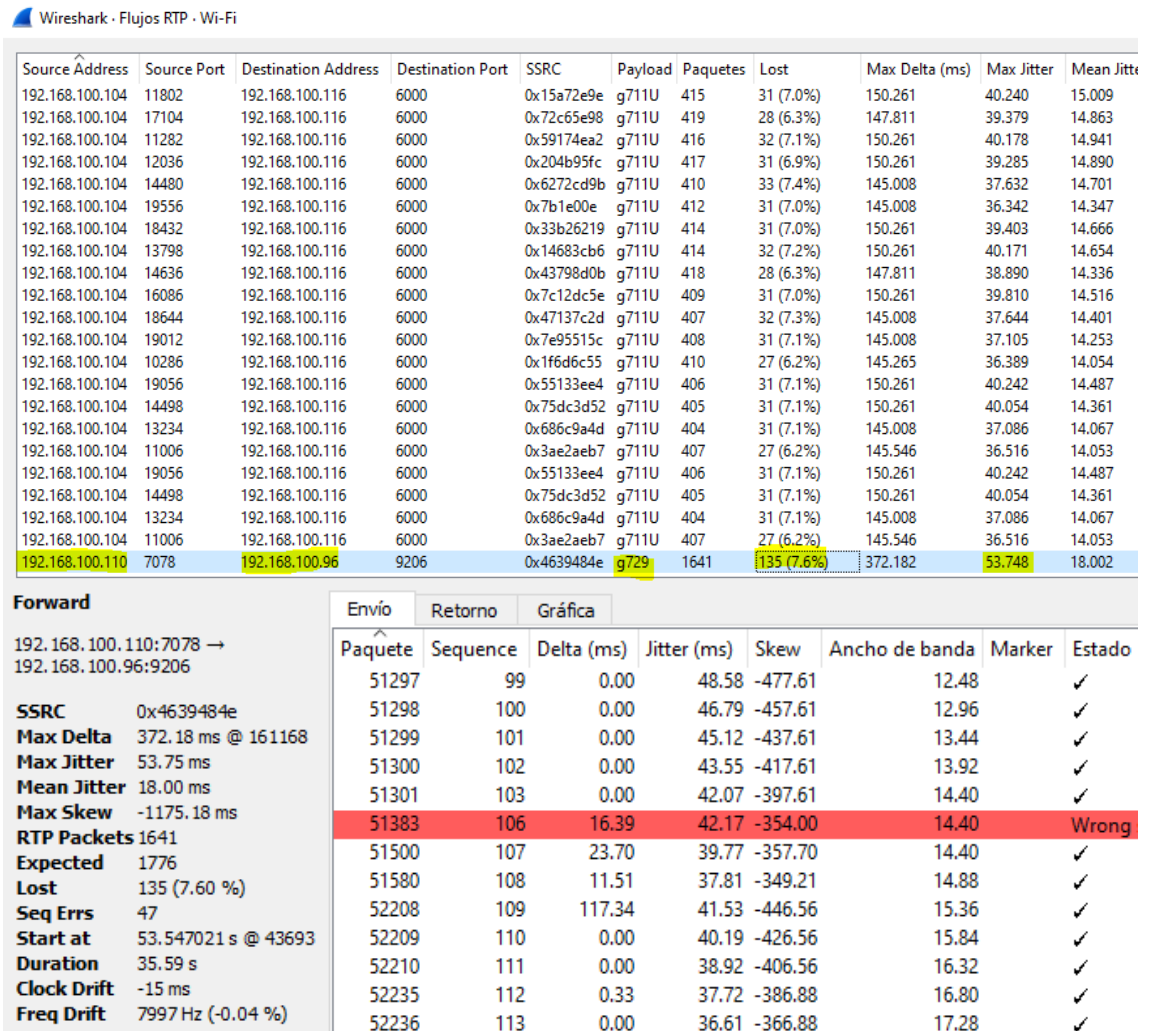


Figura 50: Escenario 100 llamadas simultaneas y una llamada real

Al ejecutar este escenario, cada periodo de tiempo, la llamada se veía afectada, degradando la comunicación, reflejándose en el

análisis de la captura. Hay una pérdida de paquetes considerable, sobrepasando el 1% recomendado. El *jitter* se mantiene debajo del rango de 100 ms, pero es mayor al anterior escenario. Así también, el flujo RTP de las llamadas simuladas presentan gran pérdida y un *jitter* variable. En consecuencia, hay menor consumo del ancho de banda debido a los paquetes perdidos.

Generando y simulando distintos escenarios, se genera la tabla 28:

Tabla 28: Análisis de QoS en distintos escenarios

Escenarios	Perdidas (%)	Max. Jitter (ms)	CPU (%)	Temperatura CPU-GPU (°C)
25	0	15.96	33.5	47 – 47.8
50	0	37.378	56.7	48 – 48.3
60	0.1	39.383	78.4	48 – 48.9
80	6	40.234	96.3	51 – 52.1
100	7.6	53.748	103.6	49 – 49.4

Como se muestra en la tabla, hay gran variación de *jitter* cuando el número de llamadas aumenta. Sin embargo, esto depende de la cantidad de datos que fluye a través de la red. Se adjunta la temperatura del CPU de la central, con el objetivo de corroborar que su nivel afecta en la cantidad de llamadas que procesa y el tiempo de llegada entre cada sesión.

Evaluando los escenarios con relación al MOS (*Mean Opinion Score*), de resume a la tabla 29 y 30:

Tabla 29: Calidad de la central VoIP en distintos escenarios:

ESCENARIOS	CALIDAD
25	EXCELENTE
50	EXCELENTE
60	ACEPTABLE
80	BAJO
100	BAJO

Tabla 30: Parámetros de QoS

Calidad VoIP	Excelente	Bueno	Aceptable	Pobre
Jitter	$t < 10$	$10 \leq t < 20$	$20 \leq t < 50$	$t \geq 50$
Latencia	$t < 50$	$50 \leq t < 150$	$150 \leq t < 300$	$t \geq 300$
Paquetes perdidos	$p < 1\%$	$1\% \leq p < 0.5\%$	$0,5 \leq p < 1,5$	$p \geq 1,5$

En base al análisis de los resultados, se asegura que la central VoIP implementada en la *Raspberry Pi* puede manejar hasta 50 llamadas simultaneas con una calidad excelente. A partir de este rango, las llamadas exitosas decrecerían exponencialmente.

3.5.5. Análisis de llamada entre centrales

Para este escenario, se configuro y preparo una central telefónica en una máquina virtual, para este caso se emplea *Asterisk* puro en una maquina Ubuntu. También se instala Issabel (anteriormente llamado Elastix) para el manejo del servidor con una interfaz gráfica.

Como se menciona anteriormente, se realiza la configuración necesaria para enlazar la troncal de una central a otra. Sea por CLI o la GUI que brinda el servidor. Para resumir la captura y que exponga los datos más relevantes, se utiliza una herramienta llamada SNGREP, que es un Shell de código abierto que captura todo el flujo SIP para su posterior análisis. No consume recursos pues utiliza ventanas CLIs para la visualización del flujo en tiempo real. Se instala en la central VoIP, para observar y analizar los paquetes entrantes y salientes en la *Raspberry* y no solo en el puerto de red del *softphone*. El flujo se captura en la figura 51.

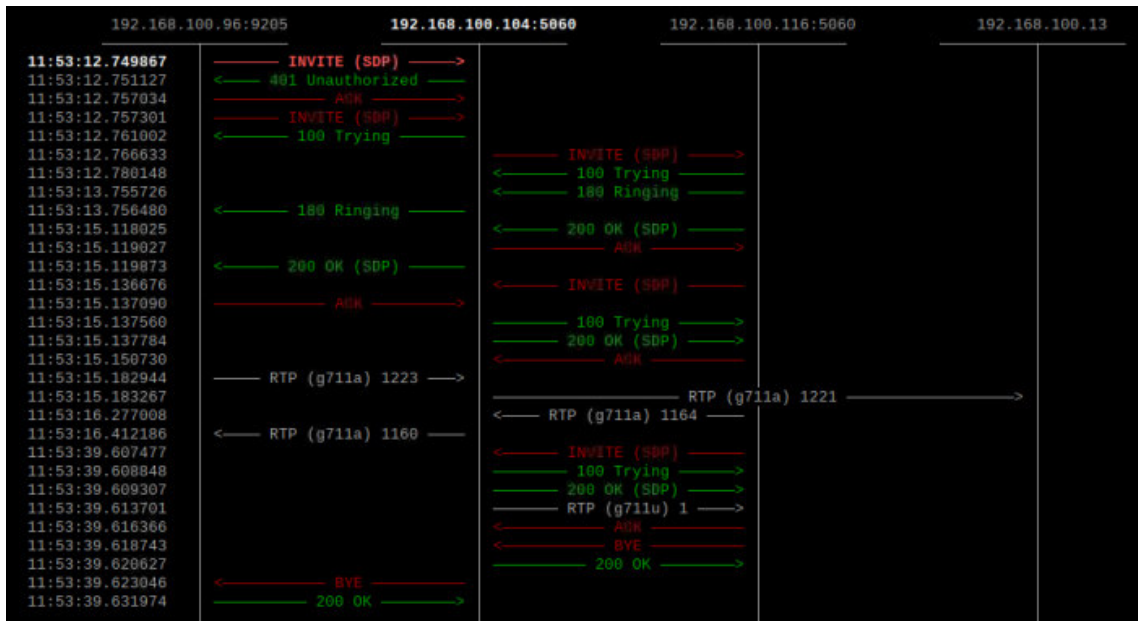


Figura 51: Flujo de llamadas entre centrales VoIP

Sngrep muestra gran información y múltiples ventanas, se puede cambiar a distintos modos y filtros, muy similar a *Wireshark*. En la imagen, se puede ver la ventana más resumida del flujo de llamada. Cada línea muestra información a detalle del evento o mensaje que se realizó en la sesión, como se observa en la figura 52.

```

INVITE sip:200@192.168.100.104 SIP/2.0
To: <sip:200@192.168.100.104>
From: 100<sip:100@192.168.100.104>;tag=a33b3124
Via: SIP/2.0/UDP 192.168.100.96:9205;branch=z9hG4bK-d87543-606104153-1--d87543-;rport
Call-ID: 480731729a0b724f
CSeq: 1 INVITE
Contact: <sip:100@192.168.100.96:9205>
Max-Forwards: 70
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
Content-Type: application/sdp
User-Agent: eyeBeam release 3007n stamp 17816
Content-Length: 236

v=0
o=- 8715521 8715528 IN IP4 192.168.100.96
s=eyeBeam
c=IN IP4 192.168.100.96
t=0 0
m=audio 9206 RTP/AVP 0 8 101
a=alt:1 1 : 04191E23 00000020 192.168.100.96 9206
a=fmtp:101 0-15
a=rtpmap:101 telephone-event/8000
a=sendrcv

```

Figura 52: Datos del primer mensaje INVITE(SDP) del flujo de llamada

Puede notarse el contacto de origen, la IP, el puerto, la extensión, además del dispositivo que realizó la llamada, en este caso, un *softphone* eyeBeam

Como se observa el flujo en la fig.51, comienza con el *INVITE* de 192.168.100.116, quien es el *softphone* que inicia la sesión. Este se comunica con su servidor, que es la *Raspberry* (192.168.100.104). Luego de confirmar (y algunos mensajes de sesión), se comunica con la otra central (Ubuntu, 192.168.100.116). Este se comunica con su extensión (200:192.168.100.13), no se visualiza este proceso pues, es tráfico de la central vecina, sin embargo, siempre está contactando y comunicando a la central principal los mensajes que recibe de la extensión solicitada. Como que está intentando (*Trying*) y sonando el teléfono (*Ringin*g). Luego del intercambio y de confirmar los mensajes (*ACK*), empieza el flujo de datos RTP. En este caso, se usa el *codec* G711 *alaw*, además de que los datos no viajan directamente entre extensiones, al tratarse de redes locales o WAN distintas (como es normal las direcciones IP públicas). Los servidores se encargan de transportar la data o media (la voz) desde y hacia las extensiones. El flujo completo se puede observar en la figura 53

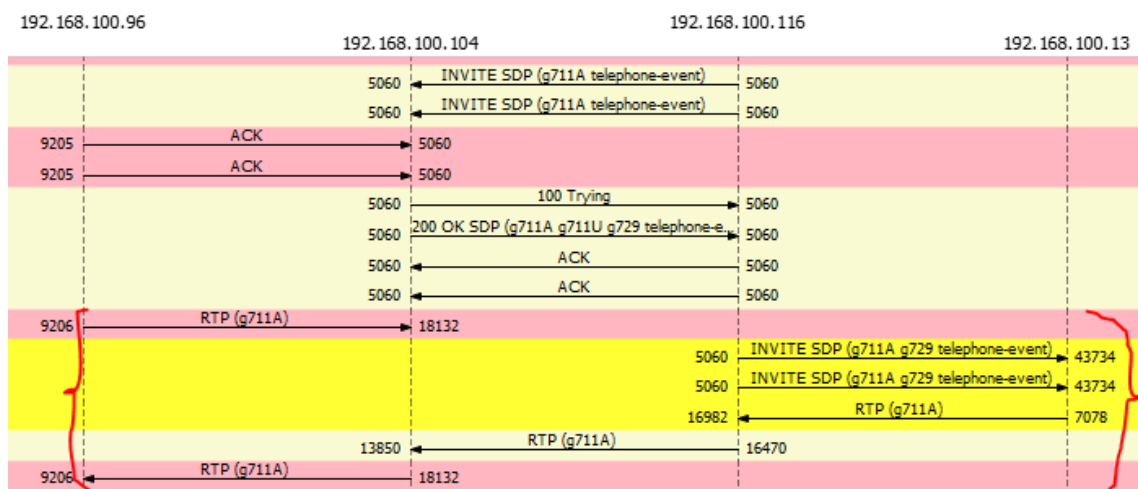


Figura 53: Flujo RTP entre las llamadas realizadas

Captura realizada en *Wireshark*, al hacer uso de una máquina virtual, se puede capturar la sesión completa. El *softphone* de la central vecina envió su media, su servidor recibe y envía a la central principal, a su vez, lo transporta hacia la extensión o usuario final que realiza la llamada.

Realizando unas llamadas, capturando el *jitter* y perdida de paquetes, se tiene en la figura 54:

Wireshark · Flujos RTP · Wi-Fi

Source Address	Source Port	Destination Address	Destination Port	SSRC	Payload	Paquetes	Lost	Max Delta (ms)	Max Jitter	Mean Jitter
192.168.100.104	13850	192.168.100.116	16470	0x18913f42	g711U	1	0 (0.0%)	0.000	0.000	0.000
192.168.100.104	18132	192.168.100.96	9206	0x714a3a9f	g711A	1160	0 (0.0%)	351.835	45.740	11.789
192.168.100.116	16470	192.168.100.104	13850	0x128cb537	g711A	2328	-1164 (-100.0%)	157.806	11.063	3.061
192.168.100.13	7078	192.168.100.116	16982	0xb428070c	g711A	1164	3 (0.3%)	157.751	17.359	6.122
192.168.100.96	9206	192.168.100.104	18132	0x2e2e	g711A	2446	-1223 (-100.0%)	39.923	1.595	0.659
192.168.100.116	10848	192.168.100.104	19794	0x5eec3216	g711A	2068	-1034 (-100.0%)	117.284	9.087	2.229
192.168.100.13	7078	192.168.100.116	10888	0x3d739499	g711A	1035	1 (0.1%)	117.279	13.966	4.431

Figura 54: *Jitter* y Paquetes perdidos de unas llamadas entre centrales

Tanto el *jitter* y los paquetes perdidos son mínimos. Los que muestran valores negativos, corresponden a datos que no se recibieron o que se esperaban recibir. Esto se debe principalmente a que la extensión (usuario) se mantuvo en silencio y *Wireshark* lo clasifica como perdida. Pero lo principal son los usuarios y envío entre las centrales, pues dependen de la red local del proveedor. Los enlaces entre troncales, serán responsabilidad de sus redes públicas.

Para la figura 55, analizando uno de los flujos RTP:

Forward		Envío	Retorno	Gráfica					
		Paquete	Sequence	Delta (ms)	Jitter (ms)	Skew	Ancho de banda	Marker	Estado
192.168.100.13:7078 →									
192.168.100.116:16982									
SSRC	0xb428070c	6959	261	27.26	2.77	111.14	123.20		✓
Max Delta	157.75 ms @ 7135	6965	262	14.00	2.97	117.15	123.20		✓
Max Jitter	17.36 ms	6971	263	21.87	2.90	115.27	123.20		✓
Mean Jitter	6.12 ms	6977	264	24.14	2.98	111.13	123.20		✓
Max Skew	-504.22 ms	6983	265	13.05	3.23	118.09	123.20		✓
RTP Packets	1164	6989	266	20.53	3.06	117.56	123.20		✓
Expected	1167	6995	267	19.39	2.91	118.16	123.20		✓
Lost	3 (0.26 %)	7001	268	22.15	2.86	116.02	123.20		✓
Seq Errs	1	7007	269	18.07	2.80	117.95	124.80		✓
Start at	402.213402 s @ 5337	7013	270	17.33	2.79	120.61	81.60		✓
Duration	23.20 s	7019	271	21.81	2.73	118.80	80.00		✓
Clock Drift	387 ms	7025	272	18.79	2.64	120.01	80.00		✓
Freq Drift	8134 Hz (1.67 %)	7031	273	27.32	2.93	112.69	80.00		✓

Figura 55: Análisis de la media enviada del usuario al servidor

Se obtiene un *jitter* promedio de 6.12 ms y un máximo de 17.36 ms. Los paquetes perdidos fueron de un 0.26%. Ambos valores encontrándose dentro del rango promedio. También se puede observar un pequeño incremento del ancho de banda, sin embargo, fue en un momento corto, volviendo a su consumo normal de 80 Kbps, correspondiente al G711.

3.5.6. Ancho de banda requerido para la central VoIP

Habiendo definido las características, funciones y rendimiento de la Raspberry Pi como central telefónica. Es vital calcular el ancho de banda requerido para realizar llamadas simultáneas y de buena calidad, según las necesidades de cada empresa. Generalmente, estas cifras se calculan en base al codificador empleado y el número de llamadas.

Las llamadas VoIP consisten en dos partes esenciales. Por un lado, se encuentra la parte de la señalización (SIP, IAX, etc.), que se encarga de realizar la llamada, establecer la sesión, el timbrado al destino, el cierre o desconexión y otras comunicaciones o sesiones realizadas en ambos extremos. Por otra parte, se encuentra el audio transportado, por medio de RTP a través de la red. En relación al BW (ancho de banda) consumido por la señalización SIP frente a RTP, es realmente mínimo, por tanto, se puede desprestigiar para el cálculo en mayoría. El tema es frente al audio que transporta RTP. Para poder transmitir la voz a través de la red, el audio en bruto tiene que ser codificado y comprimir el tamaño sin afectar la calidad. Para estos propósitos se emplean distintos *códecs* o codificadores, que, dependiendo de sus algoritmos, ofrecerán una calidad de voz distinta, un consumo de ancho de banda diferente, y algunos con un menor o mayor uso de la memoria y CPU. Por ello, el tema de la elección del codificador es importante para las aplicaciones de VoIP (ElastixTech, 2019).

Como un paquete que viaja a través de una red de internet o local, pasa o contiene varias etapas (como se mencionan en el modelo OSI). Para el cálculo del BW se realizan operaciones sencillas pero cuidadosas que contenían las cabeceras de los paquetes y el tamaño de cada protocolo, carga útil o *payload*, paquetes por segundo, velocidad de bit del codificador empleado, etc. Tal y como se muestra anteriormente en la tabla 25.

Todo ese cálculo se reducía a una pequeña pero importante fórmula:

$$BW = \text{Tamaño total del paquete} \times \frac{\text{tasa de bits del codec}}{\text{tamaño de carga útil de voz}}$$

Sin embargo, con el avance de las tecnologías y el uso de nuevas formas de transmitir a través de la red (fibra óptica y microondas, por ejemplo), reemplazando las antiguas redes que limitaban el ancho de banda, eliminaron esa práctica de realizar un cálculo muy tedioso, generalizándolo a ciertos factores de VoIP (NFON, 2021).

Anteriormente ya se ha definido y expuesto ciertos factores o elementos muy importantes para VoIP, como la latencia, el *jitter* o retardo, la pérdida de paquetes entre otros. Además de ello, el ancho de banda consumido por una llamada, en este caso, para el G729 con 24 kbps y G711 con 80 kbps aproximados. Por supuesto, con la ayuda de SIP, definir la cantidad de llamadas simultáneas, que para la Raspberry empleada, contiene un total de 50 llamadas de buena calidad relativamente.

Teniendo en cuenta esta información, se puede definir cuál es el ancho de banda necesario o para este escenario de 50 llamadas en simultáneo.

Generalizando, se requieren 100 *kbps* (kilobits por segundo) por llamada de buena calidad de audio. Por tanto, se aplica una fórmula muy práctica:

$$BW \text{ requerido} = \text{Llamadas simultáneas} \times 100 \text{ kbps}$$

Entonces, si se tienen 50 sesiones en tiempo real a la vez, se necesitarán 5000 *kbps* o 5 *mbps* (megabits por segundo).

Inclusive, si solo aplicamos el cálculo para el *codec* G729, cuyo BW es de 24 *kbps* aproximadamente, el consumo sería menor, tal y como se ve en la figura 56 y 57 con un calculador automático de BW para VoIP.

2. Bandwidth Calculator

Incoming Channel	Outgoing Channel
<input checked="" type="radio"/> Regular Audio Codecs Codec: <input type="text" value="g.729a-8.00Kibps"/>	<input checked="" type="radio"/> Regular Audio Codecs Codec: <input type="text" value="g.729a-8.00Kibps"/>
<input type="radio"/> Speex Audio Codec	<input type="radio"/> Speex Audio Codec
<input type="radio"/> MGCP <input type="radio"/> H323 <input checked="" type="radio"/> SIP <input type="radio"/> IAX2 <input type="radio"/> IAX2 trunked	<input type="radio"/> MGCP <input type="radio"/> H323 <input checked="" type="radio"/> SIP <input type="radio"/> IAX2 <input type="radio"/> IAX2 trunked
<input checked="" type="checkbox"/> RTCP	<input checked="" type="checkbox"/> RTCP
Number of simultaneous calls: <input type="text" value="50"/>	

Figura 56: Calculadora de BW con G729

Fuente: (Asteriskguru, 2021)

Incoming Bandwidth	Outgoing Bandwidth								
Calls: 50 RTP: 4.69 Kbps UDP: 3.13 Kbps IP: 7.81 Kibps Protocol: SIP Audio Codec: 8.00g.729a Kbps *SIP overhead is disregarded!	Calls: 50 RTP: 4.69 Kbps UDP: 3.13 Kbps IP: 7.81 Kibps Protocol: SIP Audio Codec: 8.00g.729a Kbps *SIP overhead is disregarded!								
Incoming bandwidth: <table style="margin-left: 20px;"> <tr><td>1240.31 Kbps</td></tr> <tr><td>1.21 Mbps</td></tr> <tr><td>155.04 KBps</td></tr> <tr><td>0.15 MBps</td></tr> </table>	1240.31 Kbps	1.21 Mbps	155.04 KBps	0.15 MBps	Outgoing bandwidth: <table style="margin-left: 20px;"> <tr><td>1240.31 Kibps</td></tr> <tr><td>1.21 Mbps</td></tr> <tr><td>155.04 KBps</td></tr> <tr><td>0.15 MBps</td></tr> </table>	1240.31 Kibps	1.21 Mbps	155.04 KBps	0.15 MBps
1240.31 Kbps									
1.21 Mbps									
155.04 KBps									
0.15 MBps									
1240.31 Kibps									
1.21 Mbps									
155.04 KBps									
0.15 MBps									
Total bandwidth (incoming and outgoing): <table style="margin-left: 20px;"> <tr><td>2480.62 Kbps</td></tr> <tr><td>2.42 Mbps</td></tr> <tr><td>310.08 KBps</td></tr> <tr><td>0.3 MBps</td></tr> </table>		2480.62 Kbps	2.42 Mbps	310.08 KBps	0.3 MBps				
2480.62 Kbps									
2.42 Mbps									
310.08 KBps									
0.3 MBps									

Figura 57: Calculadora de BW con G729

Fuente: (Asteriskguru, 2021)

Como se observa en las figuras, se tiene en cuenta el tamaño de los protocolos empleados y de la tasa de bits del *codec* G729. También puede notarse que la cabecera o el protocolo SIP es despreciado debido a su insignificante tamaño.

Para un escenario de 50 llamadas en simultaneo, con el *codec* G729, se requieren como mínimo de 2 a 3 *mbps* de ancho de banda para gestionar con la calidad ofrecida por el codificador. Sin embargo, en una red general estándar, los paquetes VoIP no son necesariamente

los únicos en viajar a través de la red. Se debe tener en cuenta los otros servicios que son empleados por los usuarios.

Cabe mencionar que el número de sesiones simultáneas, es un parámetro dependiente de las necesidades de los usuarios o la empresa, y casi nada por factores técnicos (es muy variable). Puede haber escenarios típicos de oficina, donde solo se emplean de un 20 a 30 por ciento de la cantidad total de llamadas establecidas. Pero es lógico asumir que se debe establecer el consumo de ancho de banda para un 100% total, como entornos de centro de llamadas, con el fin de no saturar la red y evitar la congestión de tráfico.

3.5.7. Recomendaciones y aplicaciones en la implementación de la central telefónica en el entorno empresarial

Conociendo los aspectos y características de la Raspberry Pi como central telefónica, y la cantidad de llamadas que puede manejar, se deben mencionar algunas recomendaciones y aplicaciones al implementar la central VoIP, en los entornos de pequeña o mediana empresa.

Generalmente estos aspectos son muy similares a los ya conocidos y practicados en cualquier red de informática, y aun más si son entornos empresariales o de negocios privados o públicos (Incibe,2021).

Esto incluye principalmente las buenas prácticas en las redes de internet, como las 5 características de una buena red de ordenadores, respaldos de seguridad y fuentes de alimentación, etc. Para el proyecto VoIP, que lógicamente trabaja en base al protocolo de internet, es de suma importancia que el proveedor de servicios que le brinda una red de comunicaciones y aplicaciones (acceso a internet) a la empresa, cumpla primordialmente en dichas características, como: la velocidad, tomando en cuenta el tipo de red o el medio en que se transmite. La seguridad, minimizando al máximo las vulnerabilidades de la red. Siempre confiable, generando redundancia para evitar pérdida de datos y que los equipos sigan trabajando. Escalable, brindando facilidad y posibilidad de añadir o

extender la red para mejorar el rendimiento. Finalmente, siempre disponible, que se halle completamente activa, o en todo caso, que se priorice los servicios cruciales de la empresa en situaciones de congestión o fallos en la red (Juliá, 2013).

Al cumplir con estas 5 características, se aseguran no solo que los servicios VoIP ofrecidos por la Raspberry cumplan al 100% con sus características. También que toda la información digital que se maneje en la empresa, viaje a través de la red, segura y sin pérdidas, generando eficiencia y eficacia, aprovechando al máximo las ventajas de las redes de comunicaciones o informáticas.

Se ha definido y analizado el ancho de banda necesario para que la central VoIP trabaje correctamente y se ha comentado los requisitos mínimos para el uso de la telefonía IP, tales como: Contar con al menos 100 kbps de BW, por usuario, subida y bajada. Una latencia inferior a 150 milisegundos, con *Jitter* y pérdida de paquetes inferior al 1%. Sin embargo, existen aspectos que se deben tener en cuenta sobre el flujo de internet que se contrata (Telsome, 2021):

- En primer lugar, el ancho de banda real no suele corresponder al paquete contratado, esto puede variar, ya sea a consecuencia de distintos factores (zona, medio físico, etc.).
- El BW en ocasiones seguidas es configurado asimétricamente.
- Por otro lado, en algunas ocasiones, el equipo o dispositivo de red es quien limita la velocidad y el ancho de banda.

Para implementar la central telefónica, y se adapte y trabaje en conjunto con la red existente en la empresa. Se debe considerar estos 3 aspectos. Debido a factores físicos, el ancho de banda real no siempre será el mismo que se contrata. Sin embargo, se puede exigir por ley que siempre se garantice un rango de porcentaje mínimo del paquete acordado para evitar fallos o desconexiones en la red local y global (Osiptel, 2021). Asimismo, exigir una simetría en el ancho de banda. Es decir, si se contrata un paquete o servicio de

10 mbps, el BW de subida debe ser el mismo que el de descarga o bajada.

Esto es realmente esencial para los servicios de comunicaciones que trabajan bajo el protocolo IP. De nada serviría si se tiene un alto BW de bajada, pero un bajo o inaceptable ancho de banda de subida. Esto generaría niveles de calidad de voz distintos y retrasos en el transcurso de la llamada o sesión.

Anteriormente se ha expuesto el ancho de banda mínimo o necesario para que la central telefónica administre 50 llamadas en simultaneo sin ningún problema. Con el G729, y contando los otros servicios que se pueden emplear, se requieren al menos 5mbps. Y esto debe ser igual para el BW de subida y bajada. Garantizando que ambas partes de la comunicación no tengan problemas al establecer llamadas o alguna otra aplicación.

El proyecto esencialmente tiene como objetivo disminuir los gastos en telefonía. Por supuesto, el uso de aplicaciones virtuales para realizar llamadas (*softphones*), son un tema que también se incluyen el desarrollo del proyecto. Para propósitos experimentales, se han empleado *softphones* de uso libre, sin costo alguno. Para el ámbito empresarial, diversos proveedores ofrecen sus propios teléfonos virtuales, sea para versiones móviles o de escritorio, en primera instancia gratuitos pero limitados.

En el caso del presente proyecto, para el ámbito de la pequeña o mediana empresa, se puede emplear sin ningún costo estas aplicaciones de uso libre y desarrollar versiones propias para cada empresa en específico. Generalmente, el costo de estos *softphones* vienen principalmente de la integración de distintos *códecs* para fines de gestión y rendimiento de la calidad de las centrales VoIP. Sin embargo, para la central telefónica del proyecto se trabaja en uno principalmente (o dos) como ya se menciono anteriormente. Licencias de uso libre estandarizadas, para distintas aplicaciones de comunicaciones unificadas.

Esto genera grandes ahorros y evita gastos realmente innecesarios, cumpliendo con los objetivos del proyecto. En consecuencia, la empresa puede gestionar dichos gastos para mejorar su red de servicios en otros aspectos, como redundancia y seguridad de sus fuentes de alimentación, como la implementación de estabilizadores o sistemas de alimentación ininterrumpidas (UPS).

CAPÍTULO IV: ANÁLISIS DE COSTO Y BENEFICIO

4.1. ANÁLISIS DE COSTOS

4.1.1. Recursos Humanos

En base a los recursos humanos utilizados para el proyecto, se abarco principalmente profesores y especialistas en redes y telefonía *VoIP*. Se realizaron consultas y se analizó el desarrollo del proyecto y en conjunto con los asesoramientos, se dieron mejores soluciones para realizar los objetivos propuestos. Todo ello se resume en la tabla 31:

Tabla 31: Costo de recursos humanos empleados en el proyecto

Descripción de la Actividad	Personal	Costo
Profesor de Redes	1	S/. 80.00
Profesor de Linux	1	S/. 60.00
Especialista en <i>VoIP</i>	1	S/. 80.00
COSTO TOTAL DE RR.HH		S/. 220.00

Como se muestra en la tabla, son los gastos de personal que se ejecutó durante las diferentes etapas del desarrollo de la aplicación. Desde la recopilación de información, el análisis, diseño, la programación, implementación y pruebas.

4.1.2. Recursos materiales

En parte por los recursos materiales requeridos, incluye solamente los gastos necesitados para la implementación de la central telefónica. No abarca gastos en la adquisición de teléfonos IP o *Gateway* de voz, *Routers* o *switchs*. Como se mencionó, la red de comunicación es escalable, el tamaño o cantidad de usuarios que se conecten, los enlaces entre sedes o centrales, dependerán exclusivamente de la empresa que planea implementar un servidor *VoIP*. El fin es que la misma empresa defina sus límites y lo que simplemente desea. Los materiales se muestran en la tabla 32:

Tabla 32: Recursos materiales usados para el proyecto

Actividades	Materiales	Costos
Adquisición de las herramientas tecnológicas	1 <i>Raspberry Pi</i> 3B+	S/. 250.00
	1 Micro SD 16GB	S/. 32.00
	1 teclado portatil	S/. 50.00
	1 pantalla táctil 7"	S/. 120.00
	1 ventilador pequeño	S/. 10.00

En la tabla se muestra los costos por hardware. No se menciona los costos por *software* pues los sistemas empleados son de uso libre, es decir gratis.

4.2. ANÁLISIS DE BENEFICIO

4.2.1. Beneficios Tangibles

Uno de los objetivos del proyecto es la disminución de costos por servicios de comunicación por voz como la telefonía tradicional. Habiendo expuesto el desarrollo y las increíbles ventajas que trae la implementación de la central *VoIP*, se puede afirmar que los beneficios tangibles que entrega la central son en gran manera muy útiles para pequeñas o medianas empresas.

En primer lugar, a diferencia de los costos de telefonía tradicional y de otros proveedores de servicios *VoIP*, no requieren pagos mensuales o anuales. Es decir, los gastos son solo en implementación, y para futuras mejoras, soporte técnico si es requerido. Esta es un gran beneficio económico ya que ahorra en gran manera gastos por comunicación a distancia.

Por otra parte, no se gasta en la habilitación o creación de nuevos anexos (usuarios) como se hace en la telefonía tradicional. La cantidad de usuarios dependerá de la escalabilidad de la red y del rendimiento de la central *VoIP*. No es necesario implantar nuevas redes o gastar en cableado. La central se acopla a la red actual de la empresa, y los servicios que brindará, dependerá netamente de la empresa.

4.2.2. Beneficios Intangibles

Contar con un sistema de comunicaciones rápido y confiable, con una calidad de servicio alta, genera en el entorno de la empresa, una agilización y trabajo ordenado por parte de los empleados.

La implementación de una central *VoIP* que ofrece calidad de voz y video, genera un ambiente profesional en parte de los empleados, hacia los clientes y viceversa. Cuando existe sistemas que agilizan y mejoran el rendimiento de trabajadores, el desarrollo económico llega como un increíble resultado, dando literalmente buenos frutos como empresa.

4.3. ANÁLISIS DE SENSIBILIDAD

4.3.1. Desarrollo del flujo de caja

El flujo de caja o *cash Flow* representa la diferencia entre los cobros y pagos realizados en periodo de tiempo concreto. Es decir, refleja las salidas y entradas financieras de la empresa.

Por lo tanto, para realizar una visión del proyecto como producto en el mercado, se realiza el siguiente flujo de caja durante 12 meses, se tiene la tabla 33:

Tabla 33: Flujo de caja anual

Mes	N° Clientes	Ingreso S/. (A)	Egreso S/. (B)	Flujo de efectivo neto S/. (A-B)
1	1	500	300	200
2	2	1000	600	400
3	4	2000	1200	800
4	6	3000	1800	1200
5	10	5000	3000	2000
6	14	7000	4200	2800
7	20	10000	6000	4000
8	24	12000	7200	4800
9	30	15000	9000	6000
10	32	16000	9600	6400
11	36	18000	10800	7200
12	40	20000	12000	8000
TOTAL		109500	65700	43800

De la tabla, el ingreso viene a ser el costo de la central VoIP. El egreso vendría a ser gasto requerido para su implementación. Teniendo a la diferencia de ambos como el flujo de efectivo neto.

4.3.2. Análisis del VAN

Para el cálculo del valor actual neto (VAN) en este proyecto. Se tiene la siguiente formula:

$$VAN = -I + \sum_{n=1}^N \frac{Q_n}{(1+r)^n}$$

Donde:

- I = Inversión
- Q_n = Flujo de caja del tiempo n (normalmente años)
- r = tasa de interés
- N = Número de años de inversión

Para este caso se tiene:

$$I = S/. 1000$$

$$r = 10\% = 0.10$$

$$N = 12 \text{ meses}$$

Para más facilidad, se calcula mediante una tabla de Excel, la figura 56 muestra la formula:

Qn	Inversion =	1000	Tasa = 10%
200			
400			
800			
1200			
2000			
2800			
4000			
4800			
6000			
6400			
7200			
8000			

$VAN = -I + \sum_{n=1}^N \frac{Q_n}{(1+r)^n}$	
VAN =	\$18,131.97

Figura 58: Calculo del valor actual neto en una tabla de excel

Se obtiene un valor 18131.97. Si el valor es positivo, indica que la inversión del proyecto es rentable, si es negativo, es claro que no será productivo.

4.3.3. Análisis del TIR

La tasa interna de retorno parte del flujo efectivo neto, pero también fuerza al VAN a cero. Lo que se pretende con TIR, es identificar la tasa de interés que vuelve al VAN a cero.

$$TIR = k_{TIR}: VAN(k_{TIR}) = 0$$

Donde k_{TIR} es la tasa de descuento que representa la TIR.

También se le encuentra de la siguiente manera:

$$TIR = \sum_{T=0}^n \frac{Fn}{(1+i)^n} = 0$$

Para el cálculo, en la figura 57 se muestra en una tabla de Excel:

-1000	Inversion =	1000	Tasa = 10%
200			
400			
800			
1200			
2000			
2800			
4000	VAN =	\$18,131.97	
4800			
6000	TIR=	0.74	74%
6400			
7200			
8000			

Figura 59: Cálculo del TIR en una tabla de Excel

Se obtiene un 0.74 con tasa de retorno en el proyecto

CONCLUSIONES

Dentro de una empresa, mantener los sistemas de comunicación en buen estado es importante, sea para agilizar la gestión o mantener un constante contacto directo con los clientes. Por lo que adquirir servicios de telefonía se vuelve imprescindible. El presente proyecto de telefonía IP ofrece varias ventajas, ya que optimiza recursos a través de la misma red de internet. Estos sistemas son adaptables, es decir, no requiere dedicación o implementación de redes especiales como lo era con la telefonía tradicional. Todo esto permitía un ahorro increíble en infraestructura, además de que muchos de los servicios de comunicación como llamadas de voz, video, mensajería, buzón de voz, *call center*, etc. Se puede alojar en un solo lugar como centralita virtual, sin necesidad de gastar por cada servicio y además de que se le puede añadir múltiples funcionalidades, que eran inalcanzables o requerían un costo elevado para la telefonía tradicional.

Al realizar las pruebas de rendimiento, se analizó y determino que la *Raspberry Pi 3 B+* soporta 50 llamadas en simultaneo con una calidad excelente y un uso de CPU y memoria estables sin saturación de sistema. Se establece el *codec* G729 como primera opción debido a su menor consumo de ancho de banda. Se tiene en cuenta los escenarios y la infraestructura de la red de empresas pequeñas y medianas, donde en su mayoría poseen una red de internet de velocidad baja o media, transportando toda la información por un solo canal. Por ello es recomendable el G729. Sin embargo, en escenarios donde se posea velocidad media o alta, o una red dedicada a la comunicación, es muy recomendable emplear el *codec* G711, debido a su mayor calidad y menor riesgo de pérdida de paquetes. Además de ello, se define el ancho de banda mínimo requerido para que la central trabaje sin congestión o interrupciones de pérdida de datos, que equivalen a unos 5 mbps aproximadamente.

En la provisión de servicios de Telefonía IP, la calidad de servicio es un factor importante para su gestión. Los principales problemas que intervienen en *QoS* para *VoIP*, son la pérdida de paquetes, el *jitter*, la latencia y el eco. Los cuales pueden ser resueltos con distintas técnicas. En las pruebas con la central *VoIP*,

se expone valores dentro de los rangos aceptables o recomendados para una comunicación estable y buena calidad de voz. Sin embargo, en redes empresariales o públicas, es un hecho que pueden variar. Para no degradar la comunicación, se aplican técnicas como reservar anchos de banda para los canales de voz, evitando latencias. Para *jitter* mayores de 100 milisegundos (no recomendable), las soluciones más prácticas son el manejo del *jitter* buffer, que implica asignar un espacio o cola para los paquetes con retraso. Esto disminuiría la pérdida de tramas, pero generaría un retraso. Dependerá de la prioridad del cliente. Si existen ecos muy altos, pueden aplicarse supresores o canceladores de ecos, aunque requieren mayor procesamiento. Para evitar la pérdida de paquetes, debe gestionarse bien los *codecs* empleados, además de evitar los silencios innecesarios en una conversación, y su solución más práctica es activando el sistema de silencio (muted), evitando congestionar los enlaces y no afectar la sesión.

El simulador *SIPP* es muy útil para analizar las centrales *VoIP* y su rendimiento, sin embargo, requieren un buen nivel de conocimiento en telefonía y redes para poder establecer los distintos escenarios que esta aplicación ofrece. Incluso en la práctica, profesionales pueden obtener errores si no estudia a fondo la operación de *SIPP*. Es muy útil pero no fácil de gestionar, sobre todo si se maneja los comandos o lenguaje que emplea. En este caso, se crea un script en Python para ejecutar de manera sencilla sus comandos, y no rendirse ante su desafiante sistema.

Con la implementación de la central *VoIP* se resuelven muchos problemas sobre la telefonía fija, que es el costo y la calidad de servicio. Lo interesante de estas tecnologías es que se desarrollan y se adaptan a la necesidad del usuario e incluso puede enlazarse con distintas sedes remotas y a la misma PSTN. Siendo una increíble solución para los sistemas de comunicación empresarial.

RECOMENDACIONES

Durante el desarrollo del proyecto, en las distintas fases que contiene, se buscaron varias y posibles soluciones tanto al hardware como *software* para desarrollar un sistema de comunicaciones completo. Al usar *software* de licencia libre, es normal que sea susceptible a fallos, o en todo caso vulnerable a equivocaciones humanas. Se debe mantener un control suave y constante para evitar problemas de conexión o relacionados.

Se debe mantener una observación clara en base a la seguridad dada para la central. Si bien *Asterisk* brinda valores añadidos como la seguridad la distribución de claves y control de llamadas para las extensiones, tanto internas como externas. Al usar protocolos no orientados a la seguridad, puede haber desviaciones malintencionadas de usuarios no registrados y acceder a la información del sistema que podrían comprometer a la central. Felizmente, Linux es un sistema lleno de herramientas antispyware, que si ocurriesen los peores casos para la centralita. Existen un gran número de soluciones.

Si bien las plataformas graficas de *FreePBX* han hecho sencillo el manejo de *Asterisk* para la central. Estas herramientas en sus inicios fueron orientadas al uso empresarial, donde normalmente se cuenta con personal técnico. Por ello, para establecer este sistema en ambientes domésticos, es recomendable realizar capacitaciones previas o en todo caso, desarrollar interfaces mucho más fáciles que los usuarios cotidianamente puedan manejar.

REFERENCIAS

- Asterisk. (23 de diciembre de 2021). *What is Asterisk?* Recuperado de <https://www.Asterisk.org/get-started>
- Asteriskguru. (15 de diciembre de 2021). *Bandwith Calculator*. Recuperado de https://www.asteriskguru.com/tools/bandwidth_calculator.php
- Astudillo, G. (18 de agosto de 2009). *Sistemas de VoIP con Asterisk: módulo I*. Recuperado de <https://es.slideshare.net/gastudillo/sistemas-de-VoIP-con-Asterisk-modulo-i>
- Bardales, D. y Flores, L. (2015). *Diseño y simulación de una red de voz sobre IP para la reducción de costos derivados de las llamadas telefónicas*. (tesis de grado). Recuperado de <http://dspace.unitru.edu.pe/handle/UNITRU/9710>
- Bastidas, G. J. (2016). *Prototipo de Central Telefónica para entornos Domésticos* (tesis de grado). Recuperado de <https://bibdigital.epn.edu.ec/handle/15000/16473>
- Cisco. (13 de abril de 2016). *Voz sobre IP – Consumo de ancho de banda por llamada*. Recuperado de https://www.cisco.com/c/es_mx/support/docs/voice/voice-quality/7934-bwidth-consume.html
- Daniel, L. P. (2015). *Guía para la toma de decisiones en redes unificadas* (tesis de maestría). Recuperado de <http://docplayer.es/622177-Universidad-de-mendoza.html>
- Dong, Q. (2011). Research on the Performance of Asterisk-Based Media Gateway. *Fourth International Symposium on Knowledge Acquisition and Modeling, Sanya, 2011*. 347-349. Recuperado de <https://doi.ieeecomputersociety.org/10.1109/KAM.2011.100>
- Editorial Etecé. (5 de agosto de 2021). *Concepto de medios de comunicación antiguos*. Recuperado de <https://concepto.de/medios-de-comunicacion-antiguos/>
- Elastixtech. (2019). *Interconexion a la PSTN*. Recuperado de <http://elastixtech.com/fundamentos-de-telefonía/interconexion-a-la-pstn/>
- Elastixtech. (2019). *Calcular Ancho de Banda en VoIP*. Recuperado de <https://elastixtech.com/calcular-ancho-de-banda-en-voip/>
- Ermez. (2014). *Calidad de Voz IP*. Recuperado de http://www.ermez.com/Soporte/Documentacion/Aqct_Web/Calidad_Voz_IP.htm
- Estévez, A. J. (2018). *Diseño e implementación de un prototipo de central telefónica con aplicación domótica de bajo costo*. (tesis de grado). Recuperado de <http://bibdigital.epn.edu.ec/handle/15000/19823>
- Gómez, J. y Gil, F. (2008). *VoIP y Asterisk Redescubriendo la Telefonía*. Recuperado de https://www.ra-ma.es/libro/voip-y-asterisk-redescubriendo-la-telefonía_48383/

- Guerra, R. F. (2013). *Diseño de un sistema de control domótico y video vigilancia supervisado por un teléfono móvil*. (tesis de grado). Recuperado <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/5375>
- Juliá, S. (15 de septiembre de 2013). *Las 5 características de una Buena Red Informática*. Recuperado de <https://www.gadae.com/blog/5-caracteristicas-red-informatica/>
- Incibe. (2021). *Buenas prácticas en el área de Informática*. Recuperado de <https://www.incibe.es/protege-tu-empresa/que-te-interesa/buenas-practicas-area-informatica>
- International Telecommunications Union. (2011). *Recommendation ITU-T Y.1540: Serie Y: Infraestructura Mundial de la Información y Aspectos del Protocolo Internet*. Aspectos del Protocolo Internet – Calidad de Servicio y Características de Red. Recuperado de <https://www.itu.int/rec/T-REC-Y.1540-201912-I/es>
- International Telecommunications Union. (2011). *Recomendación ITU-T Y.1541: Serie Y: Infraestructura Mundial de la Información, Aspectos del protocolo Internet y Redes de la próxima generación*. Aspectos del protocolo Internet – Calidad de Servicio y Características de red. Recuperado de <https://www.itu.int/rec/T-REC-Y.1541/es>
- Llontop, D. G. (2015). *Propuesta de diseño para la implementación de un servidor VoIP con Asterisk y Raspberry Pi en una oficina de Villa el Salvador*. (tesis de grado). Recuperado de <http://repositorio.untels.edu.pe//handle/123456789/416>
- Matango, F. (23 de septiembre de 2016). *Protocolos de Señalización de la VoIP* [publicación en blog]. Recuperado de <http://www.serverVoIP.com/blog/protocolos-de-senalizacion-de-la-VoIP/>
- Matango, F. (26 de agosto de 2016). *ServerVoIP* [publicación en blog]. Recuperado de <http://www.serverVoIP.com/blog/tag/calidad-servicio/>
- Matango, F. (3 de agosto 2016). *Costos sobre beneficios Voz sobre IP* [publicación en blog]. Recuperado de <http://www.serverVoIP.com/blog/tag/costos-telefonía-ip/>
- Murthy, C. (2010). *Data Communication and Networking*. Recuperado de <http://www.himpub.com/BookDetail.aspx?BookId=2870&NB=>
- Nfon. (7 de septiembre de 2021). *¿Qué ancho de banda para VoIP necesitas contratar en tu empresa?* [publicación en blog] Recuperado de <https://blog.nfon.com/es/ancho-de-banda-voip/>
- Ochoa, F. O. (21 de mayo de 2019). *Luego de 29 años ¿Cómo ha cambiado la telefonía? La batalla ganada por el móvil sobre las otras telefonías*. *Gestión*. Recuperado de <https://gestion.pe/economia/empresas/luego-29-anos-cambiado-telefonía-batalla-ganada-movil-telefonías-267308-noticia/?ref=ges>

- Olave, A., Valencia, L. y Cuéllar, J. (2014). Análisis basado en parámetros QoS para el servicio VoIP sobre wireless en un campus universitario. *Ventana Informática 1(31)*, 11-23. Recuperado de https://repository.icesi.edu.co/biblioteca_digital/bitstream/10906/79546/1/cuellar_analisis_basado_2014.pdf
- Osiptel. (1 de agosto de 2021). *OSIPTEL adecúa reglamento de calidad para garantizar el 70% de velocidad mínima en servicio de internet de banda ancha*. Recuperado de <https://www.osiptel.gob.pe/portal-del-usuario/noticias/osiptel-adecua-reglamento-de-calidad-para-garantizar-el-70-de-velocidad-minima-en-servicio-de-internet-de-banda-ancha/>
- Pastor, J. (25 de abril de 2018). *Raspberry Pi 3 Model B+, análisis: más potencia y mejor WiFi para un miniPC que sigue asombrando*. Recuperado de <https://www.xataka.com/ordenadores/Raspberry-pi-3-model-b-analisis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando>
- Pérez, L. J. (5 de septiembre de 2010). *Marco Metodológico* [publicación en blog] Recuperado de http://asesoriatesis1960.blogspot.pe/2010/09/asesoria-de-tesis-trabajos-de-grado-e_05.html
- Querea (2019) *Voz datos IP y telefonía IP*. Recuperado de http://www.quarea.com/es/Asterisk_funcionalidades_basicas_avanzadas.
- Redacción Gestión. (25 de marzo del 2016). El 25% de líneas telefónicas en el Perú operan con tecnología VoIP. *Gestión*. Recuperado de <https://gestion.pe/tecnologia/25-lineas-telefonicas-peru-operan-tecnologia-voip-112653-noticia/>
- Redacción Gestión. (6 de mayo de 2015). La evolución de la telefonía. *Gestión*. Recuperado de <https://gestion.pe/economia/sueno-tenerlo-casa-llevar-bolsillo-evolucion-telefonias-152905-noticia/?ref=gesr>
- Rivera, C. P. y Poma, N. B. (2015). *Diseño e Implementación de centrales telefónicas de voz sobre para prácticas de análisis de tráfico, señalización, protocolos de conmutación y troubleshooting VoIP para uso en el laboratorio de telecomunicaciones* (tesis de grado) Recuperado de <https://dspace.ups.edu.ec/handle/123456789/6556>
- Russoniello, A., Gil, A. y Samaniego, C. (30 de marzo 2015). *Implementación de Voz sobre IP con Asterisk en Raspberry Pi*. Recuperado de https://www.academia.edu/24465410/Implementaci%C3%B3n_de_Voz_sobre_IP_con_Asterisk_en_Raspberry_Pi
- Salazar, G. (30 de septiembre de 2016). *Fundamentos de QoS – Calidad de servicio en Capa 2 y Capa 3*. Recuperado de <https://community.cisco.com/t5/blogs-routing-y->

switching/fundamentos-de-QoS-calidad-de-servicio-en-capa-2-y-capa-3/ba-
p/3103715

Sangoma. (2016). *FreePBX – The World’s Most Popular Open-source IP-PBX Software*.
Recuperado de <https://www.sangoma.com/products/FreePBX/>

Sinaeepourfard, A., & Mohamed Hussain, H. (2011). Comparison of VoIP and PSTN services by
statistical analysis. *2011 IEEE Student Conference on Research and Development*, 459-
461. Recuperado de [https://www.semanticscholar.org/paper/Comparison-of-VoIP-
and-PSTN-services-by-statistical-Sinaeepourfard-
Hussain/5736393c02fcf0d3bac4579df08942d86b845b54](https://www.semanticscholar.org/paper/Comparison-of-VoIP-and-PSTN-services-by-statistical-Sinaeepourfard-Hussain/5736393c02fcf0d3bac4579df08942d86b845b54)

Villa, J. (2007). *Calidad de Servicio (QoS)* [publicación en blog]. Recuperado de
<http://asteriskelite.blogspot.com/2007/08/calidad-de-servicio-qos.html>

Wikibooks. (2015). *Introducción a la VOZIP*. Recuperado de
[https://es.wikibooks.org/wiki/Comunicaciones_Unificadas_Con_Elastix/Introducci%C
3%B3n_a_la_VOZIP](https://es.wikibooks.org/wiki/Comunicaciones_Unificadas_Con_Elastix/Introducci%C3%B3n_a_la_VOZIP)

GLOSARIO

IP:	Protocolo de Internet
VOIP:	Voice over IP, Voz sobre IP
PBX:	Private Branch Exchange, Central privada automática
PSTN:	Public Switched Telephone Network, Red telefónica conmutada pública
IRC:	Internet Relay Chat, protocolo de comunicación
SMS:	Short Message Servicio, servicio de mensajería
UDP:	User Datagram Protocol, Protocolo de datagramas de usuario
TCP:	Transmission Control Protocol, Protocolo de control de transmisión
SIP:	Session Initiaton Protocol, Protocolo de inicio de sesión
IAX:	Inter- <i>Asterisk</i> eXchange, protocolo original de <i>Asterisk</i>
H.323:	Conjunto de normas ITU
MGCP:	Media <i>Gateway</i> Control Protocol, protocolo de control de dispositivos
LAN:	Local Area Network, Red de area local
WAN:	Wide Area Network, Red de area global
<i>Raspberry Pi:</i>	Hardware y <i>software</i> de escritorio basado en Linux
<i>Asterisk:</i>	<i>Software</i> para telefonía
<i>FreePBX:</i>	<i>Software</i> para control de <i>Asterisk</i>
ITSP:	Proveedor de servicios de internet

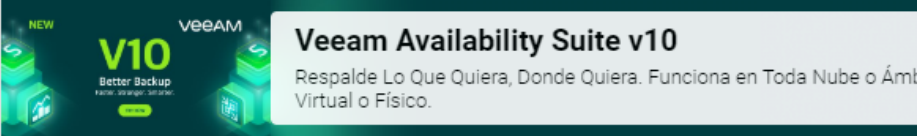
ANEXOS

ANEXO 1: Instalación de *RasPBX*

Se descarga la imagen ISO desde la página principal de *Asterisk* para *Raspberry Pi*

No es seguro | raspberry-asterisk.org/downloads/

Home Documentation Downloads FAQ Forum About Blog



Downloads

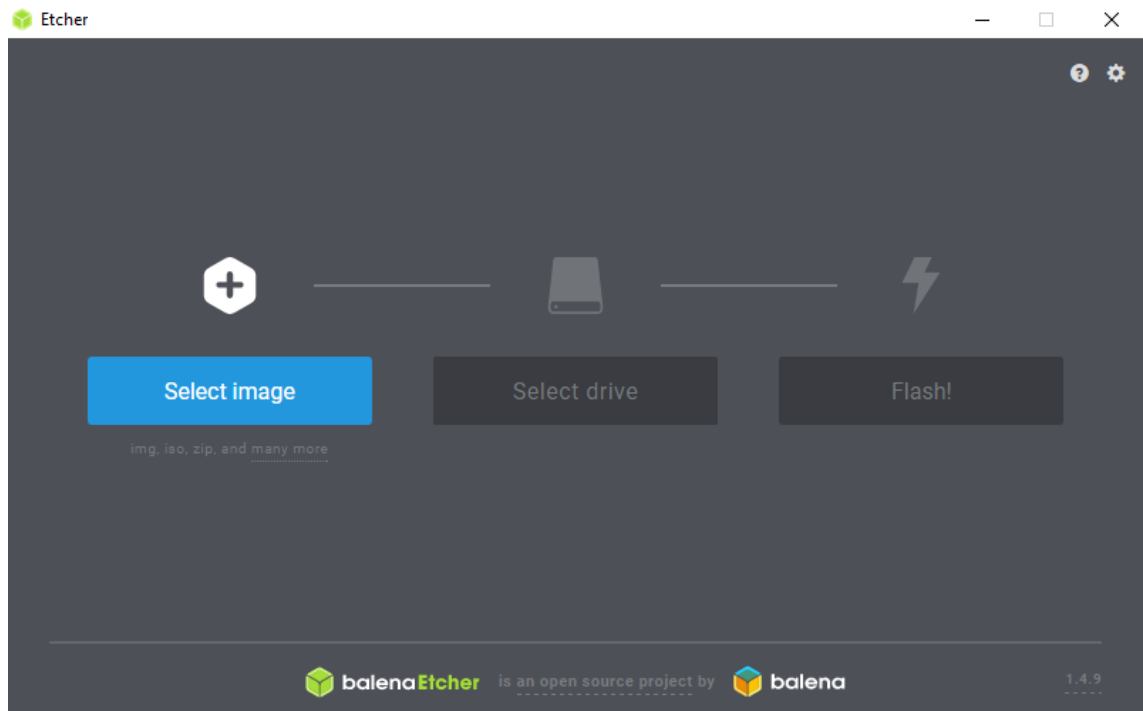
RasPBX images based on Raspbian 10 Buster:

The latest image supports Pi 4, Pi 3 and Pi 2 (Pi 1 and Pi zero no longer supported):

Torrent	raspbx-11-11-2019.zip.torrent
HTTP	raspbx-11-11-2019.zip
SHA-1	3d403f60c9f08fae96de0a7dbc7df6b7834d84df
Contents	Asterisk 16.6.1 & FreePBX 15.0.16.22

An 8GB card or larger is recommended.

Con la imagen descargada, se pasas a instalar el sistema operativo en micro SD Card, el tamaño mínimo recomendado es 8GB. Para instalar el SO en la tarjeta, existen varios programas, en este caso se usa Balena de Etcher.io:



Los pasos son simples, se selecciona la imagen, luego la tarjeta de memoria y por ultima, flash. Al finalizar, el sistema está listo para usarse.

ANEXO 2: Configuración Inicial de *RasPBX*

Al iniciar el sistema, la configuración es por defecto. Es útil, sin embargo, es necesario realizar algunas configuraciones y actualizaciones. Algunos procesos requieren internet, por lo que, si no se puede conectar por un cable de red, se usa la red inalámbrica mediante el comando `sudo-config`.

Actualizar sistema:

Sudo apt-get update

Sudo apt-get upgrade

Crear nuevas claves de host ssh

Regen-hostkeys

Elegir zona horaria y configuraciones locales

Configure-timezone

Dpkg-reconfigure locales

Configuraciones de teclado

Dpkg-reconfigure keyboard-configuration

ANEXO 3: Configuración del archivo *extensions.conf*

Una vez 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 las 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)

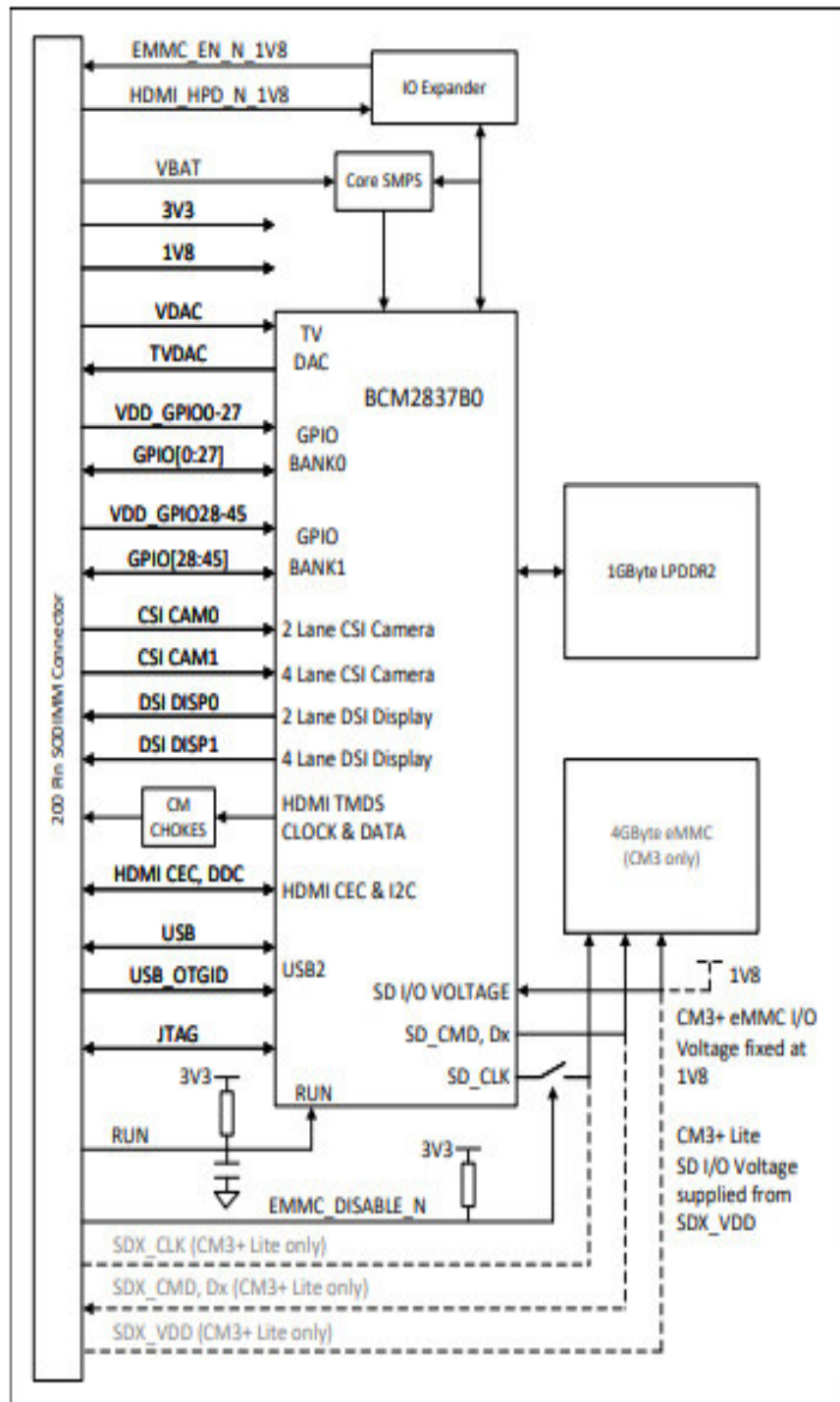
exten => s,1,Hangup(21)

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

ANEXO 4: Imágenes de Equipos



3 Block Diagram





A simple and reliable IP Phone

GXP1610/1615

A simple IP phone for small business users, the GXP1610/1615 delivers a user-friendly VoIP calling experience in a very easy-to-use IP phone. The GXP1610/1615 offers support for 1 line, 2 call appearances and includes 3-way voice conferencing to maximize productivity. Additional features include a 132x48 (2.95") LCD screen for easy viewing, 3 XML programmable soft keys for customization, 10/100 mbps ports, integrated PoE on the GXP1615 model, EHS support for Plantronics headsets and multi-language support. These features allow the GXP1610/GXP1615 to be a high-quality small business IP phone that is simple and easy-to-use.



1 SIP account, up to 2 call appearances



TLS and SRTP security encryption technology to protect calls and accounts



3-way audio conferencing for easy conference calls



Electronic Hook Switch (EHS) support for Plantronics headsets



Automated provisioning options include TR-069 and XML config files



Full-duplex speakerphone with HD audio to maximize audio quality and clarity



Use with Grandstream's UCM series of IP PBXs for Zero Config provisioning



Built-in PoE 802.3af to power the device and give it a network connection (GXP1615 only)