



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**Diseño e implementación de una herramienta generadora de llamadas
automáticas para medir la calidad del servicio de un proveedor
internacional de telefonía VoIP.**

AUTOR:

Grijalva Arias, Leslie Elizabeth

Trabajo de Titulación previo a la obtención del título de
INGENIERO EN TELECOMUNICACIONES

TUTOR:

M. Sc. Zamora Cedeño, Nestor Armando

Guayaquil, Ecuador

11 de Marzo del 2019



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Srta.
Grijalva Arias, Leslie Elizabeth como requerimiento para la obtención del
título de **INGENIERO EN TELECOMUNICACIONES**.

TUTOR

M. Sc. Zamora Cedeño, Nestor Armando

DIRECTOR DE CARRERA

M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, a los 11 días del mes de Marzo del año 2019



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Grijalva Arias, Leslie Elizabeth**

DECLARO QUE:

El trabajo de titulación “**Diseño e implementación de una herramienta generadora de llamadas automáticas para medir la calidad del servicio de un proveedor internacional de telefonía VoIP**” previo a la obtención del Título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 11 del mes de Marzo del año 2019

AUTOR

GRIJALVA ARIAS, LESLIE ELIZABETH



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Grijalva Arias, Leslie Elizabeth**

Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **“Diseño e implementación de una herramienta generadora de llamadas automáticas para medir la calidad del servicio de un proveedor internacional de telefonía VoIP”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 11 del mes de Marzo del año 2019

EL AUTOR

GRIJALVA ARIAS, LESLIE ELIZABETH

REPORTE DE URKUND

URKUND

Documento: Titulacion-Grijalva.docx (D48129881)
Presentado: 2019-02-20 09:15 (-05:00)
Presentado por: Néstor Zamora (nestor.zamora@cu.ucsg.edu.ec)
Recibido: nestor.zamora.ucsg@analysis.orkund.com
Mensaje: Trabajo Titulación Grijalva [Mostrar el mensaje completo](#)
0% de estas 32 páginas, se componen de texto presente en 0 fuentes.

Lista de fuentes Bloques

Categoría	Enlace/nombre de archivo
>	Titulacion-Grijalva.docx
Fuentes alternativas	
	Titulacion-Grijalva.docx
	03-11-15 TESIS IPF.docx
	Tesis Laboratorio IP - Maestría Teleco.docx

1 Advertencias. Reiniciar. Exportar. Compartir

100% #1 Activo **Archivo de registro Urkund: Universidad Católica de Santiago de Guayaquil / Titulacion-Grijalva.docx 100%**

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA: Diseño e implementación de una herramienta generadora de llamadas automáticas para medir la calidad del servicio de un proveedor internacional de telefonía VoIP.

AUTOR: Grijalva Arias, Leslie Elizabeth

Trabajo de Titulación previo a la obtención del título de INGENIERO EN TELECOMUNICACIONES

TUTOR: M. Sc. Zamora Cedeño, Nestor Armando

Guayaquil, Ecuador

9 de Marzo del 2019

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA: Diseño e implementación de una herramienta generadora de llamadas automáticas para medir la calidad del servicio de un proveedor internacional de telefonía VoIP.

AUTOR: Grijalva Arias, Leslie Elizabeth

Trabajo de Titulación previo a la obtención del título de INGENIERO EN TELECOMUNICACIONES

TUTOR: M. Sc. Zamora Cedeño, Nestor Armando

Guayaquil, Ecuador

9 de Marzo del 2019

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

DEDICATORIA

De todo corazón a mis queridos padres, quienes han sido y serán mi fuente de inspiración, dándome fuerzas cuando pensaba rendirme, quienes continuamente brindan su apoyo moral, espiritual y emocional. A mi hermano y hermana, mentores y amigos que compartieron sus consejos y estímulos para terminar este trabajo de titulación.

A Dios, gracias por la orientación, la fuerza, la protección, las destrezas y una vida saludable.

Todo esto, es para ustedes.

EL AUTOR

Grijalva Arias, Leslie Elizabeth

AGRADECIMIENTO

A Dios por guiar mis pasos hasta este objetivo.

A mis padres por su esfuerzo, sacrificio y por nunca dejar de creer en mí.

A mis hermanos que siempre me alientan a seguir adelante.

A los docentes por las enseñanzas impartidas

A mis amigos por la ayuda incondicional.

Gracias infinitas.

EL AUTOR

Grijalva Arias, Leslie Elizabeth



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

f. _____

M. Sc. ROMERO PAZ, MANUEL DE JESÚS
DECANO

f. _____

M. Sc. PALACIOS MELÉNDEZ, EDWIN FERNANDO
COORDINADOR DE ÁREA

f. _____

M. Sc. BOHÓRQUEZ ESCOBAR, CELSO BAYARDO
OPONENTE

Índice General

Índice de Figuras	XII
Índice de Tablas.....	XIV
Resumen.....	XV
Abstract.....	XVI
CAPÍTULO 1: DESCRIPCIÓN GENERAL.....	2
1.1. Introducción.....	2
1.2. Antecedentes.	3
1.3. Definición del Problema.	4
1.4. Justificación del Problema.	4
1.5. Objetivos del Problema de Investigación.	4
1.5.1. Objetivo General.....	4
1.5.2. Objetivos Específicos.	5
1.6. Hipótesis.	5
1.7. Metodología de Investigación.	5
CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA.....	6
2.1. VoIP.	6
2.2. Beneficios de VoIP.....	7
2.2.1. Incremento de la productividad.	8
2.2.2. Eficiencia empresarial.	9
2.2.3. Reducción de costos.....	10
2.2.4. Simplificación de la gestión administrativa.....	11
2.2.5. Mejora de la productividad.	12
2.3. Componentes de VoIP.	12
2.4. Protocolos VoIP.	14
2.4.1. Protocolo de Inicio de Sesión.....	15
2.4.1.1. Visión general de las operaciones de SIP.....	16
2.4.1.1.1. Direccionamiento SIP.....	16
2.4.1.1.2. Localización de un servidor SIP.	16
2.4.1.1.3. Transacción SIP.....	16
2.4.1.1.4. Invitación SIP.	17
2.4.1.1.5. Localización de un usuario.....	17
2.4.1.1.6. Cambiando una sesión existente.....	17

2.4.1.1.7. El servidor salto.....	17
2.4.1.2. Códigos de respuesta SIP.....	18
2.5. Análisis de VoIP.....	19
2.5.1. Retardo (Delay).....	19
2.5.2. Fluctuación de fase (Jitter).....	20
2.5.3. Pérdida de paquetes.....	21
2.5.4. Eco.....	23
2.5.5. Rendimiento.....	23
2.6. Mediciones de calidad de servicio VoIP.....	24
2.6.1. Método subjetivo.....	24
2.6.1. Método objetivo (intrusivo y no intrusivo).....	24
2.7. Desafíos de seguridad de VoIP.....	25
2.8. Asterisk en VoIP.....	27
2.8.1. Arquitectura de Asterisk.....	28
2.8.1.1. Conmutación PBX.....	29
2.8.1.2. Iniciador de aplicaciones.....	29
2.8.1.3. Traductor de códecs.....	29
2.8.1.4. Módulo cargable de API'S.....	29
2.8.1.5. Canal API.....	30
2.8.1.6. Aplicación API.....	30
2.8.1.7. Traductor Códecs API.....	30
2.8.1.8. Formato de archivo API.....	30
2.8.2. Servicios que proporciona Asterisk.....	31
CAPÍTULO 3: SIMULACIÓN Y RESULTADOS OBTENIDOS.....	33
3.1. Asterisk.....	33
3.2. Asignación de recursos.....	33
3.3. Configuraciones principales.....	34
3.3.1. Permisos de administrador.....	34
3.3.2. Paquetes de desarrollo.....	34
3.3.3. Instalación de Jansson.....	35
3.3.4. Instalación de Asterisk.....	36
3.3.5. Instalación de dependencia adicionales para Asterisk.....	37
3.3.6. Habilitación de servicio Asterisk.....	38
3.3.7. Deshabilitación de firewall.....	39

3.3.8.	Generación de CDRs en Asterisk.....	40
3.3.9.	Instalación de ODBC.....	41
3.3.10.	Configuración de ODBC y MySQL.....	42
3.3.11.	Configuración de Asterisk para usar ODBC y MySQL.....	42
3.3.12.	Configuración Res_odbc.....	43
3.3.13.	Creación de la tabla de base de datos.....	43
3.4.	Interfaz de usuario.....	44
3.5.	Interfaz de usuario envío de la llamada.....	44
3.5.1.	Interfaz de resultados para el usuario.....	48
3.6.	Pruebas y resultados.....	51
3.6.1.	Análisis de señalización.....	53
3.6.2.	Calidad de llamada.....	54
3.6.2.1.	Puntuación de opinión media.....	54
3.6.2.2.	Análisis de calidad.....	54
3.6.2.3.	Llamada fallida.....	56
3.6.3.	Resultados.....	57
CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.....		58
4.1.	Conclusiones.....	58
4.2.	Recomendaciones.....	58
Bibliografía.....		59

Índice de Figuras

Capítulo 2

Figura 2. 1: Componentes de VoIP.	13
Figura 2. 2: Arquitectura de protocolo de VoIP sobre WLAN IEEE 802.11..	14
Figura 2. 3: Clasificación de los métodos de medición de calidad de voz..	25
Figura 2. 4: Perspectiva general de un sistema basado en Asterisk.....	28

Capítulo 3

Figura 3. 1: Asignación de recursos de máquina virtual de Google Cloud... 33	33
Figura 3. 2: Permisos de administrador..... 34	34
Figura 3. 3: Instalación de epel-release..... 35	35
Figura 3. 4: Descarga de Jansson..... 35	35
Figura 3. 5: Instalación de Jansson..... 36	36
Figura 3. 6: Instación de Asterisk..... 37	37
Figura 3. 7: Instalación de dependencias adicionales..... 38	38
Figura 3. 8: Inicialización de Asterisk..... 38	38
Figura 3. 9: Revisión del estado de Asterisk..... 39	39
Figura 3. 10: Deshabilitación en el sistema..... 39	39
Figura 3. 11: Deshabilitación en el archivo de configuración..... 39	39
Figura 3. 12: Generación de CDRS en Asterisk..... 40	40
Figura 3. 13: Instalación segura MySQL..... 40	40
Figura 3. 14: Configuración de usuario y base de datos..... 41	41
Figura 3. 15: Comprobación de permisos para el inicio de sesión..... 41	41
Figura 3. 16: Instalación de ODBC..... 41	41
Figura 3. 17: Configuración de ODBC y de MySQL..... 42	42
Figura 3. 18: Configuración MySQL..... 42	42
Figura 3. 19: Configuración de Asterisk para usar ODBC y MySQL..... 43	43
Figura 3. 20: Configuración Res_odbc..... 43	43
Figura 3. 21: Creación de la tabla de base de datos..... 44	44
Figura 3. 22: Interfaz de usuario envío de la llamada..... 45	45
Figura 3. 23: Configuración de estilos, fondo y margen de la página..... 45	45

Figura 3. 24: Configuración de tabla y botones.	46
Figura 3. 25: Configuración de estilo de caja de texto, tabulación y botones. Elaborado por: Autor.....	46
Figura 3. 26: Limpieza de caracteres en la caja de texto y de ingreso de números. Elaborado por: Autor.....	47
Figura 3. 27: Código generador de llamadas y variables de conexión.	47
Figura 3. 28: Generador de bloque y alerta.	48
Figura 3. 29: Controladores de página.	48
Figura 3. 30: Interfaz de usuario.	49
Figura 3. 31: Función altRows.	49
Figura 3. 32: Configuración de estilo de tabla.	50
Figura 3. 33: Conexión a la base de datos.	50
Figura 3. 34: Presentación de información.	51
Figura 3. 35: Recarga de página.	51
Figura 3. 36: Primer escenario de prueba.	52
Figura 3. 37: Mensaje “Número procesado correctamente”.	52
Figura 3. 38: Muestra de resultados.	53
Figura 3. 39: Análisis de señalización de la llamada de prueba.	53
Figura 3. 40: Análisis de calidad.	55
Figura 3. 41: Análisis de señalización del segundo escenario de prueba. ...	55
Figura 3. 42: Análisis de calidad dentro del segundo escenario de prueba. ...	56
Figura 3. 43: Análisis de señalización del escenario de llamada perdida. ...	56
Figura 3. 44: Análisis de calidad dentro del escenario de llamada perdida..	57

Índice de Tablas

Capítulo 2

Tabla 2. 1: Códigos de respuesta SIP.	18
Tabla 2. 2: Códecs de VoIP estándar de retardo de una vía y tolerancia de pérdida.	20

Capítulo 3

Tabla 3. 1: Comparación de valores de MOS.....	54
Tabla 3. 2: Resultados dentro de los 3 escenarios de prueba.	57

Resumen

Es posible que una infraestructura de comunicación VoIP deba servir a un número sustancial de usuarios con una calidad de voz aceptable. El presente trabajo de titulación investiga la idoneidad y calidad del servicio de proveedores internacionales de telefonía de voz sobre IP mediante el servidor PBX de Asterisk para proporcionar capacidades de comunicación con una calidad aceptable de Puntuación de Opinión Media (MOS), utilizado para evaluar la calidad de las llamadas de voz, Jitter promedio y Pérdida de Paquetes promedio, para un gran número de usuarios. Los resultados muestran la idoneidad de tres escenarios: llamada exitosa, llamada sin respuesta por el usuario final y llamada rechazada por el proveedor, para caracterizar la capacidad del servidor PBX de Asterisk. Los resultados experimentales han demostrado que el PBX de Asterisk puede manejar efectivamente gran cantidad de llamadas de voz, mientras que proporciona un MOS promedio superior a 4.

Palabras claves: ASTERISK, VOIP, LLAMADA, HERRAMIENTA, PROGRAMACIÓN, TELEFONÍA.

Abstract

It is possible that a VoIP communication infrastructure should serve a substantial number of users with acceptable voice quality. The present work investigates the suitability and quality of the service of international providers of telephony of voice over IP by means of the PBX server of Asterisk to provide communication capacities with an acceptable quality of Mean Opinion Score (MOS), used to evaluate the quality of voice calls, average Jitter and average packet loss, for a large number of users. The results show the suitability of three scenarios: successful call, unanswered call by the end user and call rejected by the provider, to characterize the capacity of the Asterisk PBX server. Experimental results have shown that Asterisk PBX can effectively handle a large number of voice calls, while providing an average MOS greater than 4.

Keywords: ASTERISK, VOIP, CALL, TOOL, PROGRAMMING, TELEPHONY.

CAPÍTULO 1: DESCRIPCIÓN GENERAL

1.1. Introducción.

Este proyecto plantea el diseño e implementación de una herramienta generadora de llamadas automáticas para medir la calidad de servicio de un proveedor internacional de telefonía VoIP, la cual propone una nueva alternativa para las empresas encargadas de enrutar llamadas locales al extranjero, el fin del uso de la herramienta es utilizar la tecnología VoIP con el propósito de automatizar las pruebas manuales. La herramienta basará su programación en Asterisk, software de código abierto, que permite crear un sistema de telefonía de acuerdo con las diversas necesidades.

Las pequeñas y medianas empresas han estado experimentando los beneficios de los sistemas telefónicos de Asterisk durante muchos años. Todos los sistemas telefónicos Asterisk son totalmente compatibles con VoIP y, a menudo, eliminan la necesidad de un PBX tradicional costoso, una solución práctica para las organizaciones más pequeñas que buscan obtener los beneficios de un PBX corporativo pero que carecen del capital para obtener uno (Masjedi, 2013).

Si bien Asterisk fue diseñado inicialmente para ejecutarse en una plataforma Linux, es totalmente compatible con la mayoría de las plataformas modernas. Hay varios beneficios ventajosos asociados con el uso de Asterisk junto con una solución VoIP para pequeñas empresas de alta calidad. No solo es extremadamente rentable en comparación con la mayoría de las otras opciones de PBX, sino que también cuenta con muchas más características y funciones que los competidores. Dado que los sistemas telefónicos de Asterisk son completamente dependientes de VoIP, utilizan las conexiones de Internet de banda ancha existentes para aprovechar la red de telefonía corporativa. Cuando se usan los productos Asterisk, las llamadas de voz aún se hacen y reciben de la misma manera que lo harían a través de un PBX convencional, pero dadas las capacidades de VoIP, los asteriscos permiten que los empleados remotos y en tránsito se

conecten directamente al sistema telefónico corporativo a través de una conexión activa a Internet (Masjedi, 2013).

Asterisk admite una gran variedad de protocolos VoIP y es totalmente compatible con los servicios VoIP para pequeñas empresas de Nextiva. La mayoría de los teléfonos IP modernos son totalmente interoperables con Asterisk, lo que permite que el sistema se integre de manera transparente y rentable con las empresas establecidas. La opción de interconectarse con líneas telefónicas convencionales ciertamente existe; sin embargo, es más probable que las empresas vean valor cuando se usan en un escenario de VoIP para pequeñas empresas (Masjedi, 2013).

1.2. Antecedentes.

Asterisk tiene un largo camino por recorrer, es un entorno de código abierto, que se convierte en estándar para el cambio de voz y las funciones de PBX. Asterisk ayuda a construir una plataforma PBX flexible con hardware de bajo costo, que es posible mediante el procesamiento de host de TDM y DSP (Meggelen, Madsen, & Smith, 2014).

Asterisk es una plataforma eficiente que soporta múltiples protocolos de paquetes de voz y proporciona una aplicación PBX de alto nivel. El refuerzo de las redes IAXTE y la comunidad de código abierto ayudan a Asterisk a permanecer confiable y actualizado con la última tecnología (CounterPath USA, 2018). Al mismo tiempo, debido al escaso interés comercial de Asterisk, está creciendo lentamente en algunas áreas.

Asterisk permite crear un PBX que se compone de las funciones, características y calidad de los conmutadores de telefonía tradicionales. En comparación con Asterisk, otros PBX son difíciles de implementar, caros y no estandarizados. Asterisk es flexible para manejar redes de voz y datos con bajo costo y mantenimiento (Datarkar, Bobade, & Gaikwad, 2015). Anteriormente experimentos realizados, mostraron el éxito que tiene este tipo de aplicaciones evaluadoras. En el laboratorio de voz sobre IP de la Universidad Federal de Rio de Janeiro, según el artículo de 2008 "ACME: An

Automated Tool for Generating and Evaluating the Quality of VoIP Calls”, elaboraron un software denominado ACME, el cual examina la disponibilidad del ancho de banda y la capacidad de procesamiento para determinar la mayor cantidad de llamadas simultáneas que un nodo puede operar en una topología determinada, monitoreando así su calidad y eficiencia (Lustosa, Souza, de A. Rodrigues, & Quinellato, 2008).

1.3. Definición del Problema.

Necesidad de las empresas que brindan servicios de enrutamiento de llamadas internacionales de diseñar e implementar una herramienta generadora de llamadas automáticas para medir y comparar la calidad del servicio de proveedores internacionales de telefonía VoIP.

1.4. Justificación del Problema.

Actualmente existen diversos tipos de empresas de telecomunicaciones tanto privadas como públicas que, como toda compañía, tienen un propósito lucrativo, esta es la razón por lo que buscan desarrollar herramientas en las cuales la inversión sea siempre mas baja y así realizar un mejor trabajo cumpliendo con diversos estándares de calidad.

Es en esta situación que el diseño e implementación de una herramienta generadora de llamadas que analice la calidad de estas a través de tecnología telefónica VoIP, es de necesidad y gran importancia, para que de esta manera todos los servicios brindados por los diversos proveedores y empresas internacionales sean de excelencia.

1.5. Objetivos del Problema de Investigación.

1.5.1. Objetivo General.

Diseñar una herramienta que permita generar llamadas automáticamente hacia la red de un proveedor de telefonía IP para medir la calidad de servicio que ofrece, mediante el uso de Asteriks.

1.5.2. Objetivos Específicos.

- Diseñar un generador de llamadas automáticas empleando el software Asteriks.
- Analizar los diferentes tipos de problemas que puede presentar una red de telefonía VoIP.
- Realizar pruebas técnicas para medir la calidad del servicio de un proveedor internacional de telefonía VoIP, mediante la aplicación diseñada.

1.6. Hipótesis.

El diseño e implementación de una herramienta generadora de llamadas automáticas para medir la calidad del servicio de proveedores internacionales de telefonía VoIP mediante el uso de Asterisk, software de PBX, que se utiliza para establecer y controlar las llamadas telefónicas entre endpoints de redes de telecomunicaciones, será el fundamental evaluador de estos sistemas y de la misma forma beneficiará tanto a las compañías que brindan su servicio, como a los usuarios finales que emplean los mismos.

1.7. Metodología de Investigación.

La metodología empleada en este trabajo de titulación es exploratoria, debido a que se examina el protocolo de voz a través de internet (VoIP), sus beneficios, componentes, arquitectura y desafíos, con el fin de familiarizarnos con dichos temas, se utiliza también una metodología descriptiva pues se detalla el funcionamiento de los diferentes elementos de Asterisk y VoIP, y los procesos para la evaluación de calidad de las llamadas generadas para su posterior estudio y análisis con el fin de desarrollar un algoritmo completo y preciso.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA

2.1. VoIP.

El protocolo de voz a través de Internet (VoIP) es una forma de transmisión que le permite a cualquier persona realizar llamadas telefónicas a través de una conexión a internet de banda ancha. Por lo general, el acceso VoIP le permite al usuario llamar a otras personas que también reciben llamadas a través de Internet. Las conexiones VoIP interconectadas también permiten a los usuarios hacer y recibir llamadas hacia y desde números de líneas fijas convencionales, generalmente por un cargo adicional del servicio. Se utiliza un tipo de adaptador en algunos servicios de VoIP que requieren una computadora y un teléfono VoIP dedicado. VoIP también se puede describir como una solución distinta que permite la transmisión de señales de voz a través de una conexión a Internet en lugar de la línea telefónica tradicional (Sonkar, Singh, Chauhan, & Singh, 2013).

En las implementaciones de VoIP de hoy, las señales analógicas de voz se muestrean y codifican mediante un códec que luego se encapsulan en un paquete IP y se transfieren a través de cables de datos o la infraestructura de Internet de la misma manera que se transportan los paquetes de datos (Phithakkitnukoon, Dantu, & Baatarjav, 2015).

Anteriormente, VoIP requería que se conectaran unos auriculares a la computadora, y el altavoz y el receptor solo podían hablar con otras personas que tenían una configuración similar. Tuvieron que informarse mutuamente con anticipación, a fin de señalar al usuario al otro extremo de la llamada entrante y el momento de la llamada (Sanneck, 2016).

En noviembre de 1977, el Grupo de trabajo de ingeniería de Internet publicó las Especificaciones para el NVP (protocolo de voz de red). El documento ilustra el desarrollo y la demostración de la viabilidad de la seguridad y comunicaciones digitales de voz a voz en dúplex completo, en tiempo real y en tiempo real, a través de redes de comunicación de datos de conmutación de paquetes (Sonkar et al., 2013).

A mediados de los años 90, las redes IP estaban creciendo, la tecnología había avanzado y ha habido un amplio uso de computadoras personales. La creencia de que VoIP podría tener un impacto significativo en el mercado dio lugar a altas expectativas que dieron como resultado la distribución del primer paquete de software (Sonkar et al., 2013).

En sus primeras etapas, la tecnología VoIP no estaba completamente desarrollada y había muchas lagunas. Hubo una gran brecha entre la estructura de marketing y la realidad tecnológica. Esto puede concluirse que la escasez técnica detuvo cualquier desarrollo importante o cambios en VoIP. Sin embargo, últimamente VoIP ha seguido haciendo progresos tecnológicos y viables. Los protocolos de señalización se utilizan para configurar y eliminar llamadas, transportar datos necesarios para localizar usuarios y negociar capacidades (Sanneck, 2016).

Una de las ventajas clave de VoIP es hacer llamadas de larga distancia a precios muy económicos, que incluyen llamadas a otros países con la flexibilidad de usar el mismo número en diferentes partes del mundo (Ghafarian & Jackson, 2014).

Básicamente, el sistema VoIP se puede configurar en estos modos de conexión respectivamente; PC a PC, Teléfono a Teléfono y PC a Teléfono. Además, el teléfono puede ser de tipo digital o analógica. En el caso de un teléfono analógico, se conectará al sistema mediante adaptadores que convertirán las señales analógicas en formato digital (Yong-feng & Jiang-ling, 2015).

2.2. Beneficios de VoIP.

Hay muchos beneficios de pasar a VoIP, algunos de los cuales ni siquiera se ofrecen como opción con TDM, tecnología del teléfono basada en circuitos eléctricos que están físicamente encendidos en la red telefónica pública conmutada (PTSN) (BullsEye Telecom, 2016). El costo para mantener y administrar la infraestructura de TDM se refleja tanto a nivel de cliente como a nivel de operador. Los costos operativos de los proveedores

de servicios VoIP son más bajos que los de las compañías telefónicas TDM tradicionales, que deben lidiar con una arquitectura de telefonía costosa de mantener y regulaciones costosas. Los costos más bajos permiten que el proveedor de VoIP cobre menos al tiempo que brinda más funciones (CenturyLink Advanced Technology Solutions, 2016).

Muchas de las funciones y servicios están integradas en la plataforma VoIP o están incluidos en los negocios que ahorran en el proveedor y no gastan en la misma funcionalidad a nivel empresarial. Con TDM, se paga mensualmente por muchas de las funciones que necesita, como identificador de llamadas, desvío de llamadas, llamadas a tres vías y más. En contraste, los proveedores de VoIP incluyen estas características en el costo mensual del servicio. Estos pueden ser implementados usuario por usuario o ubicación por ubicación. Dentro de los principales beneficios de uso de VoIP se tiene: (CenturyLink Advanced Technology Solutions, 2016).

2.2.1. Incremento de la productividad.

Los servicios de VoIP hacen que las capacidades de comunicación de voz sean más flexibles. Permiten el ahorro de costos y las funciones de productividad que TDM tiene problemas para igualar:

- **Siempre disponible:** Hoy en día, muchas ofertas de VoIP están configuradas para que suenen simultáneamente en múltiples dispositivos, como el celular facilitando el acceso a clientes y socios. Lograr esta característica de "siempre disponible" con un TD requeriría líneas adicionales para reenviar la llamada a un dispositivo secundario y posiblemente a un equipo PBX adicional.
- **Habilita una fuerza de trabajo virtual:** Siempre que los usuarios tengan acceso a una conexión de banda ancha, pueden continuar manejando llamadas fuera de las paredes físicas de la oficina, incluso mientras viaja, mediante software instalado en PC y dispositivos móviles que permiten a los usuarios hacer y recibir llamadas. Las personas que llaman de forma externa pueden comunicarse con los empleados utilizando el número de teléfono de la compañía, mientras mantienen el número de teléfono móvil del empleado en privado.

- **Correo de voz por correo electrónico:** Con VoIP, el correo de voz se puede reenviar automáticamente al buzón de correo electrónico del usuario como un archivo adjunto de formato wav o transcrito y enviado como texto.
- **Combina y simplifica:** La estandarización de VoIP elimina la necesidad de que los empleados dominen múltiples sistemas telefónicos a medida que se mueven en la organización. Los empleados pueden ser receptivos y productivos desde el primer día, con menos capacitación requerida.
- **Portabilidad:** Una vez que un empleado tiene un número de teléfono, ese número puede seguirlo en toda la empresa. Si un empleado se muda a otra ubicación física, ya sea dentro del edificio o a otra ubicación, puede tomar su teléfono, conectarlo a un puerto LAN y, junto con el número, seguirán todas las características y capacidades. Esto ahorra tiempo al personal de TI, ya que no hay necesidad de mover físicamente el equipo y la administración del sistema se da a través de un portal basado en la web del proveedor (CenturyLink Advanced Technology Solutions, 2016).

2.2.2. Eficiencia empresarial.

VoIP puede ayudar a las empresas al aumentar la eficiencia en las operaciones de TI:

- **Números virtuales en cualquier lugar:** Muchas empresas están expandiendo sus servicios a mercados donde no tienen presencia física. Con VoIP, puede tener un número de teléfono local con un código de área que se desee seleccionar. No necesitas un sistema físico en esa ubicación. La línea VoIP puede conectar llamadas a una ubicación en una región diferente del país. Debido a la rigidez y las regulaciones de los TDM, se debe comprar un número de teléfono local en ese mercado que requiera una forma de reenvío de llamadas que redirija la llamada a otro dispositivo o ubicación. En estos casos, hay un costo mensual en cada mercado y, a menudo, un costo por minuto para cada llamada procesada.

- **Migración y despliegue más sencillos:** VoIP se puede implementar con poco costo porque opera en su LAN y WAN existentes. La realización de evaluaciones de red puede identificar lo que se necesita para introducir VoIP en su ecosistema. La migración a VoIP es fácil de implementar. Se trata de configurar el servicio con un proveedor de VoIP y migrar a la arquitectura de LAN y WAN convergente. VoIP puede coexistir con un entorno TDM, con la migración coordinada a lo largo del tiempo (CenturyLink Advanced Technology Solutions, 2016).

2.2.3. Reducción de costos.

- **Ahorros mensuales continuos:** Las facturas de TDM se basan en tarifas y servicios regulados. Con VoIP, puede llamar a otras compañías en la misma red sin cargos adicionales. Si se encuentra en un área del país donde se usa un minuto de uso existen cargos por llamadas locales, este costo no se aplica cuando se utiliza VoIP.
- **Mejor utilización del ancho de banda:** Durante una conversación de voz hay un promedio de 35-50% de silencio durante cada llamada, al pasar a VoIP, se pueden aplicar estos ahorros de utilización para que más llamadas puedan usar la misma cantidad de ancho de banda. Durante esos períodos de silencio, el ancho de banda está disponible en la red para otras comunicaciones de voz, datos o video.
- **Administración de costos de recursos técnicos:** Hoy en día, la mayoría de las PBX TDM y otro hardware de voz requieren que una empresa tenga uno o más recursos técnicos disponibles para realizar actividades en el equipo. Para algunos de estos sistemas, los recursos disponibles son limitados ya que es posible que el fabricante ya no sea compatible con el equipo, por lo que es posible que deba pagar una prima por el mantenimiento y los técnicos para respaldar su entorno. El teléfono es otro dispositivo y se administra utilizando los recursos técnicos de datos. Ya no es necesario que contrate, capacite y administre al personal de TI en TDM. Con VoIP, esto puede liberar esos recursos de TI para centrarse en los esfuerzos estratégicos alineados con los objetivos centrales del negocio (CenturyLink Advanced Technology Solutions, 2016).

2.2.4. Simplificación de la gestión administrativa.

- **Arquitectura de red convergente:** En la actualidad, muchas empresas tienen redes de voz y datos separadas. En general, estos son gestionados por recursos separados. Con VoIP, el teléfono se convierte en otro dispositivo en la infraestructura de datos; por lo tanto, ahora hay una sola red para administrar. La mayoría de los teléfonos VoIP disponibles en la actualidad tienen capacidades adicionales, lo que permite que la PC se conecte a la parte posterior del teléfono, compartiendo así una única conexión LAN para ambos dispositivos utilizando VLAN (LAN virtual) para separar y priorizar el tráfico.
- **Continuidad de negocio:** Es común que un entorno de datos tenga algún nivel de redundancia; este aspecto de la continuidad del negocio ahora puede extenderse a la arquitectura de VoIP. Con varias opciones de nivel de red y de sitio disponibles, tiene la opción de crear muchos niveles de redundancia. Este ahorro ya no requiere una duplicación de circuitos TDM y equipos PBX para crear una estrategia de voz de conmutación por error.
- **Rápido aumento de capacidad:** Utilizando su red de datos, procesar todas las comunicaciones permite a una empresa aumentar rápidamente la capacidad de su red. Las empresas pueden reasignar dinámicamente el ancho de banda existente de los datos a la voz para adaptarse a los aumentos en el volumen de llamadas, un proceso que no es posible con los circuitos de voz tradicionales de TDM.
- **Escalabilidad:** Dado que VoIP utiliza su conectividad de datos, la expansión y la disminución del ancho de banda de datos se puede lograr rápidamente con su operador para acomodar los picos y valles del tráfico de voz a lo largo del año. Esto significa horas/días para agregar ancho de banda en lugar de las semanas requeridas en la red TDM. Además, no se requerirá ningún gasto de PBX para adaptarse a este crecimiento del tráfico, ni gastos de técnicos para realizar el trabajo en la PBX, ni gastos de equipo ni costos adicionales de administración o mantenimiento (CenturyLink Advanced Technology Solutions, 2016).

2.2.5. Mejora de la productividad.

- **Soporte mejorado para trabajadores remotos:** Trabajadores remotos hoy se conectan a la red de datos mediante VPN o Internet. Con las aplicaciones VoIP ejecutándose en la red de datos, amplía la capacidad de usar este acceso para comunicaciones de voz también.
- **Integración de aplicaciones de datos:** Existe una gran demanda empresarial para una mayor colaboración e integración de telefonía con herramientas de datos. A los desarrolladores de aplicaciones de red les resultará más fácil desarrollar e implementar aplicaciones emergentes para comunicación mediante VoIP, ya que se basa en estándares de la industria bien establecidos en el mercado actual. Esto permite que la empresa se expanda y use nuevas aplicaciones más rápidamente.
- **Soporte de video:** El video no se admite fácilmente en la mayoría de las redes locales de TDM sin comprar un circuito diseñado para ese propósito. Con VoIP, se permite el video punto a punto, siempre que los usuarios tengan una aplicación compatible, una PC con capacidades de video o un dispositivo con capacidad de video en cada extremo (CenturyLink Advanced Technology Solutions, 2016).

2.3. Componentes de VoIP.

VoIP consta de tres componentes esenciales: CÓDEC (Codificador/Decodificador), empaquetador y búfer de reproducción (Cai, Xiao, (Sherman) Shen, Cai, & W. Mark, 2014). En el lado del remitente, una muestra adecuada de señales de voz analógicas se convierte en señales digitales, se comprimen y luego se codifican en un formato predeterminado utilizando el códec de voz. Existen varios códecs de voz desarrollados y estandarizados por la Unión Internacional de Telecomunicaciones (UIT-T) como G.711, G.729, G.723.1a, etc. A continuación, se realiza un proceso de paquetización que fragmenta la voz codificada en el mismo tamaño de los paquetes. Además, en cada paquete, algunos encabezados de protocolo de diferentes capas se adjuntan a la voz codificada. Los encabezados de los protocolos que se agregan a los paquetes de voz son el Protocolo de transporte en tiempo real (RTP), el Protocolo de datagramas del usuario (UDP) y el Protocolo de Internet (IP), así como el encabezado de la capa de

enlace de datos. Además, RTP y el Protocolo de control en tiempo real (RTCP) se diseñaron en la capa de aplicación para admitir aplicaciones en tiempo real. Aunque el protocolo de transporte TCP se usa comúnmente en Internet, el protocolo UDP es preferido en VoIP y otras aplicaciones en tiempo real sensibles al retraso. El protocolo TCP es adecuado para paquetes de datos menos sensibles a la demora y no para paquetes sensibles a la demora debido al esquema de acuse de recibo (ACK) que aplica TCP. Este esquema introduce demora ya que el receptor tiene que notificar al remitente para cada paquete recibido enviando un ACK. Por otro lado, UDP no aplica este esquema y, por lo tanto, es más adecuado para aplicaciones de VoIP (Kazemitabar, Ahmed, & Nisar, 2013).

Luego, los paquetes se envían a través de la red IP a su destino, donde se lleva a cabo el proceso inverso de decodificación y despaquetización de los paquetes recibidos. Durante el proceso de transmisión, pueden ocurrir variaciones en el tiempo de entrega de paquetes (jitter). Por lo tanto, se utiliza un búfer de reproducción en el extremo del receptor para suavizar la reproducción, mitigando el jitter incurrido. Los paquetes se ponen en cola en el búfer de reproducción durante un tiempo de reproducción antes de ser reproducidos. Sin embargo, los paquetes que llegan más tarde que el tiempo de reproducción se descartan. Los componentes principales de un sistema VoIP, que cubre la transmisión de voz de extremo a extremo, se ilustran en la figura 2.1 (Kazemitabar et al., 2013).

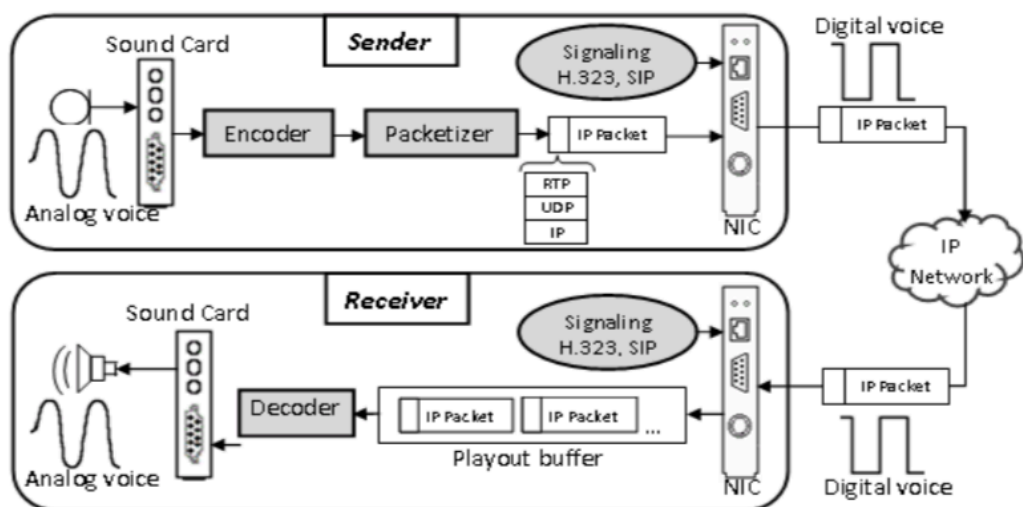


Figura 2. 1: Componentes de VoIP.
Fuente: (Kazemitabar, Ahmed, & Nisar, 2013).

Además, hay protocolos de señalización de VoIP, como, Protocolo de Inicio de Sesión (SIP) y H.323. Estos protocolos de señalización se requieren desde el principio para establecer llamadas VoIP y al final para cerrar las transmisiones de medios entre los clientes (Swale, 2015). H.323 fue estandarizado por ITU-T específicamente para trabajar sin problemas junto con PSTN, mientras que SIP fue estandarizado por Grupo de Trabajo de Ingeniería de Internet (IETF) para soportar aplicaciones de internet como la telefonía. En la figura 2.2, se ilustra la pila de protocolo VoIP. Además, en las redes IP, las direcciones IP pueden cambiarse de una sesión a otra, especialmente en casos de acceso telefónico. Por lo tanto, existe la necesidad de un punto de encuentro común compartido entre los usuarios para que puedan encontrarse en la etapa de establecimiento de la comunicación. Este punto de reunión común se conoce genéricamente como un servidor de llamadas.

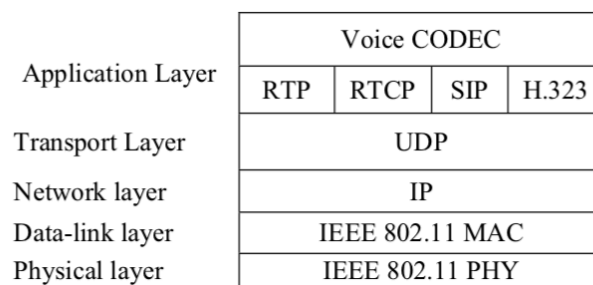


Figura 2. 2: Arquitectura de protocolo de VoIP sobre WLAN IEEE 802.11.
Fuente: (Kazemitabar, Ahmed, & Nisar, 2013).

2.4. Protocolos VoIP.

Hay una serie de protocolos que pueden emplearse para proporcionar los servicios de comunicación VoIP. Se discutirá principalmente en los aspectos más comunes para la mayoría de los dispositivos ya implementados o que se implementan en la actualidad.

Prácticamente todos los dispositivos en el mundo utilizan un estándar llamado Real-Time Protocol (RTP) para transmitir paquetes de audio y video entre computadoras que se comunican. RTP está definido por el Grupo de Trabajo de Ingeniería de Internet (IETF) en la Solicitud de comentarios (RFC) 3550. Los formatos de carga útil para varios CÓDEC se definen en RFC 3551, aunque las especificaciones de formato de carga útil se definen

en varios documentos de la UIT y en otros IETF RFCs. RTP también aborda problemas como el orden de un paquete y proporciona mecanismos (a través del Protocolo de control en tiempo real, o RTCP, definido en RFC 3550) para ayudar a resolver el retraso (delay) y la fluctuación de fase (jitter) (Kumar, 2013).

El Protocolo H.323 y el Protocolo de Inicio de Sesión (SIP) tienen sus orígenes en 1995 cuando los investigadores buscaron resolver el problema de cómo dos computadoras pueden iniciar la comunicación para intercambiar secuencias de audio y video. H.323 disfrutó del primer éxito comercial, debido al hecho de que aquellos que trabajan en el protocolo en la UIT trabajaron rápidamente para publicar el primer estándar a principios de 1996. SIP, por otro lado, progresó mucho más lentamente en el IETF, con el primer borrador publicado en 1996, pero el primer "estándar" reconocido se publicó más tarde en 1999. SIP se revisó a lo largo de los años y se volvió a publicar en 2002 como RFC 3261, que es el estándar actualmente reconocido para SIP. Estos retrasos en el proceso de estándares resultaron en demoras en la adopción de mercado del protocolo SIP (Kumar, 2013).

En la actualidad, H.323 aún controla la mayor parte de las implementaciones de VoIP en el mercado de proveedores de servicios para el tránsito de voz, especialmente para el transporte internacional de llamadas de voz (Packetizer Inc., 2019). H.323 también se usa ampliamente en sistemas de videoconferencia y es el protocolo principal para sistemas de video basados en IP. SIP, más recientemente, se ha vuelto más popular para su uso en sistemas de mensajería instantánea (Kumar, 2013).

2.4.1. Protocolo de Inicio de Sesión.

El Protocolo de inicio de sesión (SIP) es un protocolo de control de señalización de nivel de aplicación que se utiliza para establecer, mantener y finalizar sesiones multimedia. Las sesiones multimedia incluyen VoIP, conferencias y otras aplicaciones similares relacionadas con medios como audio, video y datos (Kumar, 2013).

SIP, en el que se basa el RFC 2543, es un protocolo basado en texto que forma parte de la arquitectura multimedia global del Grupo de Trabajo de Ingeniería de Internet (IETF). La arquitectura IETF también incluye el Protocolo de reserva de recursos (RSVP; RFC 2205), el Protocolo de Transporte en Tiempo Real (RTTP; RFC1889) y el Protocolo de Descripción de Sesión (SDP; RFC 2327). Sin embargo, las funciones del SIP son independientes de cualquier función de estos protocolos (TechTarget Inc., 2019). Es importante tener en cuenta que SIP puede funcionar junto con otros protocolos de señalización, como H.323 (Kumar, 2013).

2.4.1.1. Visión general de las operaciones de SIP.

Las personas que reciben y realizan llamadas se identifican por las direcciones SIP. Al realizar una llamada SIP, la persona que llama primero debe localizar el servidor apropiado y enviar una solicitud. La persona que llama puede comunicarse directamente con el destinatario o indirectamente a través de los servidores de redirección. El campo ID de llamada en el encabezado del mensaje SIP identifica de forma única las llamadas. A continuación se muestra una breve explicación de cómo el protocolo realiza sus operaciones (RFC 2543) (Kumar, 2013).

2.4.1.1.1. Direccionamiento SIP.

Los hosts SIP se identifican mediante una URL SIP, que tiene la forma *sip:username@host*. Una dirección SIP puede designar un individuo o un grupo completo (Kumar, 2013).

2.4.1.1.2. Localización de un servidor SIP.

El cliente puede enviar la solicitud a un servidor proxy SIP o puede enviar la solicitud directamente a la dirección IP y al puerto correspondiente al Identificador de Solicitud Uniforme (URI) (Kumar, 2013).

2.4.1.1.3. Transacción SIP.

Una vez que la parte del host del URI de solicitud se ha resuelto en un servidor SIP, el cliente puede enviar solicitudes a ese servidor. Una solicitud junto con las respuestas desencadenadas por esa solicitud conforma una

transacción SIP. Las solicitudes se pueden enviar a través de TCP confiable o de UDP no confiable (TechTarget Inc., 2019).

2.4.1.1.4. Invitación SIP.

Una invitación SIP exitosa consta de dos solicitudes: una *INVITACIÓN* seguida de *ACK*. La solicitud de *INVITACIÓN* le pide a la persona que llama que se una a una conferencia en particular o que establezca una conversación entre dos partes. Una vez que la persona que llama ha aceptado participar en la llamada, la persona que llama confirma que ha recibido esa respuesta mediante el envío de una solicitud *ACK*. La solicitud *INVITACIÓN* contiene una descripción de la sesión que proporciona a la parte llamada información suficiente para unirse a la sesión (TechTarget Inc., 2019). Si la persona que llama desea aceptar la llamada, responde a la invitación devolviendo una descripción de sesión similar (Kumar, 2013).

2.4.1.1.5. Localización de un usuario.

Las personas que reciben una llamada pueden seguir cambiando su posición con el tiempo. Estas ubicaciones se pueden registrar dinámicamente con el servidor SIP. Cuando se consulta al servidor SIP sobre la ubicación de la persona que llama, devuelve una lista de posibles ubicaciones. Un servidor de ubicación en el sistema SIP en realidad genera la lista y la pasa al servidor SIP (Kumar, 2013).

2.4.1.1.6. Cambiando una sesión existente.

A veces es posible que se tenga que cambiar los parámetros de una sesión existente. Esto se hace volviendo a emitir el mensaje *INVITACIÓN* usando la misma ID de llamada, pero un nuevo cuerpo para transmitir la nueva información (Kumar, 2013).

2.4.1.1.7. El servidor salto.

Una vez que se ha alcanzado el Servidor de Agente de Usuario (UAS), envía una respuesta al servidor proxy. El servidor proxy a su vez envía una respuesta al cliente. El cliente luego confirma que ha recibido la respuesta enviando un *ACK*. En este caso, se asume que la solicitud *INVITACIÓN* del

cliente se reenvió al servidor proxy. Sin embargo, si se había reenviado a un servidor de redirección, entonces el servidor de redirección devuelve la dirección IP del servidor del siguiente salto al cliente y luego se comunica directamente con el UAS (Kumar, 2013).

2.4.1.2. Códigos de respuesta SIP.

En SIP, cada elemento de red se identifica mediante un URI de SIP (Identificador uniforme de recursos) tal como como una dirección. Se desglosan como: respuestas provisionales 1xx, respuestas exitosas 2xx, respuestas de redirección 3xx, respuestas de fallo de cliente 4xx, respuestas de fallo de servidor 5xx y respuestas de fallo global 6xx.

La tabla 2.1 resume la serie de códigos de respuesta de SIP y su función correspondiente.

Tabla 2. 1: Códigos de respuesta SIP.

RESPUESTAS SIP		
TIPO DE CÓDIGO	CÓDIGO	FUNCIÓN
Códigos 1XX son informativos: Solicitud recibida / continuando el proceso de la solicitud.	100	Tratando
	180	Sonando
	181	La llamada está siendo reenviada
	182	En cola
Códigos 2XX indican éxito: Las solicitudes fueron recibidas exitosamente, entendidas y aceptadas.	200	Ok
	202	Aceptado
Códigos 3XX son de redirección: El proxy no quiere manejar la llamada. Estas respuestas son usualmente enviadas a los servidores proxy.	301	Movido permanentemente
	302	Movido temporalmente
	305	Usar Proxy
Códigos 4XX indican falla del cliente: La solicitud del cliente no es aceptada y no puede ser procesada por ese servidor.	400	Solicitud incorrecta
	404	Usuario no encontrado
	407	Autenticación de proxy requerida
Códigos 5XX indican falla en el servidor: Indican un problema encontrado en el servidor. La solicitud es aparentemente válida pero no puede ser cumplida. Los clientes deberían intentar la petición más tarde.	500	Error interno del servidor
	501	No implementado
	502	Pasarela incorrecta
Códigos 6XX indica falla global: La solicitud no puede ser cumplida por ningún servidor.	600	Ocupado
	604	No existe en ninguna parte

Fuente: (Matango, 2016)

2.5. Análisis de VoIP.

Para crear un diseño de red adecuado, es importante conocer todas las limitaciones y el funcionamiento interno de la tecnología de redes. A continuación, se explica muchos de los problemas que enfrenta VoIP.

2.5.1. Retardo (Delay).

Puede definirse como el tiempo total que tarda una persona en comunicarse, hablar y escuchar en el otro extremo. A diferencia de las aplicaciones de datos, las aplicaciones de VoIP son muy sensibles al retraso, aunque pueden tolerar la pérdida de paquetes en cierta medida. El retardo de extremo a extremo o de boca a oreja es uno de los principales factores que afectan a la QoS (Calidad de Servicio) y debe ser inferior a 150 ms para una buena conexión de red, tal como lo define la UIT G.114 (Xiuzhong Chen et al., 2013), mientras que el Instituto de Estándares de Telecomunicaciones Europeas (ETSI) definen un retardo de menos de 100 ms (ETSI). Principalmente, el retraso es causado por la congestión de la red que conduce a una entrega lenta del paquete (Sahabudin & Alias, 2014). Además, el retardo se ve afectado por varios parámetros o algoritmos que se pueden clasificar en, retardo en la fuente, retardo en el receptor y retardo en la red:

- **Retardo en la fuente:** El retraso de todo el proceso realizado en el lado del remitente antes de transmitir el paquete de voz a través de la red está causado por varios componentes: códec, paquetización y proceso (Ahson & Ilyas, 2009). Las funciones de códec introducen algunos retrasos al procesar la conversión de analógico a digital. Códec también puede realizar técnicas de compresión para la eficiencia en el uso del ancho de banda. Existe una compensación entre la eficiencia del ancho de banda y el retardo de compresión. Cuantos más bits se comprimen, menos ancho de banda se requiere, y mayor será el retraso agregado. Por lo tanto, los códecs agregan un retardo adicional al retardo total de transmisión de paquetes. La tabla 2.2 muestra algunos de los códecs estándar del UIT-T y sus retrasos (Kazemitabar et al., 2013). Para el retraso de la paquetización, es el tiempo que se tarda en colocar los trozos de tramas en paquetes que

se transmitirían a través de la red. El tercer componente del retraso de la fuente es cuando la computadora pasa los paquetes a la red para transmitirlos al otro lado.

- **Retardo en el receptor:** El proceso inverso que se lleva a cabo en el remitente se realiza en el receptor, lo que agrega más retardo: retardo del proceso y retardo de decodificación, incluido el retardo de descompresión. Además, se produce un retraso en la reproducción cuando se produce el flujo de voz que también incluye el retraso del búfer por fluctuaciones de fase (Kazemitabar et al., 2013).
- **Retardo de la red:** El retardo de la red en el entorno WLAN es el retardo total tanto de la red WLAN como de la red troncal. La cola, la transmisión y la propagación son otros componentes de los retrasos de la red. Los paquetes tienen que alinearse en algunos nodos intermedios a lo largo del curso de transmisión, como el nodo enrutador que causa el retraso en la cola. El retardo de propagación es el retardo en los medios físicos de la red, mientras que el retardo de transmisión incluye el retardo del enrutador y el retardo de retransmisión de MAC (Jing Liu & Zhisheng Niu, 2014).

Tabla 2. 2: Códecs de VoIP estándar de retardo de una vía y tolerancia de pérdida.

Codec standard	Compression Technique	Bit Rate (Kbps)	Encoding Delay (ms)	Loss Tolerance (%)
G.711	Pulse Code Modulation (PCM)	64 (uncompressed)	0.13 (negligible)	7-10
G.726	Adaptive Differential PCM (ADPCM)	24, 32, or 40	0.4	5
G.729	Conjugate Structure Algebraic Code-Excited Linear Prediction (CS-ACELP)	8	about 25	<2
G.723.1	Algebraic Code Excited Linear Prediction (ACELP)	6.4 Kbps or 5.3	about 67.5	<1

Fuente: (Kazemitabar, Ahmed, & Nisar, 2013).

2.5.2. Fluctuación de fase (Jitter).

La red IP no garantiza el tiempo de entrega de paquetes, lo que introduce variaciones en el retardo de transmisión. Esta variación se conoce

como fluctuación de fase (Habib, Nirmala, & Nirmala, 2018) y tiene más efectos negativos en la calidad de la voz (Cai et al., 2014).

Dado que los paquetes de voz del mismo flujo no se reciben al mismo tiempo. Por lo tanto, se introducen búferes de fluctuación de fase para disminuir el efecto de jitter y hacer que la conversación sea fluida, ya que contiene una serie de paquetes en una cola antes de la reproducción. El tamaño de la cola del búfer puede ser fijo o adaptable, que varía según la condición de la red, el carácter de voz (Jing Liu & Zhisheng Niu, 2014), para un mejor rendimiento. Las técnicas adaptativas de fluctuación de fase del búfer funcionan mejor, ya que reduce la posibilidad de desbordamiento y desbordamiento del búfer

El problema de desbordamiento es cuando la cantidad de paquetes recibidos se hace más grande que el tamaño del búfer, ya que el búfer resultante descarta los paquetes que no pueden contener. Por otro lado, el búfer de subflujo ocurre cuando se necesitan algunos paquetes para la reproducción, pero el búfer está vacío. Por lo tanto, se introducen varios algoritmos de búfer adaptativo, como en la referencia, para ajustar el tamaño del búfer con el fin de mejorar la calidad de VoIP (McNeill, Liu, & Rodriguez, 2016).

2.5.3. Pérdida de paquetes.

Los paquetes transmitidos a través de la red IP pueden perderse en la red o llegar dañados o con retraso. Los paquetes se descartarían cuando llegasen tarde al búfer de fluctuaciones del receptor o cuando haya un desbordamiento en el búfer de fluctuaciones o en el búfer del enrutador. Por lo tanto, la pérdida de paquetes es la pérdida total que se produce debido a la congestión de la red y la llegada tardía (McNeill et al., 2016). En caso de pérdida de paquetes, se informa al remitente para que retransmita los paquetes perdidos y esto causaría más demora y, por lo tanto, afectaría la QoS de la transmisión (Kazemitabar et al., 2013).

Además, el sistema VoIP puede tolerar la pérdida de paquetes hasta cierto punto, ya que el 1% o menos es aceptable para la calidad, mientras que para la calidad empresarial es aceptable el 3% o menos (Karapantazis & Pavlidou, 2015). Por lo tanto, más del 3% de la pérdida de paquetes degrada la calidad del habla. En consecuencia, se introdujeron varias técnicas de recuperación de pérdida y ocultación de errores.

Se han diseñado técnicas y algoritmos para resolver problemas de pérdida de paquetes. La subcapa MAC en el lado del remitente utiliza el esquema de acuse de recibo para retransmitir paquetes perdidos. La técnica de redundancia ha sido diseñada donde los paquetes se envían varias veces, pero esta técnica consume ancho de banda. Además, la corrección de errores hacia adelante (FEC) es una técnica matemática que ayuda al receptor a reconstruir el paquete perdido de los paquetes enviados previamente, ya que un paquete consiste en cierta información del paquete anterior (Kazemitabar et al., 2013).

Por otro lado, los códecs en la capa superior emplean sus propias técnicas de encubrimiento de errores para recuperar paquetes perdidos. Las técnicas utilizadas por el códec se denominan ocultación de pérdida de paquetes (PLC). De hecho, existe un intercambio entre complejidad y calidad en las técnicas de pérdida de paquetes; la técnica más compleja es la menor tasa de pérdida de paquetes. Se ha propuesto una técnica que hace uso de las técnicas de recuperación disponibles. El esquema presentado cambia dinámicamente el esquema de recuperación de acuerdo con la tasa de pérdida de paquetes (Nasr & Napoleon, 2014).

Además, la técnica de fragmentación en la capa MAC también desempeña un papel en el problema de la pérdida de paquetes debido a la relación entre el tamaño del paquete y la tasa de pérdida de paquetes. El paquete con un tamaño más grande tiene mayor probabilidad de ser descartado que aquellos con un tamaño pequeño en caso de errores en el paquete (Korhonen & Ye Wang, 2015). Por lo tanto, las técnicas de

fragmentación deben diseñarse cuidadosamente considerando su efecto en la tasa de pérdida de paquetes.

2.5.4. Eco.

En VoIP, eco se experimenta cuando la persona que llama en el lado del remitente escucha un reflejo de su voz después de hablar por teléfono o por el micrófono, mientras que la persona que llama no nota el eco. Eco es el término de los reflejos de las señales de voz enviadas por el extremo lejano. El eco puede ser el eco eléctrico que existe en las redes PSTN o el eco acústico que es un problema en las redes VoIP (Mintandjian & Naylor, 2016). El eco eléctrico se genera debido a la conversión entre las 2 líneas de cable y otro circuito de otro tipo en un entorno híbrido. Mientras que el acoplamiento acústico entre el micrófono y el altavoz del teléfono provoca el eco acústico. El eco se ve afectado por el retardo y se vuelve perceptible cuando es fuerte y se retrasa principalmente cuando el retardo de ida y vuelta es más de 50 ms (Habib et al., 2018). Como el retardo de ida y vuelta de VoIP puede exceder los 100 ms, se utilizan los compensadores de eco.

2.5.5. Rendimiento.

Este parámetro se refiere al número máximo de bits recibidos del número total de bits enviados durante un intervalo de tiempo. En las redes IEEE 802.11, la velocidad de bits de cada estándar se especifica, por ejemplo, IEEE 802.11b se transmite a 1Mbps, 2Mbps, 5.5Mbps y 11Mbps. Además, el estándar tiene la capacidad de realizar un cambio de velocidad dinámico con el objetivo de mejorar el rendimiento del enlace inalámbrico (Kazemitabar et al., 2013).

Sin embargo, el rendimiento de transmisión es aún menor que el especificado en el estándar debido a la sobrecarga introducida por los protocolos de red WLAN. El rendimiento alcanzado se encuentra entre el rango del 50% al 70% de la velocidad de transmisión, que es bajo en comparación con el rendimiento de Ethernet, que alcanza del 80% al 90% de la velocidad de transmisión (Ni, Romdhani, & Turletti, 2014). Se han

introducido métodos para mejorar el rendimiento de WLAN, como la técnica de agregación y las técnicas adaptativas (Ergen, Ergen, & Varaiya, 2013).

2.6. Mediciones de calidad de servicio VoIP.

Los métodos de medición de la calidad del habla para VoIP se dividen en métodos subjetivos y objetivos, como se exponen a continuación:

2.6.1. Método subjetivo.

Los métodos subjetivos apuntan a encontrar la percepción del usuario promedio de la calidad del habla de un sistema preguntando a un panel de oyentes humanos y brindándoles una opción de respuesta limitada. La Recomendación UIT-T en la Recomendación P.800 introdujo la Puntuación de Opinión Media (MOS) basada en la percepción del usuario que varía entre 1 (deficiente) y 5 (excelente) para la determinación subjetiva de la calidad de la voz (Kazemitabar et al., 2013).

2.6.1. Método objetivo (intrusivo y no intrusivo).

Los métodos objetivos se basan en las matemáticas que miden las cantidades físicas del sistema (es decir, la demora y la pérdida de paquetes) (Karapantazis & Pavlidou, 2015). Los métodos objetivos incluyen dos técnicas:

- **Intrusivo:** Este método generalmente utiliza la inyección de señal de prueba en la red y durante este tiempo la red no puede transportar tráfico en vivo. Se pueden aplicar métodos intrusivos durante el desarrollo del sistema antes de la presencia de usuarios (Reynolds & Rix, 2014).

- **No intrusivo:** Este método se ejecuta sin señal de referencia y se ha desarrollado para aplicarse en el tráfico en tiempo real. Por lo general, este tipo de método realiza su predicción directamente a partir de parámetros de degradación de la red como jitter, retardo y pérdida de paquetes (Rango, Tropea, Fazio, & Marano, 2016).

Los métodos de medición de QoS se ilustran en la figura 2.3.

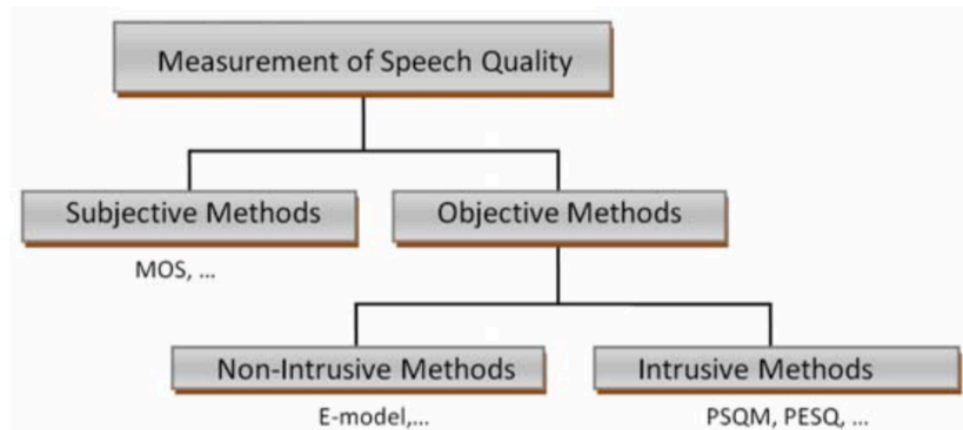


Figura 2. 3: Clasificación de los métodos de medición de calidad de voz.
Fuente: (Kazemitabar, Ahmed, & Nisar, 2013).

2.7. Desafíos de seguridad de VoIP.

En comparación con PSTN, VoIP enfrenta muchos problemas de seguridad y PSTN proporciona un alto nivel de seguridad debido a sus caracteres (Garuba, Li, & Yi, 2013), pero algunos de los problemas son similares para ambos, como el tipo de ataques y el fraude de llamadas. "Man in the middle" es un tipo de escuchas ilegales en las que escuchar y cambiar de conversación es posible para el atacante. El atacante puede interrumpir las conversaciones y luego reproducir la conversación anterior, en lugar de cambiar la conversación de "sí" a "no" o tomar ventaja al pedir el nombre de usuario y la contraseña (Butcher, Li, & Guo, 2014). En el fraude de llamadas telefónicas, la motivación es hacer llamadas gratuitas, especialmente para llamadas de larga distancia de alto costo (Hung & Martin, 2016), en realidad las llamadas fraudulentas se facturan a una víctima, y generalmente es posible debido a la falta de configuraciones.

Dado que Internet es inseguro y es una red subyacente para VoIP, algunas amenazas y riesgos se heredan de esta red a VoIP. El riesgo más importante es el ataque de denegación de servicio (DoS), que intenta hacer que los recursos no estén disponibles o impedir que un servicio funcione bien, lo que varía en cuanto a rendimiento y motivación. El ataque de VoIP DoS tuvo como objetivo hacer caer un servicio. VoIP ejecuta la transmisión de paquetes IP por voz, por ejemplo, los hackers pueden mantener la transmisión de paquetes en ejecución o pueden hacer que el sistema esté

ocupado con la inundación de solicitudes de llamadas. En consecuencia, harán ineficiencia en el uso de recursos, escucharán señales de ocupado a largo plazo y forzarán la desconexión de llamadas debido a la falta de recursos (Garuba et al., 2013).

Otra amenaza para la seguridad es SPIT (Spam sobre telefonía por Internet), que solo afecta a VoIP, como VoIP. El spam está relacionado con las bandejas de entrada del correo de voz, de hecho, es similar al correo basura en el buzón, los spammers enviarán muchos mensajes no deseados y anuncios de audio para llenar el buzón de entrada de correos de voz para que así se necesite capacidad de almacenamiento de voz adicional (Kazemitabar et al., 2013).

Otro problema, como el Protocolo fuzzing, pertenece a las aplicaciones IP (Garuba et al., 2013). Fuzzing genera datos de entrada aleatorios o secuenciales para inyectarlos en una aplicación que normalmente utiliza fuzzing para la depuración y prueba de aplicaciones IP (Abdelnur, State, & Festor, 2015). En el protocolo fuzzing (mensajes mal formados) el objetivo es un protocolo como el protocolo de pila SIP. El atacante utiliza este método para encontrar vulnerabilidades y crea varios tipos de paquetes que apuntan al sistema de destino para procesar estos paquetes en lugar del paquete normal, y como resultado, el sistema se enfrentará a fallas, desbordamiento de búfer, retrasos de aplicaciones y bucles infinitos.

VoIP a través de WLAN y la red inalámbrica utiliza un medio abierto para la comunicación, en tal red se requiere la autenticación del tráfico de VoIP. Desde un punto de vista técnico inalámbrico, el cifrado debe implementarse para proteger los datos y recursos contra accesos y modificaciones no autorizados. Las soluciones disponibles se muestran como:

- El aislamiento de la red VoIP corporativa de la internet pública mejorará la seguridad y el rendimiento de la red. Debido a este hecho, la incorporación de túneles VPN se sugiere (Stanton, 2016). Se pueden aplicar filtros de correo no deseado para analizar los mensajes SPIT

basados en voz para busca de contenido prohibido. Algunos problemas de seguridad pueden controlarse utilizando IPSec (un marco de trabajo para asegurar la comunicación cifrando cada paquete). Firewall y NAT es otro mecanismo de seguridad bien conocido.

- A pesar de la existencia de poderosas tecnologías de prevención, VoIP aún enfrenta muchos desafíos de seguridad que deben considerarse en cualquier implementación. La implementación de mecanismos de seguridad, como el cifrado, conllevará un mayor retraso debido al cifrado y al descifrado, además de crear nuevos encabezados a paquetes (Karapantazis & Pavlidou, 2015), por lo que tiene un efecto en la calidad del servicio y la capacidad, es así que debe considerarse antes de ajustar y desarrollar políticas de seguridad.

2.8. Asterisk en VoIP.

Asterisk es un PBX completo en software escrito en lenguaje de programación C y se ejecuta en sistemas operativos Linux. Asterisk maneja VoIP en muchos protocolos, y puede interoperar con casi todos los equipos de telefonía basados en estándares utilizando hardware relativamente barato como para tarjetas ex PCI. De hecho, Asterisk crea un PBX que rivaliza con las funcionalidades de los sistemas telefónicos tradicionales (Alam, Bose, Rahman, & Al-Mumin, 2007).

Una gran parte de la literatura sobre Asterisk revela que es diferente por muchas razones, la más importante es que se trata de un enfoque de software. En lugar de cambiar las líneas analógicas en hardware, enruta y manipula paquetes de Voz sobre Protocolos de Internet (VoIP) en el software. La columna vertebral del sistema generalmente se convierte en una red habilitada para IP, y los teléfonos pueden conectarse a esa red. Sin embargo, también admite teléfonos analógicos antiguos que utilizan dispositivos de puerta de enlace (Department of Business Administration, East West University, Bangladesh, Masudur Rahman, & Sarwar Islam, 2014). Asterisk proporciona más de lo que se podría esperar de un PBX convencional. Los usuarios obtienen una variedad de funciones, como la paginación (que puede ser de uno a uno o de muchos a uno, según los

requisitos de uso), respuestas de voz interactivas (IVR), conferencia, correo de voz, música en espera, etc. (Imran, Qadeer, & Khan, 2015).

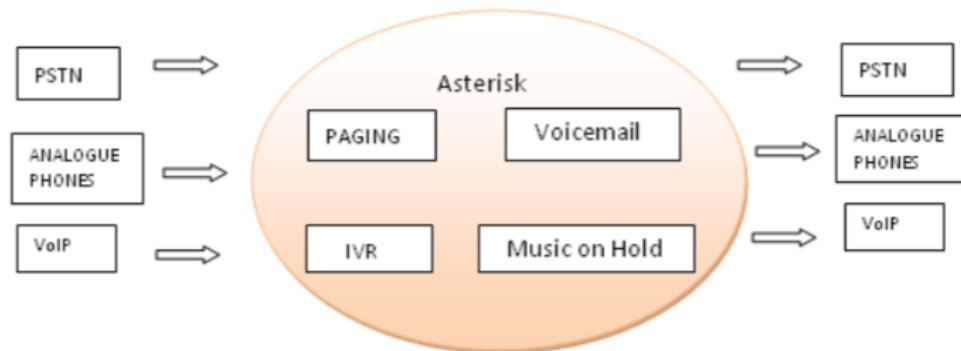


Figura 2. 4: Perspectiva general de un sistema basado en Asterisk.
Fuente: (Department of Business Administration, East West University, Bangladesh, 2014)

Además, los usuarios pueden obtener interfaces para el sistema operativo y los lenguajes de programación para una alta eficiencia, interfaces de administración opcionales basadas en la web, configuración en bases de datos SQL o archivos planos, registro detallado de llamadas en una base de datos y muchas más funciones. Por lo tanto, para resumirlo, se puede decir que el usuario de Asterisk puede: (Department of Business Administration, East West University, Bangladesh et al., 2014):

- Brindar servicio básico a teléfonos analógicos y digitales.
- Desarrollar una lógica de enrutamiento de llamadas para elegir una forma menos costosa de enrutar una llamada en particular.
- Enrutar llamadas de voz entrantes y salientes a través de líneas de voz estándar o de Internet (Bourreau & Doğan, 2014).
- Proporcionar servicios de correo de voz y teleconferencia.
- Desarrollar complejos o simples menús interactivos.
- Opera pequeñas o grandes colas para los call center.
- Anunciar los tiempos estimados de espera para los generadores de llamadas.

2.8.1. Arquitectura de Asterisk.

Asterisk está cuidadosamente diseñado para la flexibilidad. Los API (Application Programming Interface) específicos se definen para un máximo

alrededor de un sistema central de núcleo PBX. La interconexión interna anticipada de PBX de los protocolos específicos, códecs interfaces de hardware de la aplicación de telefonía, permiten a Asterisk utilizar cualquier hardware y tecnología adecuados disponibles ahora o en el futuro para realizar sus funciones esenciales, conectando aplicaciones de hardware (Department of Business Administration, East West University, Bangladesh et al., 2014).

2.8.1.1. Conmutación PBX.

La esencia de Asterisk es, por supuesto, un PBE que conecta llamadas entre tareas automatizadas. La conmutación conecta de manera transparente a las personas que llegan a diferentes interfaces de hardware y software (Qadeer & Imran, 2008).

2.8.1.2. Iniciador de aplicaciones.

Inicia aplicaciones que realizan servicios para los usuarios, como el correo de voz, la reproducción de archivos y la lista de directorios (Department of Business Administration, East West University, Bangladesh et al., 2014).

2.8.1.3. Traductor de códecs.

Utiliza módulos de códec para la codificación y decodificación de varios formatos utilizados en la industria de la telefonía. Hay una serie de códecs disponibles para reunir diversas necesidades y lograr el mejor equilibrio entre la calidad de audio y el ancho de banda (Thorne, 2001).

2.8.1.4. Módulo cargable de API's.

Se definen cuatro tipos de módulos cargables, que facilitan la abstracción del hardware y protocolo. Al utilizar este sistema de módulos cargables, el núcleo de Asterisk no tiene que preocuparse por los detalles de cómo se conecta una persona que llama, qué códigos se están utilizando, etc.

Utilizando estos API'S, Asterisk logra una abstracción absoluta entre sus funciones principales como un sistema de servidor PBX y las diversas

tecnologías existentes (o en desarrollo) en el ámbito de la telefonía. La aplicación API proporciona un uso flexible de los módulos de la aplicación para realizar cualquier función de manera flexible bajo demanda, y permite el desarrollo abierto de nuevas aplicaciones para satisfacer necesidades y situaciones únicas. Además, la carga de todas las aplicaciones como módulos permite un sistema flexible, lo que brinda a los administradores la capacidad de diseñar la ruta más adecuada para las personas que llaman en el sistema PBX y modificar las rutas de llamadas para adaptarse a las necesidades de comunicación cambiantes (Department of Business Administration, East West University, Bangladesh et al., 2014).

2.8.1.5. Canal API.

El canal API maneja el tipo de conexión a la que llega una persona que llama, ya sea una conexión VoIP, ISDN, PRI, señalización de bits Robbed o varias otras tecnologías. Los módulos dinámicos se cargan para contener los detalles de la capa inferior de estas conexiones (Thorne, 2001).

2.8.1.6. Aplicación API.

La aplicación de API permite que se ejecuten varios módulos de tareas realizando diversas funciones como conferencias, paginación y listado de directorios. El correo de voz, la transmisión de datos en línea y cualquier otra tarea que un sistema PBX pueda ejecutar ahora o en el futuro son manejados por estos módulos separados (Thorne, 2001).

2.8.1.7. Traductor Códecs API.

Carga módulos de códecs para admitir varios formatos de codificación y decodificación de audio, como GSM, Mu-Law, A-law e incluso MP3 (Department of Business Administration, East West University, Bangladesh et al., 2014).

2.8.1.8. Formato de archivo API.

Esto maneja la lectura y escritura de varios formatos de archivo para el almacenamiento de datos en el sistema de archivos (Bourreau & Doğan, 2014). Con el uso de estos APIs Asterisk logra una abstracción completa

entre sus funciones principales como un sistema de servidor PBX y las tecnologías divergentes existentes (o en desarrollo) en el ámbito de la telefonía. La forma modular es lo que permite a Asterisk incorporar sin problemas tanto el hardware de conmutación de telefonía implementado actualmente como las tecnologías de Packet Voice que están surgiendo en la actualidad (Imran et al., 2015). La capacidad de cargar módulos de códecs permite a Asterisk mantener el mismo aspecto compacto necesario para Packet Voice en conexiones lentas, como un módem telefónico, al tiempo que proporciona una alta calidad de audio en conexiones menos restringidas. La aplicación API proporciona un uso flexible de los módulos de la aplicación para realizar cualquier función de manera flexible bajo demanda, y permite el desarrollo abierto de nuevas aplicaciones para satisfacer necesidades y situaciones únicas. Además, la carga de todas las aplicaciones como módulos admite un sistema flexible, lo que permite al administrador diseñar la ruta más adecuada para las personas que llaman en el sistema PBX y modificar las rutas de llamadas para adaptarse al cambio (Department of Business Administration, East West University, Bangladesh et al., 2014).

2.8.2. Servicios que proporciona Asterisk.

Por lo general, VoIP utiliza dos tipos de protocolos: 1) Protocolos de señalización para establecer una conversación 2) Protocolos de transferencia de medios para la transferencia real de datos, una vez que se ha establecido la conexión. El Protocolo de inicio de sesión (SIP) es un protocolo de control (señalización) de nivel de aplicación para crear, modificar y finalizar sesiones con uno o más participantes. Estas sesiones incluyen llamadas telefónicas por Internet, distribución multimedia y multimedia. SIP tiene las siguientes características:

- Ligerero, ya que SIP tiene solo seis métodos, lo que reduce la complejidad.
- Independiente del transporte, ya que SIP se puede usar con UDP, TCP, ATM, etc.
- Basado en texto, permitiendo a los humanos leer mensajes SIP (Imran et al., 2015).

Los firewalls generalmente bloquean los tipos de paquetes de medios como UDP, aunque una forma de evitar esto es usar túneles y relés TCP para los medios con el fin de proporcionar NAT y el paso del firewall. Una solución consiste en hacer un túnel de los paquetes de medios dentro de los paquetes TCP o HTTP a un relé. Esta solución utiliza funcionalidad adicional junto con SIP, y empaqueta los paquetes de medios en una secuencia TCP que luego se envía al relé. El relé luego extrae los paquetes y los envía al otro punto final. Si el otro punto final está detrás de un NAT simétrico o firewall corporativo que no permite el tráfico VOIP, el relé transferiría los paquetes a otro túnel (Imran et al., 2015).

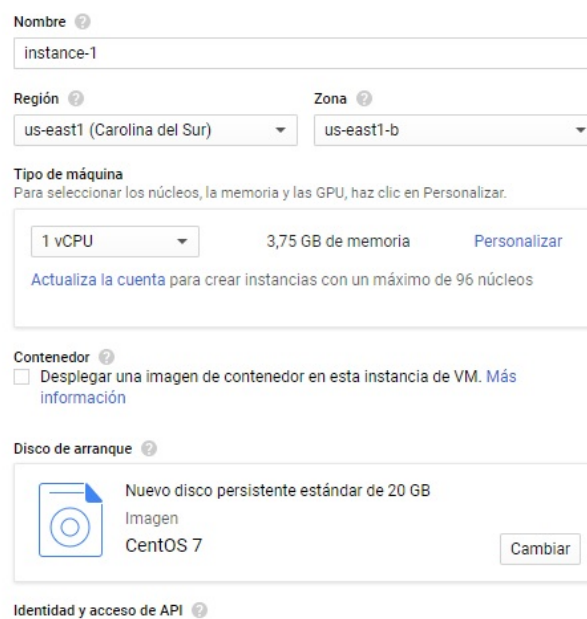
CAPÍTULO 3: SIMULACIÓN Y RESULTADOS OBTENIDOS

3.1. Asterisk.

Asterisk es una implementación de software libre y de código abierto de un teléfono Private Branch Exchange (PBX) creado originalmente por Mark Spencer de Digium. Como cualquier PBX, permite que varios teléfonos conectados se realicen llamadas y se conecten a otros servicios telefónicos. incluida la red telefónica pública conmutada (PSTN). Su nombre proviene del símbolo asterisco *, que en Unix (y sistemas operativos similares a Unix como Linux) y los entornos DOS representan un comodín, que coincide con cualquier secuencia de caracteres en un nombre de archivo. El software básico de Asterisk incluye muchas funciones disponibles en el propiedades de los sistemas PBX: (correo de voz, menú del teléfono de conferencia y distribución automática de llamadas) (Ghossoon, 2017).

3.2. Asignación de recursos.

El código de programación de la herramienta generadora de llamadas está alojado en una máquina virtual (VM) de Google Cloud con las siguientes características: sistema operativo, CentOS 7 x86_64; memoria ram, 4 GB; disco duro, 20 GB como se aprecia en la figura 3.1.



The image shows a configuration interface for a Google Cloud VM instance. The fields are as follows:

- Nombre:** instance-1
- Región:** us-east1 (Carolina del Sur)
- Zona:** us-east1-b
- Tipo de máquina:** 1 vCPU, 3,75 GB de memoria. A link for 'Personalizar' is visible.
- Contenedor:** A checkbox labeled 'Desplegar una imagen de contenedor en esta instancia de VM. Más información' is unchecked.
- Disco de arranque:** A new persistent standard disk of 20 GB with the image 'CentOS 7'. A 'Cambiar' button is present.
- Identidad y acceso de API:** This section is partially visible at the bottom.

Figura 3. 1: Asignación de recursos de máquina virtual de Google Cloud.

Elaborado por: Autor.

3.3. Configuraciones principales.

Una vez creada la VM y levantado el servicio se procede a realizar las configuraciones principales de la maquina tales como permisos de administrador (root) e instalación de los paquetes de desarrollo necesarios para el correcto funcionamiento del código, así como Asterisk y SIPPP los cuales serán detallados más adelante.

3.3.1. Permisos de administrador.

Mediante el comando `sudo passwd` se procede a asignar la contraseña de la máquina para el usuario administrador (root user) el cual tiene asignado todos los permisos para realizar modificaciones en general, tal como se muestra en la figura 3.2.

```
[lgrijalva_ucsg@instance-1 ~]$ sudo passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[lgrijalva_ucsg@instance-1 ~]$
```

Figura 3. 2: Permisos de administrador.

Elaborado por: Autor.

3.3.2. Paquetes de desarrollo.

En primer lugar, se habilita el repositorio `epel-release`, como se expone en la figura 3.3, el cual permite ingresar a una serie de paquetes de alta calidad para trabajar en Linux, éste se habilita mediante el comando `yum install epel-release` y posteriormente se instala un paquete de desarrollo (development tool) el cual contiene herramientas necesarias para la instalación del asterisk en la VM.

Otra utilidad necesaria es `wget`, habilitada mediante el comando `yum install wget`, esta línea de código permite limitar el ancho de banda utilizado para las descargas, realizar descargas de archivos, en segundo plano, reanudación de estas, etc.

```

https://ssh.cloud.google.com/projects/rock-dragon-227317/zones/us-east1-b/instances/instance-1?authuser=1&hl=es&projectNumbe...
[root@instance-1 lgrijalva ucsg]# yum install epel-release
Loaded plugins: fastestmirror
Determining fastest mirrors
epel/x86_64/metalink | 18 kB 00:00:00
* base: mirror.fileplanet.com
* epel: mirror.cogentco.com
* extras: mirror.hmc.edu
* updates: mirror.fileplanet.com
base | 3.6 kB 00:00:00
epel | 4.7 kB 00:00:00
extras | 3.4 kB 00:00:00
google-cloud-compute/signature | 454 B 00:00:00
google-cloud-compute/signature | 1.8 kB 00:00:00 !!!
google-cloud-sdk/signature | 454 B 00:00:00
google-cloud-sdk/signature | 1.4 kB 00:00:00 !!!
updates | 3.4 kB 00:00:00
(1/10): epel/x86_64/group_gz | 88 kB 00:00:00
(2/10): epel/x86_64/updateinfo | 955 kB 00:00:00
(3/10): base/7/x86_64/group_gz | 166 kB 00:00:00
(4/10): epel/x86_64/primary_db | 6.6 MB 00:00:00
(5/10): google-cloud-compute/updateinfo | 1.1 kB 00:00:00
(6/10): google-cloud-compute/primary | 3.6 kB 00:00:00
(7/10): google-cloud-sdk/primary | 71 kB 00:00:00
(8/10): updates/7/x86_64/primary_db | 1.3 MB 00:00:00
(9/10): extras/7/x86_64/primary_db | 156 kB 00:00:01
(10/10): base/7/x86_64/primary_db | 6.0 MB 00:00:01
google-cloud-compute | 10/10
google-cloud-sdk | 471/471
Package epel-release-7-11.noarch already installed and latest version
Nothing to do
[root@instance-1 lgrijalva ucsg]#

```

Figura 3. 3: Instalación de epel-release.
Elaborado por: Autor.

3.3.3. Instalación de Jansson.

Jansson es una librería de lenguaje c necesaria para la administración de Asterisk pues permite manipular data en formato JSON, este es un formato de lenguaje de programación en texto plano que usa la sintaxis de JavaScript, generalmente utilizado para transmitir datos en aplicaciones web. Con el uso del comando wget se descarga la librería: wget-Ojansson.tar.gz, la cual se encuentra en la página web: <https://github.com/akheron/jansson/archive/v2.12.tar.gz>, tal como la figura 3.4 lo muestra.

```

[root@instance-1 lgrijalva ucsg]# yum install wget
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirror.fileplanet.com
* epel: mirror.cogentco.com
* extras: mirror.hmc.edu
* updates: mirror.fileplanet.com
Resolving Dependencies
--> Running transaction check
--> Package wget.x86_64 0:1.14-18.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch             Version          Repository      Size
=====
Installing:
wget                   x86_64           1.14-18.el7     base            547 k
=====
Transaction Summary
-----
Install 1 Package

Total download size: 547 k
Installed size: 2.0 M
Is this ok [y/d/N]:

```

Figura 3. 4: Descarga de Jansson.
Elaborado por: Autor.

Posteriormente se mueve a una carpeta asignada mediante el comando `mv jansson.tar.gz /usr/local/src/` se descomprime el archivo mediante el comando `tar xvzf jansson.tar.gz` y se instala Jansson en la máquina virtual (figura 3.5).

```
jansson-2.12/test/suites/valid/utf-surrogate-four-byte-encoding/input
jansson-2.12/test/suites/valid/utf-surrogate-four-byte-encoding/output
[root@instance-1 src]# cd jansson-2.12/
[root@instance-1 jansson-2.12]# ./configure
bash: ./configure: No such file or directory
[root@instance-1 jansson-2.12]# makels -la
bash: makels: command not found
[root@instance-1 jansson-2.12]# ls -la
total 104
drwxrwxr-x. 8 root root 4096 Nov 26 08:10 .
drwxr-xr-x. 3 root root 48 Jan 19 18:33 ..
drwxrwxr-x. 2 root root 30 Nov 26 08:10 android
-rw-rw-r--. 1 root root 578 Nov 26 08:10 Android.mk
-rw-rw-r--. 1 root root 399 Nov 26 08:10 appveyor.yml
-rw-rw-r--. 1 root root 22005 Nov 26 08:10 CHANGES
-rw-rw-r--. 1 root root 2227 Nov 26 08:10 CleanSpec.mk
drwxrwxr-x. 2 root root 275 Nov 26 08:10 cmake
-rw-rw-r--. 1 root root 23254 Nov 26 08:10 CMakeLists.txt
-rw-rw-r--. 1 root root 5165 Nov 26 08:10 configure.ac
drwxrwxr-x. 3 root root 274 Nov 26 08:10 doc
drwxrwxr-x. 2 root root 46 Nov 26 08:10 examples
-rw-rw-r--. 1 root root 360 Nov 26 08:10 .gitignore
-rw-rw-r--. 1 root root 239 Nov 26 08:10 jansson.pc.in
-rw-rw-r--. 1 root root 1081 Nov 26 08:10 LICENSE
-rw-rw-r--. 1 root root 311 Nov 26 08:10 Makefile.am
-rw-rw-r--. 1 root root 1882 Nov 26 08:10 README.rst
-rwxrwxr-x. 1 root root 1640 Nov 26 08:10 release.sh
drwxrwxr-x. 2 root root 4096 Nov 26 08:10 src
drwxrwxr-x. 5 root root 101 Nov 26 08:10 test
-rw-rw-r--. 1 root root 1337 Nov 26 08:10 .travis.yml
[root@instance-1 jansson-2.12]# autoreconf -i
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, `.'.
libtoolize: copying file `./ltmain.sh'
libtoolize: Consider adding `AC_CONFIG_MACRO_DIR([m4])' to configure.ac and
libtoolize: rerunning libtoolize, to keep the correct libtool macros in-tree.
libtoolize: Consider adding `-I m4' to ACLOCAL_AMFLAGS in Makefile.am.
configure.ac:12: installing `./config.guess'
configure.ac:12: installing `./config.sub'
configure.ac:5: installing `./install-sh'
configure.ac:5: installing `./missing'
src/Makefile.am: installing `./depcomp'
parallel-tests: installing `./test-driver'
[root@instance-1 jansson-2.12]# █
```

Figura 3. 5: Instalación de Jansson.

Elaborado por: Autor.

3.3.4. Instalación de Asterisk.

Mediante el comando `wget` se descarga `asterisk wgt` en la página web: <http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-15-current.tar.gz>, se descomprime y se procede a correr los scripts de instalación, este proceso se presenta en la figura 3.6.

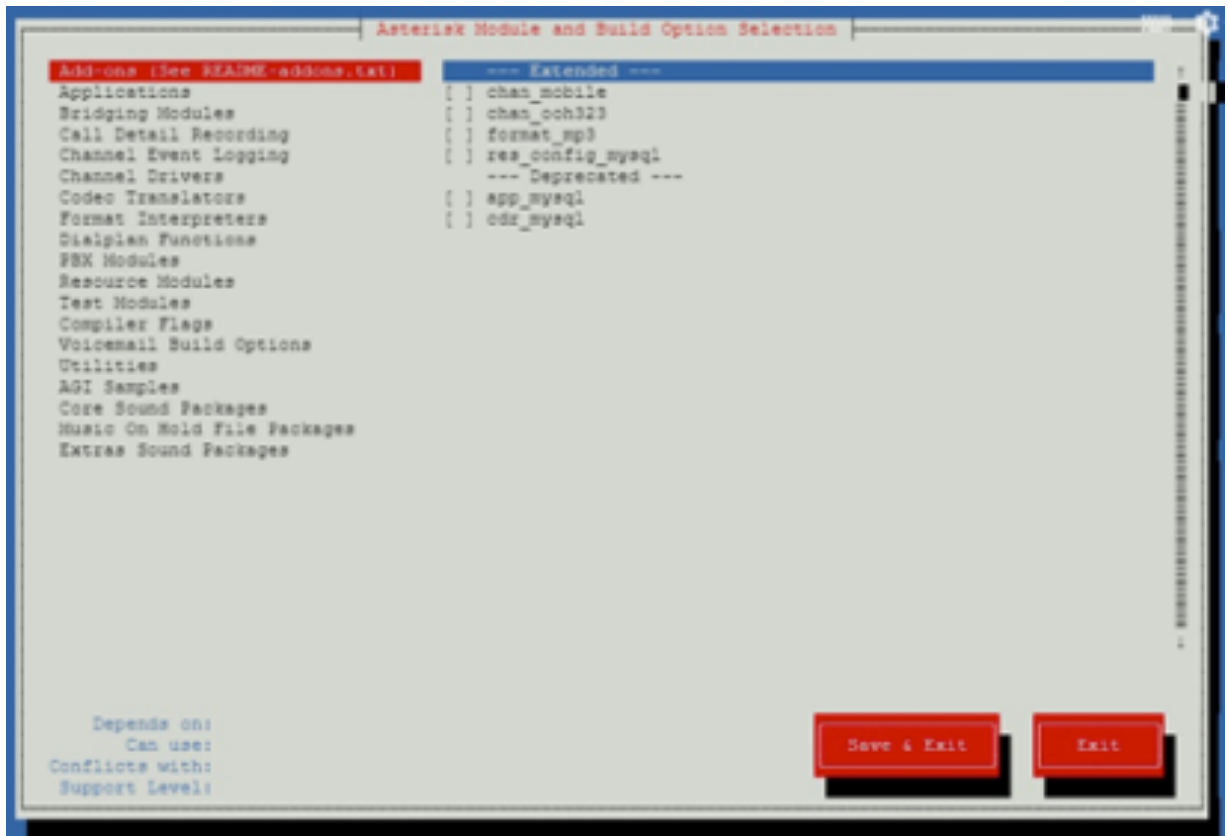


Figura 3. 6: Instación de Asterisk.
Elaborado por: Autor.

3.3.5. Instalación de dependencia adicionales para Asterisk.

Para la ejecución y una correcta compilación del software Asterisk es necesario instalar una serie de librerías adicionales para dar soporte a las herramientas utilizadas, estas librerías son instaladas mediante el comando `yum -y install libedit-devel sqlite-devel psmisc gmime-devel ncurses-devel libtermcap-devel sox newt-devel libxml2-devel libtiff-devel audiofile-devel gtk2-devel uuid-devel libtool libuuid-devel subversion kernel-devel kernel-devel-$(uname -r) git subversion kernel-devel crontabs cronie cronie-anacron`. Esta línea de comando se muestra en la figura 3.7.

```
[root@instance-1 asterisk-16.1.1]# yum -y install libedit-devel sqlite-devel psmisc gmime-devel ncurses-devel libt
rmcap-devel sox newt-devel libxml2-devel libtiff-devel audiofile-devel gtk2-devel uuid-devel libtool libuui
subversion kernel-devel kernel-devel-$(uname -r) git subversion kernel-devel crontabs cronie cronie-anacron wget vi
m
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.fileplanet.com
 * epel: mirror.cogentco.com
 * extras: mirror.hmc.edu
 * updates: mirror.fileplanet.com
Package libedit-devel-3.0-12.20121213cvs.el7.x86_64 already installed and latest version
Package sqlite-devel-3.7.17-8.el7.x86_64 already installed and latest version
Package psmisc-22.20-15.el7.x86_64 already installed and latest version
Package gmime-devel-2.6.23-1.el7.x86_64 already installed and latest version
Package ncurses-devel-5.9-14.20130511.el7_4.x86_64 already installed and latest version
Package ncurses-devel-5.9-14.20130511.el7_4.x86_64 already installed and latest version
Package sox-14.4.1-6.el7.x86_64 already installed and latest version
Package newt-devel-0.52.15-4.el7.x86_64 already installed and latest version
Package libxml2-devel-2.9.1-6.el7_2.3.x86_64 already installed and latest version
Package libtiff-devel-4.0.3-27.el7_3.x86_64 already installed and latest version
Package 1:audiofile-devel-0.3.6-4.el7.x86_64 already installed and latest version
Package gtk2-devel-2.24.31-1.el7.x86_64 already installed and latest version
Package uuid-devel-1.6.2-26.el7.x86_64 already installed and latest version
Package libtool-2.4.2-22.el7_3.x86_64 already installed and latest version
Package libuuid-devel-2.23.2-59.el7.x86_64 already installed and latest version
Package subversion-1.7.14-14.el7.x86_64 already installed and latest version
Package kernel-devel-3.10.0-957.1.3.el7.x86_64 already installed and latest version
Package kernel-devel-3.10.0-957.1.3.el7.x86_64 already installed and latest version
Package git-1.8.3.1-20.el7.x86_64 already installed and latest version
Package subversion-1.7.14-14.el7.x86_64 already installed and latest version
Package kernel-devel-3.10.0-957.1.3.el7.x86_64 already installed and latest version
Package crontabs-1.11-6.20121102git.el7.noarch already installed and latest version
Package cronie-1.4.11-20.el7_6.x86_64 already installed and latest version
Package cronie-anacron-1.4.11-20.el7_6.x86_64 already installed and latest version
Package wget-1.14-18.el7.x86_64 already installed and latest version
Package 2:vim-enhanced-7.4.160-5.el7.x86_64 already installed and latest version
Nothing to do
[root@instance-1 asterisk-16.1.1]#
```

Figura 3. 7: Instalación de dependencias adicionales.

Elaborado por: Autor.

3.3.6. Habilitación de servicio Asterisk.

Para inicializar Asterisk es necesario aplicar la línea de comando `systemctl start asterisk`, la cual habilita el servicio, este proceso se presenta en la figura 3.8.

```
https://ssh.cloud.google.com/projects/rock-dragon-227317/zones/us-east1-b/instances/instance-1?authuser=1&hl=es&projectNumber...
[root@instance-1 asterisk-16.1.1]# systemctl start asterisk
[root@instance-1 asterisk-16.1.1]#
```

Figura 3. 8: Inicialización de Asterisk.

Elaborado por: Autor.

Sin embargo, es necesario revisar el estado del servicio mediante el comando `systemctl status asterisk`, debido a que puede estar inicializado, pero generando cierto error, la inspección del servicio se indica en la figura 3.9.

```
[root@instance-1 asterisk-16.1.1]# systemctl status asterisk
● asterisk.service - LSB: Asterisk PBX
   Loaded: loaded (/etc/rc.d/init.d/asterisk; bad; vendor preset: disabled)
   Active: active (running) since Sat 2019-01-19 23:47:02 UTC; 50min ago
     Docs: man:systemd-sysv-generator(8)
   Main PID: 28384 (asterisk)
   CGroup: /system.slice/asterisk.service
           └─28382 /bin/sh /usr/sbin/safe_asterisk
             └─28384 /usr/sbin/asterisk -f -vvvg -c

Jan 19 23:47:02 instance-1 systemd[1]: Stopped LSB: Asterisk PBX.
Jan 19 23:47:02 instance-1 systemd[1]: Starting LSB: Asterisk PBX...
Jan 19 23:47:02 instance-1 asterisk[28355]: Starting asterisk:
Jan 19 23:47:02 instance-1 systemd[1]: PID file /var/run/asterisk/asterisk.pid not readable (yet?) after start.
Jan 19 23:47:02 instance-1 systemd[1]: asterisk.service: Supervising process 28384 which is not our child. ...xits.
Jan 19 23:47:02 instance-1 systemd[1]: Started LSB: Asterisk PBX.
Jan 20 00:30:43 instance-1 systemd[1]: asterisk.service: Supervising process 28384 which is not our child. ...xits.
Jan 20 00:34:39 instance-1 systemd[1]: asterisk.service: Supervising process 28384 which is not our child. ...xits.
Hint: Some lines were ellipsized, use -l to show in full.
[root@instance-1 asterisk-16.1.1]#
```

Figura 3. 9: Revisión del estado de Asterisk.

Elaborado por: Autor.

3.3.7. Deshabilitación de firewall.

Asterisk por defecto tiene habilitado un firewall el cual se debe deshabilitar pues bloquea la consola de este, esta deshabilitación debe de realizarse en dos instancias, la primera en el sistema y la segunda en el archivo de configuración. La deshabilitación en el sistema se la realiza mediante la línea de comandos `setenforce 0` (figura 3.10).

```
https://ssh.cloud.google.com/projects/rock-dragon-227317/zones/us-east1-b/instances/instance-1?authuser=1&hl=es&projectNumber...
[root@instance-1 asterisk-16.1.1]# setenforce 0
[root@instance-1 asterisk-16.1.1]#
```

Figura 3. 10: Deshabilitación en el sistema.

Elaborado por: Autor.

La deshabilitación en el archivo de configuración se realiza mediante la línea de comando `vi/etc/selinux/config` (figura 3.11).

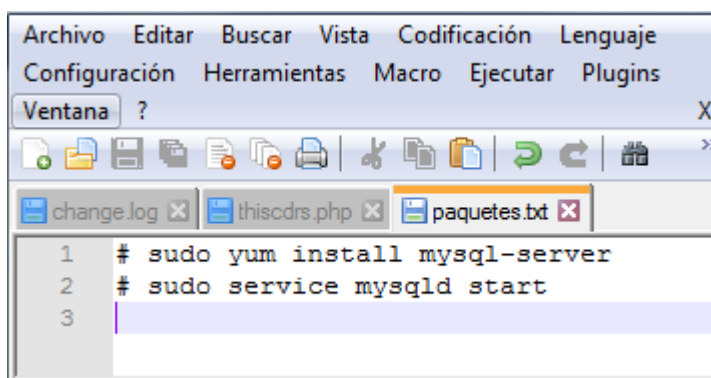
```
https://ssh.cloud.google.com/projects/rock-dragon-227317/zones/us-east1-b/instances/instance-1?authuser=1&hl=es&projectNumber...
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
~
~
```

Figura 3. 11: Deshabilitación en el archivo de configuración.

Elaborado por: Autor.

3.3.8. Generación de CDRs en Asterisk.

Para generar CDRs (Call Detail Record/Registro de Detalles de Llamadas) es necesario crear la posibilidad que Asterisk se conecte a una base de datos de MySQL a través de ODBC (Open DataBase Connectivity). Para configurar MySQL se instala el paquete del servidor MySQL, se inicia el servicio DB, se aplica seguridad a la instalación y por último se configura un usuario y base de datos para Asterisk en dentro MySQL. Todo esto se lo realiza mediante las líneas de comando, tal como se muestra en la figura 3.12.

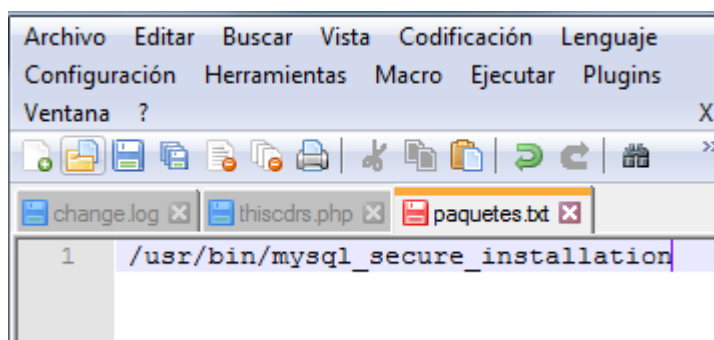


```
1 # sudo yum install mysql-server
2 # sudo service mysqld start
3
```

Figura 3. 12: Generación de CDRS en Asterisk.

Elaborado por: Autor

Al aplicar seguridad a la instalación como previamente se menciona, se ejecuta el siguiente script (figura 3.13).

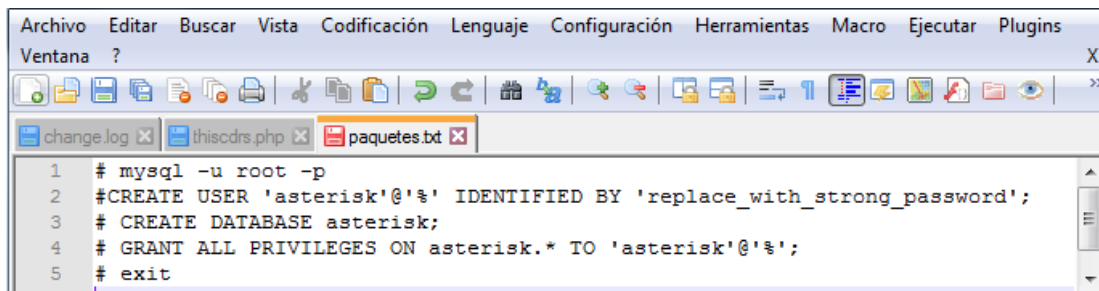


```
1 /usr/bin/mysql_secure_installation
```

Figura 3. 13: Instalación segura MySQL.

Elaborado por: Autor

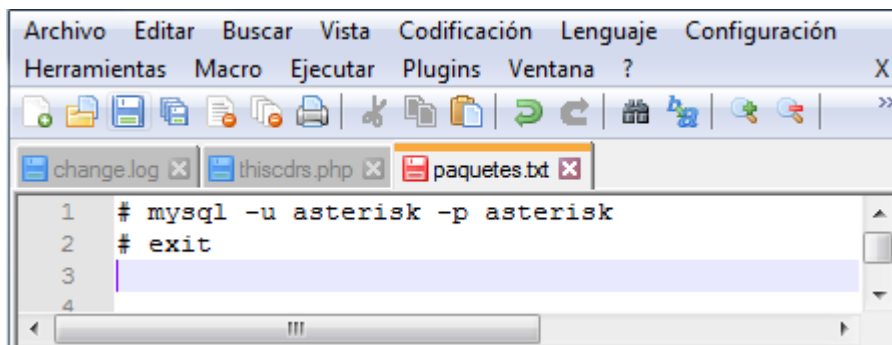
Configurar usuario y base de datos para Asterisk en MySQL se realiza mediante las siguientes líneas de comando (figura 3.14).



```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins
Ventana  ?
change.log x thiscds.php x paquetes.txt x
1 # mysql -u root -p
2 #CREATE USER 'asterisk'@'%' IDENTIFIED BY 'replace_with_strong_password';
3 # CREATE DATABASE asterisk;
4 # GRANT ALL PRIVILEGES ON asterisk.* TO 'asterisk'@'%' ;
5 # exit
```

Figura 3. 14: Configuración de usuario y base de datos.
Elaborado por: Autor.

Al comprobar los permisos se puede iniciar sesión en la base de datos de Asterisk y luego cerrar sesión nuevamente mediante los comandos presentados en la figura 3.15.

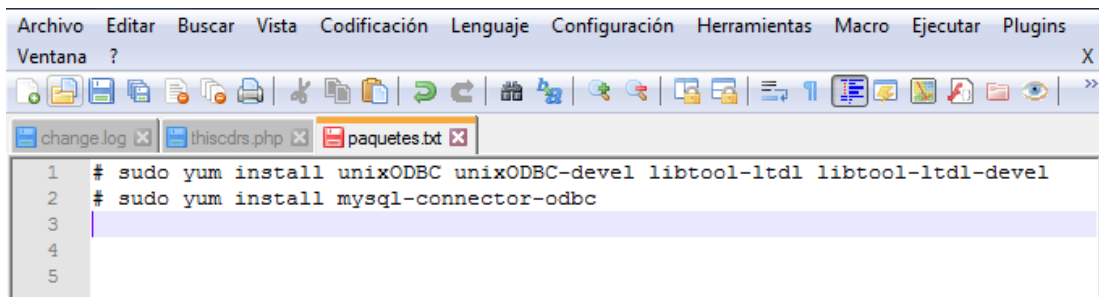


```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración
Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
change.log x thiscds.php x paquetes.txt x
1 # mysql -u asterisk -p asterisk
2 # exit
3
4
```

Figura 3. 15: Comprobación de permisos para el inicio de sesión.
Elaborado por: Autor.

3.3.9. Instalación de ODBC.

Se instala los paquetes más recientes de UnixODBC y GNU Libtool Dynamic Module Loader, estos paquetes de desarrollo también son necesarios, pues más adelante, Asterisk debe usarlos al crear módulos relacionados con ODBC, esto se realiza mediante las líneas de comando indicadas en la figura 3.16.

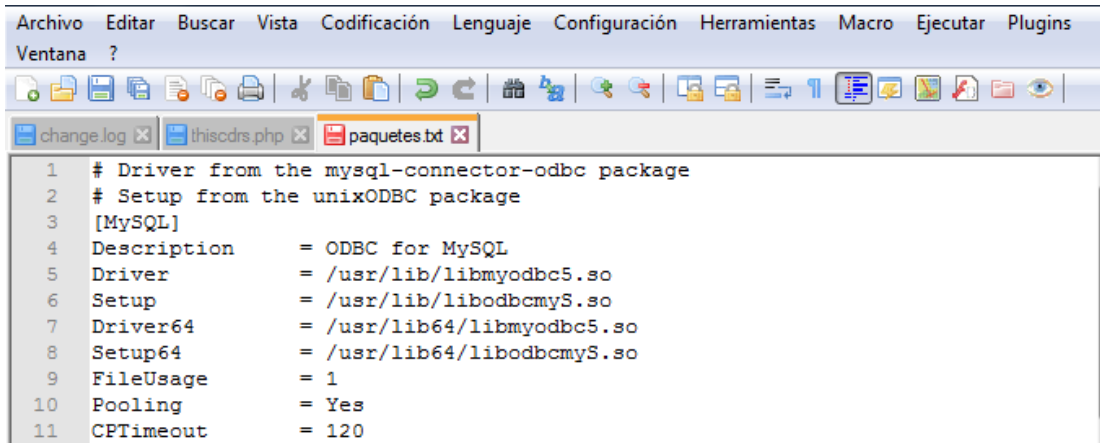


```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins
Ventana  ?
change.log x thiscds.php x paquetes.txt x
1 # sudo yum install unixODBC unixODBC-devel libtool-ltdl libtool-ltdl-devel
2 # sudo yum install mysql-connector-odbc
3
4
5
```

Figura 3. 16: Instalación de ODBC.
Elaborado por: Autor.

3.3.10. Configuración de ODBC y MySQL.

Con la reciente versión de UnixODBC, su configuración se halla en el archivo `/etc/odbcinst.ini`, es de gran importancia verificar que posea la siguiente configuración (figura 3.17):

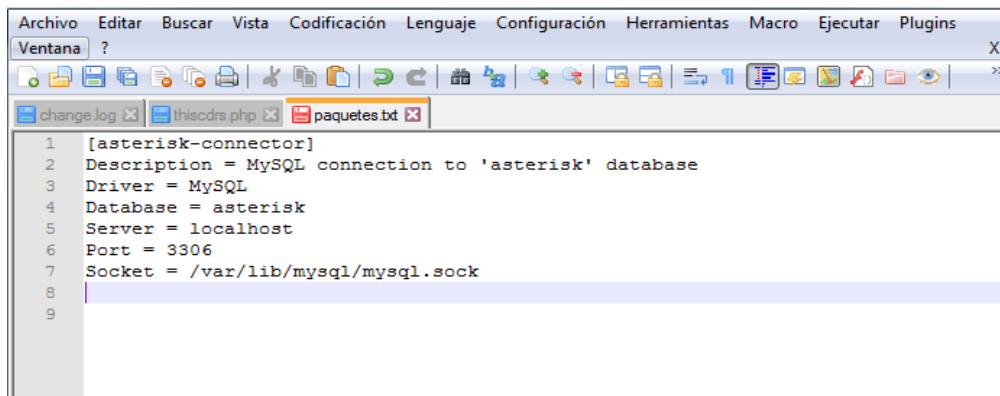


```
1 # Driver from the mysql-connector-odbc package
2 # Setup from the unixODBC package
3 [MySQL]
4 Description      = ODBC for MySQL
5 Driver           = /usr/lib/libmyodbc5.so
6 Setup           = /usr/lib/libodbcmyS.so
7 Driver64        = /usr/lib64/libmyodbc5.so
8 Setup64         = /usr/lib64/libodbcmyS.so
9 FileUsage       = 1
10 Pooling         = Yes
11 CPTimeout      = 120
```

Figura 3. 17: Configuración de ODBC y de MySQL.

Elaborado por: Autor.

La configuración de MySQL se realiza mediante el archivo `/etc/odbc.ini` para crear un DSN (Sistemas de Nombre de Dominio) para Asterisk. El archivo puede estar vacío, por esta razón se escribe las siguientes líneas de comandos que se muestran en la figura 3.18.



```
1 [asterisk-connector]
2 Description = MySQL connection to 'asterisk' database
3 Driver = MySQL
4 Database = asterisk
5 Server = localhost
6 Port = 3306
7 Socket = /var/lib/mysql/mysql.sock
8
9
```

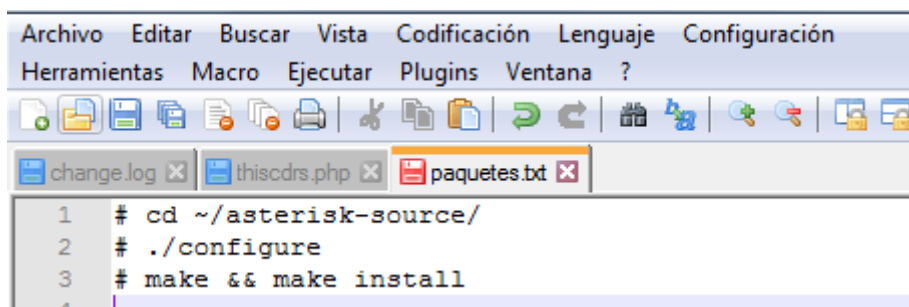
Figura 3. 18: Configuración MySQL.

Elaborado por: Autor.

3.3.11. Configuración de Asterisk para usar ODBC y MySQL.

Con la base de datos MySQL, ODBC y un conector MySQL ODBC instalado y básicamente configurado, el siguiente paso es recompilar Asterisk para lograr construir los módulos ODBC. Una vez que estos módulos existen, se configura los archivos adecuados en Asterisk,

dependiendo de la información que desee escribir o leer desde MySQL, para esto se emplea las líneas de comando indicadas en la figura 3.19.



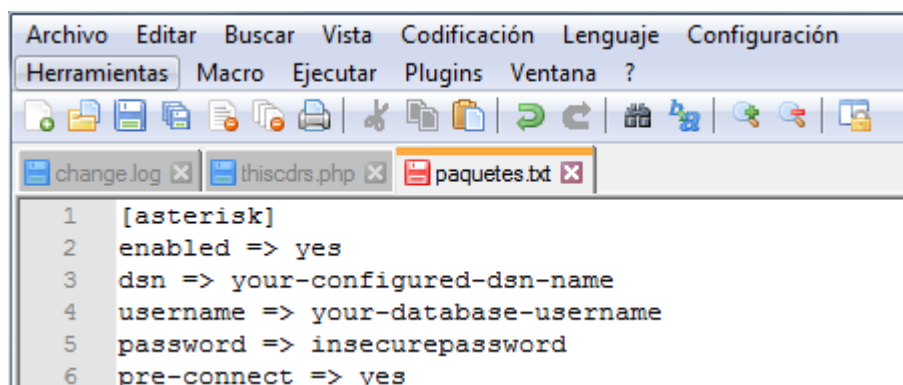
```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración
Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
change.log x thiscdrs.php x paquetes.txt x
1 # cd ~/asterisk-source/
2 # ./configure
3 # make && make install
4
```

Figura 3. 19: Configuración de Asterisk para usar ODBC y MySQL.

Elaborado por: Autor

3.3.12. Configuración Res_odbc.

El módulo **res_odbc** para Asterisk proporciona conectividad a varios backends de base de datos a través de ODBC. Las características de Asterisk como Asterisk Realtime Architecture, Registros de Detalles de Llamadas, Registro de Eventos de Canal, pueden conectarse a una base de datos a través de res_odbc, esta configuración se realiza mediante las líneas de comando a continuación en la figura 3.20.



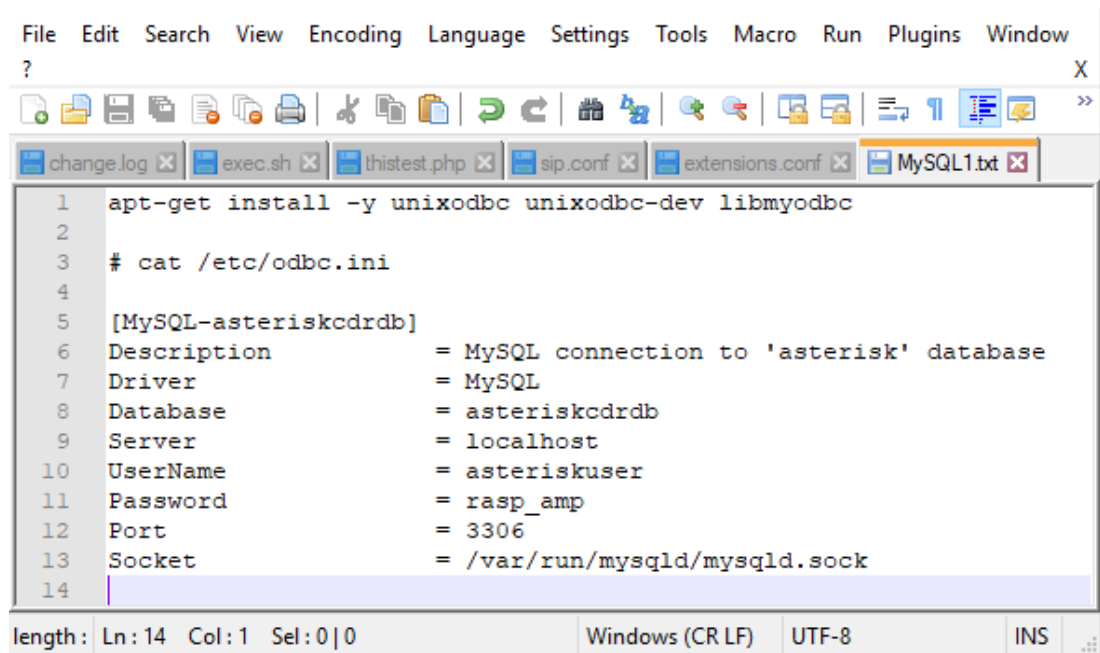
```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración
Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
change.log x thiscdrs.php x paquetes.txt x
1 [asterisk]
2 enabled => yes
3 dsn => your-configured-dsn-name
4 username => your-database-username
5 password => insecurepassword
6 pre-connect => yes
```

Figura 3. 20: Configuración Res_odbc.

Elaborado por: Autor

3.3.13. Creación de la tabla de base de datos.

Para configurar el odbc para que Asterisk lo utilice como conector a la tabla cdr de la base de datos MySQL, se instala en un BeagleBone Black (Plataforma de desarrollo de código abierto compatible con la comunidad de desarrolladores y aficionados del procesador ARM). La instalación es basada en Ubuntu, este proceso se muestra en la figura 3.21.



```
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window
?
change.log x exec.sh x thistest.php x sip.conf x extensions.conf x MySQL1.txt x
1 apt-get install -y unixodbc unixodbc-dev libmyodbc
2
3 # cat /etc/odbc.ini
4
5 [MySQL-asteriskcdrdb]
6 Description      = MySQL connection to 'asterisk' database
7 Driver           = MySQL
8 Database         = asteriskcdrdb
9 Server           = localhost
10 UserName        = asteriskuser
11 Password        = rasp_amp
12 Port            = 3306
13 Socket          = /var/run/mysqld/mysqld.sock
14
length: Ln: 14 Col: 1 Sel: 0|0 Windows (CR LF) UTF-8 INS
```

Figura 3. 21: Creación de la tabla de base de datos.
Elaborado por: Autor.

3.4. Interfaz de usuario.

Para la interfaz de usuario se crea dos páginas web alojadas en el servidor con dirección IP 35.211.100.136, estas páginas se desarrollaron con PHP y HTML, con la finalidad de brindar al usuario final una solución compacta y de fácil manejo, en esta sección el usuario envía una llamada de prueba al proveedor internacional de telefonía IP y analiza la calidad mediante la captura del audio.

En cuanto al nivel de calidad se realizan las pruebas mediante el uso del software SIP HOMER, un sistema de monitoreo de funcionamiento de la red de un proveedor de telefonía IP, mediante la cuantificación del MOS (Mean Opinion Score) que determina en una escala del 1 al 5 el nivel de calidad proporcionado, siendo el valor 1 el nivel más bajo y 5 el máximo.

3.5. Interfaz de usuario envío de la llamada.

En esta sección el usuario ingresa el número que desea sea conectado por el proveedor (figura 3.22).

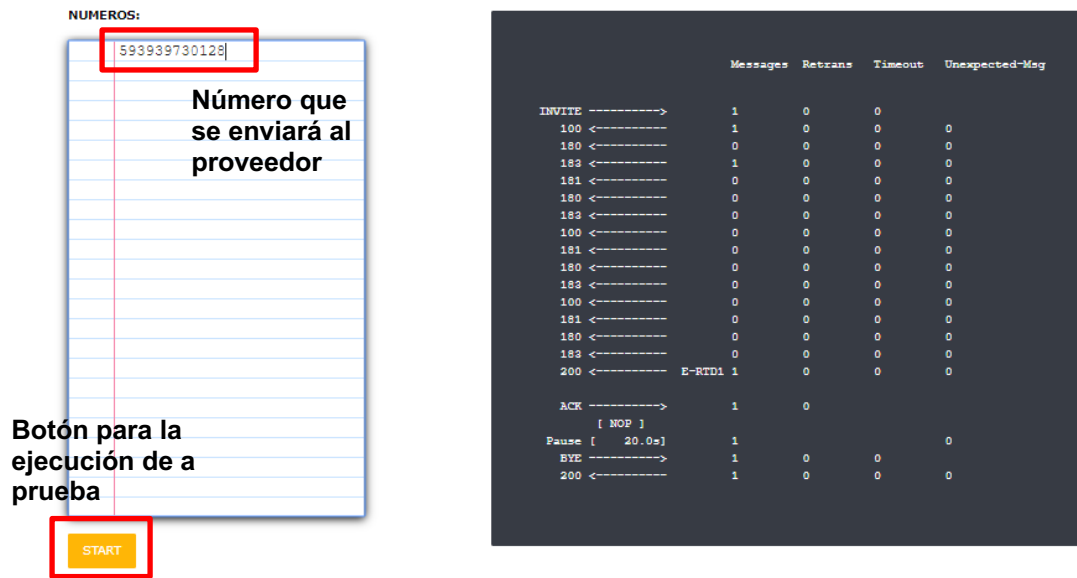


Figura 3. 22: Interfaz de usuario envío de la llamada.

Elaborado por: Autor.

En primera instancia se cargan los archivos para el estilo de los controles y estilo para el cuerpo de la página, asignando 5 puntos en todos los márgenes de la página, así como el color del fondo y negrilla para el cuerpo de la pagina, como lo indica la figura 3.23.

```

Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
change.log  iniscdrs.php  paquetes.bd  thistest.php
1  <!DOCTYPE html>
2  <html xmlns="http://www.w3.org/1999/xhtml" lang="en">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5
6  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
7  <link href="telerik/styles/kendo.common-material.min.css" rel="stylesheet" type="text/css" />
8  <link href="telerik/styles/kendo.material.min.css" rel="stylesheet" type="text/css" />
9  <link href="telerik/styles/kendo.material.mobile.min.css" rel="stylesheet" type="text/css" />
10 <script src="telerik/js/jquery.min.js"></script>
11 <script src="telerik/js/kendo.web.min.js"></script>
12
13
14 <title>Prueba Calidad de Audio</title>
15
16 <style type="text/css">
17 body
18 {
19     margin-bottom: 5px;
20     margin-left: 5px;
21     margin-right: 5px;
22     margin-top: 5px;
23     font-family: verdana,arial,sans-serif;
24     font-size: 12px;
25     color: #000000 ;
26     background-color: #FFFFFF;
27 }
28
29
30 .bold
31 {
32     font-weight: bold;
33 }
  
```

Figura 3. 23: Configuración de estilos, fondo y margen de la página.

Elaborado por: Autor.

Adicionalmente se configura el estilo de la tabla en donde se ingresa el número a ser enviado al proveedor y el estilo para los botones, en este caso el botón start (figura 3.24).

```

44
45 .curved-table-color
46 {
47     font-family: verdana,arial,sans-serif;
48     border-collapse:separate;
49     border: solid rgba(255, 165, 0, 0.1)::;
50     border-radius: 5px;
51     font-size:25px;
52 }
53
54 input[type=submit]
55 {
56     border-radius: 5px;
57     border: 0;
58     width: 90px;
59     height:20px;
60     font-family: verdana,arial,sans-serif;
61     font-size:12px;
62     background: #59B329;
63     background: -moz-linear-gradient(top, #59B329 1%, #59B329 100%);
64     background: -webkit-gradient(linear, left top, left bottom, color-stop(1%, #59B329), color-stop(100%, #59B329));
65     background: -webkit-linear-gradient(top, #59B329 1%, #59B329 100%);
66     background: -o-linear-gradient(top, #59B329 1%, #59B329 100%);
67     background: -ms-linear-gradient(top, #59B329 1%, #59B329 100%);
68     background: linear-gradient(to bottom, #59B329 1%, #59B329 100%);
69     filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#59B329', endColorstr='#59B329', GradientType=0);
70 }

```

Figura 3. 24: Configuración de tabla y botones.
Elaborado por: Autor.

Se configura el estilo para la caja de texto y la tabulación para el grupo number, grupo bottom y su tabulación, de la misma manera se realiza la configuración para el área de notificaciones y el estilo para los otros controladores (figura 3.25).

```

74 input[type="text"]
75 {
76     padding: 3px;
77     font-family: "Courier New", Courier, monospace;
78     font-size: 12px;
79     color: #000000;
80     border: 1px solid #a9c6e9;
81     background: #66CCFF;
82     background: -moz-linear-gradient(top, #c2c2c2 1%, #c2c2c2 100%);
83     background: -webkit-gradient(linear, left top, left bottom, color-stop(1%, #c2c2c2), color-stop(100%, #c2c2c2));
84     background: -webkit-linear-gradient(top, #c2c2c2 1%, #c2c2c2 100%);
85     background: -o-linear-gradient(top, #c2c2c2 1%, #c2c2c2 100%);
86     background: -ms-linear-gradient(top, #c2c2c2 1%, #c2c2c2 100%);
87     background: linear-gradient(to bottom, #c2c2c2 1%, #c2c2c2 100%);
88     filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#c2c2c2', endColorstr='#c2c2c2');
89     -moz-border-radius: 3px;
90     border-radius: 3px;
91     border-radius: 3px;
92 }
93
94 p.numbers
95 {
96     position: absolute;
97     top: 10%;
98     left: 35%;
99 }
100
101 p.buttonpos
102 {
103     position: absolute;
104     top: 100px;
105     left: 80px;
106 }
107

```

Figura 3. 25: Configuración de estilo de caja de texto, tabulación y botones.
Elaborado por: Autor.

Se codifica también una línea de comandos para limpiar el texto en la caja de texto de ingreso de números y otra línea de comandos para la creación de texto a ser escrito en el archivo csv utilizado por SIPP para ejecutar las pruebas, esta línea de codificación se muestra en la figura 3.26.

```

192
193 <body bgcolor="#E0FFFF">
194
195 <?php
196     require_once 'telerik/Xendo/Autoload.php';
197     $showinfotestsummary="Listos para iniciar la prueba.....!";
198     if (isset($_POST['test'])) {
199
200         $cleaningtextarea = preg_replace("/[^0-9,\n]/", "", $_POST['numberlist']);
201         if (!empty($cleaningtextarea)) {
202             $numberstobetested = $cleaningtextarea;
203         }
204
205         if (!empty($numberstobetested)) {
206             $numberscount=0;
207             $numberscountdiscarded=0;
208             $stringtofile = "SEQUENTIAL" . "\n";
209             $numberstobetestedarray = explode("\n", trim(ltrim($numberstobetested)));
210             foreach ($numberstobetestedarray as $numberstobetesteditem) {
211                 $cleaningnumberinline = preg_replace("/[^0-9]/", "", $numberstobetesteditem);
212                 if (!empty($cleaningnumberinline)) {
213                     if ($numberscount < 10) {
214                         $stringtofile = $stringtofile . "0999123123;$cleaningnumberinline;\n";
215                         $numberscount = $numberscount + 1;
216                     } else {
217                         $numberscountdiscarded = $numberscountdiscarded + 1;
218                     }
219                 }
220             }
221         }

```

Figura 3. 26: Limpieza de caracteres en la caja de texto y de ingreso de números.
Elaborado por: Autor.

Las siguientes líneas de comando contienen el código a ejecutarse en el servidor para generar la llamada de prueba y las variables para la conexión al servidor vía SSH, así como la conexión al servidor vía SSH y la ejecución de comando en el servidor (figura 3.27).

```

229
230 //*****
231 $serverip = "10.142.0.2";
232 $scriptex = "sudo sh /var/www/html/script/exec.sh $numberscount $filecsv0 $serverip";
233
234
235 //*****
236
237 $runhostname = 'localhost';
238 $runusername = 'root';
239 $runpassword = 'Tesis01';
240
241
242
243 $runconnect = ssh2_connect($runhostname, 22);
244 ssh2_auth_password($runconnect, $runusername, $runpassword);
245
246
247 //*****
248
249 $stream1 = ssh2_exec($runconnect, $scriptex);
250
251

```

Figura 3. 27: Código generador de llamadas y variables de conexión.
Elaborado por: Autor

Se agrega un bloqueo hasta que se complete la ejecución total del comando y una línea de comandos para generar una alerta indicando la cantidad de números procesados y descartados con un máximo de diez números por ejecución, este proceso está indicado en la figura 3.28.

```

252
253
254
255
256
257
258
259
260
261
262
263
264
265
stream_set_blocking($stream1, true);
$thisoutput = stream_get_contents($stream1);

// .....
if ($numberscountdiscarded > 0) {
    echo "<script type='text/javascript'>alert('$numberscountdiscarded numeros descartados, $numberscount seran procesados.....!')</script>";
} else {
    echo "<script type='text/javascript'>alert('$numberscount seran procesados.....!')</script>";
}
// .....

```

Figura 3. 28: Generador de bloque y alerta.
Elaborado por: Autor.

Finalmente se realiza la creación de controladores en la página (figura 3.29).

```

334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
</form>
<script>
$(document).ready(function() {
    function onTest(e) {
        document.getElementById('notification').value='Prueba en progreso.....!';
    }
    $("#testbutton").kendoButton({
        click: onTest
    });
});
</script>
/* Creacion de controladores en pagina */
</body>
</html>

```

Figura 3. 29: Controladores de página.
Elaborado por: Autor.

3.5.1. Interfaz de resultados para el usuario.

En esta sección el usuario obtiene los resultados de la prueba y varios datos relevantes como: fecha de ejecución, duración, estado e ID de la llamada, número de origen, número de destino y el audio (figura 3.30).

0:00 / 0:00

Inicio	Duracion	Origen	Destino	Estado	Audio	Play	Call-ID
2019-01-30 22:01:30	20.0022	0999123123	0999123123	ANSWERED	WAV	🔊	61394aae5b89b2cc1eff6bcc78b0e0a1
2019-01-30 22:36:04	20.0041	0999123123	0999123123	ANSWERED	WAV	🔊	3a3056cd3582f4892119dddf1df4b82a
2019-01-31 02:01:31	20.0049	0999123123	0999123123	ANSWERED	WAV	🔊	1348608e708a933c1fa130e42e55b8a5
2019-02-01 01:52:59	45.2757	0999123123	0999123123	ANSWERED	WAV	🔊	2fa5aa9672a65fed6c6598e25fe05aa5
2019-02-01 03:52:18	20.009	0999123123	333333333	ANSWERED	WAV	🔊	0464f9967a46100d075668982ddb11f9
2019-02-01 03:55:07	20.006	0999123123	3333333	ANSWERED	WAV	🔊	7db66b6d2ed74efa043e30e375b28b29
2019-02-01 03:58:12	20.0082	0999123123	4444444444	ANSWERED	WAV	🔊	0c84c5557cac996722b174fd00da56ad
2019-02-01 04:19:44	20.0115	0999123123	55555	ANSWERED	WAV	🔊	6ccf8d553787549a0db8eee60979857b
2019-02-01 04:21:32	20.004	0999123123	121212	ANSWERED	WAV	🔊	13d8daa0299599c906bccd522cf62b19
2019-02-03 01:30:51	20.0027	0999123123	12121212	ANSWERED	WAV	🔊	2607e7482ea2f6e878c7201a0a80b9e6
2019-02-03 01:33:27	20.0037	0999123123	14141414	ANSWERED	WAV	🔊	357f0689577cea6116a007c76fd660e6
2019-02-03 01:42:30	20.0033	0999123123	15151515	ANSWERED	WAV	🔊	72353f5c71cf259b7f0fd0272f05a11b
2019-02-03 01:46:48	20.0025	0999123123	24242424	ANSWERED	WAV	🔊	257e53f1042cef445f9d84807e2c76e6
2019-02-03 01:48:02	20.004	0999123123	26262626	ANSWERED	WAV	🔊	6ff927b1286b13934582e7ad2949219e
2019-02-03 01:48:38	20.0029	0999123123	4545454545	ANSWERED	WAV	🔊	42488be277206d7320c46e4e31ca1c09
2019-02-03 22:21:25	20.0034	0999123123	121414141	ANSWERED	WAV	🔊	1e84ff73044375c148c7a05f48f8c87a
2019-02-03 22:36:23	20.0039	0999123123	24242424	ANSWERED	WAV	🔊	4d8a9ee239504d79420d124c35e6fd0
2019-02-03 22:52:29	20.0028	0999123123	55555555	ANSWERED	WAV	🔊	779f5d7b67ab63ed0d577da72c1446f3
2019-02-06 01:29:43	20.0027	0999123123	0999123123	ANSWERED	WAV	🔊	2327fb154e6c20c679cf35c553a1e95e
2019-02-06 01:34:06	60.0047	0999123123	0999123123	ANSWERED	WAV	🔊	46fa517c276e85c037facc61407e931f
2019-02-06 01:48:01	60.004	0999123123	1212121212	ANSWERED	WAV	🔊	237bd59034fadfb311fc3066c4cd92
2019-02-06 01:59:48	60.003	0999123123	0999123123	ANSWERED	WAV	🔊	183c29967c4493656f6ce130417c0be9
2019-02-06 02:04:14	60.003	0999123123	0999123123	ANSWERED	WAV	🔊	298850c843e7f59f749ff3d63b703f32
2019-02-06 02:08:23	60.0044	0999123123	12121212	ANSWERED	WAV	🔊	148591e524476502586e6f3573b92fd3

Figura 3. 30: Interfaz de usuario.

Elaborado por: Autor

Se crea una línea de comandos para generar colores alternados entre cada resultado mediante la función `altRows` y así brindar un resultado más claro al usuario, tal como lo muestra la figura 3.31.

```

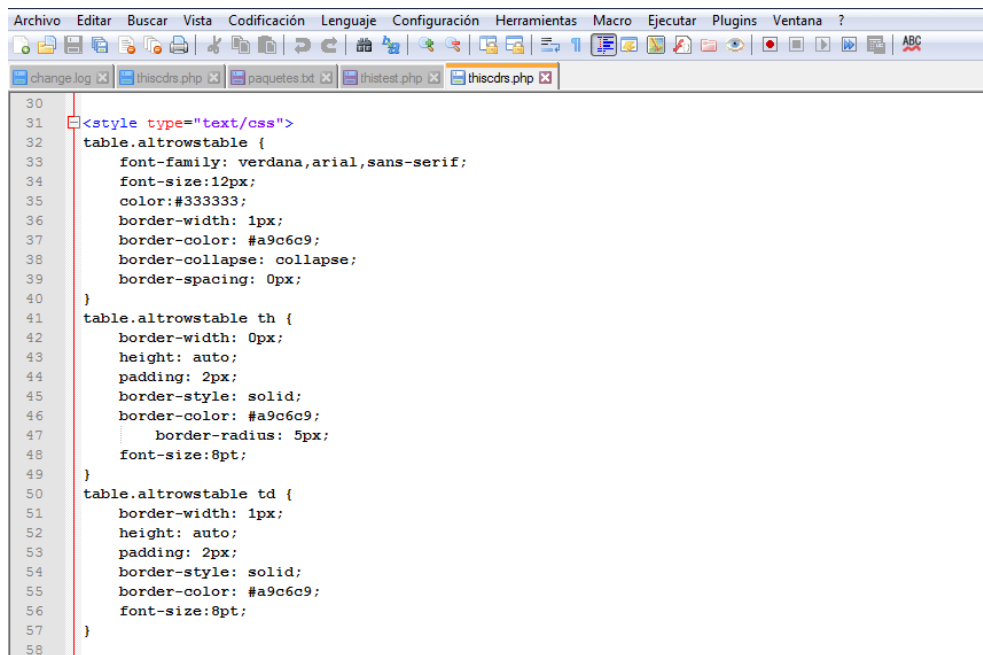
1 <DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml" lang="en">
3 <head>
4
5 <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
7
8 <title>CDRs</title>
9
10 <script type="text/javascript">
11 function altRows(id) {
12     if(document.getElementsByTagName) {
13         var table = document.getElementById(id);
14         var rows = table.getElementsByTagName("tr");
15         for(i = 0; i < rows.length; i++){
16             if(i % 2 == 0){
17                 rows[i].className = "evenrowcolor";
18             }else{
19                 rows[i].className = "oddrowcolor";
20             }
21         }
22     }
23 }
24 window.onload=function() {
25     altRows('alternatecolor');
26 }
27 </script>
28

```

Figura 3. 31: Función `altRows`.

Elaborado por: Autor.

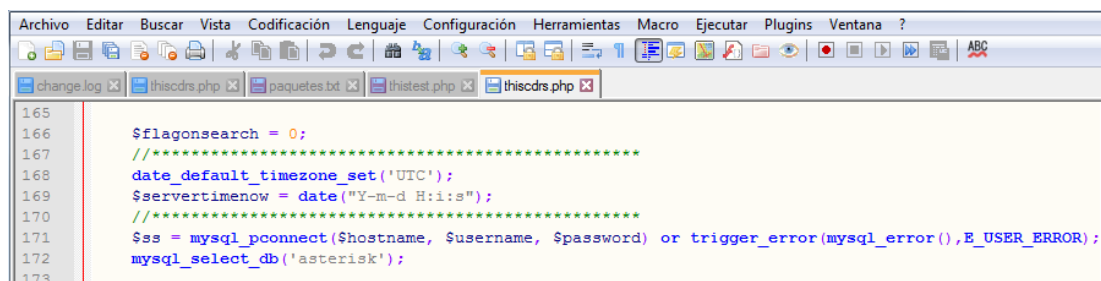
Posteriormente se configura el estilo a la tabla que contiene la información relevante como fecha, origen, destino, etc. (figura 3.32).



```
30
31 <style type="text/css">
32 table.altrowstable {
33     font-family: verdana,arial,sans-serif;
34     font-size:12px;
35     color:#333333;
36     border-width: 1px;
37     border-color: #a9c6c9;
38     border-collapse: collapse;
39     border-spacing: 0px;
40 }
41 table.altrowstable th {
42     border-width: 0px;
43     height: auto;
44     padding: 2px;
45     border-style: solid;
46     border-color: #a9c6c9;
47     border-radius: 5px;
48     font-size:8pt;
49 }
50 table.altrowstable td {
51     border-width: 1px;
52     height: auto;
53     padding: 2px;
54     border-style: solid;
55     border-color: #a9c6c9;
56     font-size:8pt;
57 }
58
```

Figura 3. 32: Configuración de estilo de tabla.
Elaborado por: Autor.

La conexión a la base de datos se realiza mediante la línea de comando que se muestra a continuación en la figura 3.33, esta línea de comando permite acceder a la información alojada en las tablas de MySQL que posteriormente se muestra en pantalla



```
165
166 $flagonsearch = 0;
167 //*****
168 date_default_timezone_set('UTC');
169 $servertimenow = date("Y-m-d H:i:s");
170 //*****
171 $ss = mysql_pconnect($hostname, $username, $password) or trigger_error(mysql_error(),E_USER_ERROR);
172 mysql_select_db('asterisk');
173
```

Figura 3. 33: Conexión a la base de datos.
Elaborado por: Autor

La presentación de la información que se obtiene se realiza mediante una línea de comandos que genera una consulta a la base de datos para recibir la información almacenada en la tabla utilizada por Asterisk, para depositar los CDRs y un script adicional que muestra dichos valores conseguidos correspondientes a cada CDR (figura 3.34).

```

212
213
214 $sql="SELECT * FROM cdr:";
215 $query = mysql_query($sql) or die(mysql_error());
216
217
218 while($field=mysql_fetch_array($query)){
219     echo "<td align='right'>" . $field['calldate'] . "</td>";
220     echo "<td align='right'>" . $field['billsec'] . "</td>";
221     echo "<td align='right'>" . $field['src'] . "</td>";
222     echo "<td align='right'>" . $field['dst'] . "</td>";
223     echo "<td align='right'>" . $field['disposition'] . "</td>";
224
225     //*****
226     if ($field['wavfile'] != "") {
227         echo "<td align='center'>" . "<a href='\" . $field['wavfile'] . \"' download style='text-decoration:none'<font color='\" . $field['wavfile'] . \"'><input type='radio' name='tobeplay' id='\" . $callcounter . \"' onclick='MoveValue()' value='\" . $field['wavfile'] . \"'></td>";
228         $callcounter=$callcounter+1;
229     } else {
230         echo "<td align='center'>" . "" . "</td>";
231         echo "<td align='center'>" . "" . "</td>";
232     }
233
234     //*****
235     if ($field['calldout'] != "") {
236         echo "<td id='callid$rowcounter' align='center'>" . $field['calldout'] . "</td></tr>";
237     } else {
238         echo "<td id='callid$rowcounter' align='center'></td></tr>";
239     }
240
241     //*****
242     $rowcounter = $rowcounter + 1;

```

Figura 3. 34: Presentación de información.
Elaborado por: Autor.

Finalmente se ejecuta una línea de comandos para recargar la página cada cinco minutos y así refrescar la data cargada, sin embargo, en caso de que una reproducción este en curso, ésta se detendrá (figura 3.35).

```

279 <script src="//code.jquery.com/jquery-1.10.2.js"></script>
280 <script src="//code.jquery.com/ui/1.11.4/jquery-ui.js"></script>
281
282 <script>
283     var refresher5min = setInterval(function(){ reFresh5min() }, 300000);
284     function reFresh5min() {
285         var myAudio = document.getElementById('audiowav');
286         if ((myAudio.duration == 0) || myAudio.paused) {
287             document.getElementById("CDRform").submit();
288         }
289     }
290 </script>

```

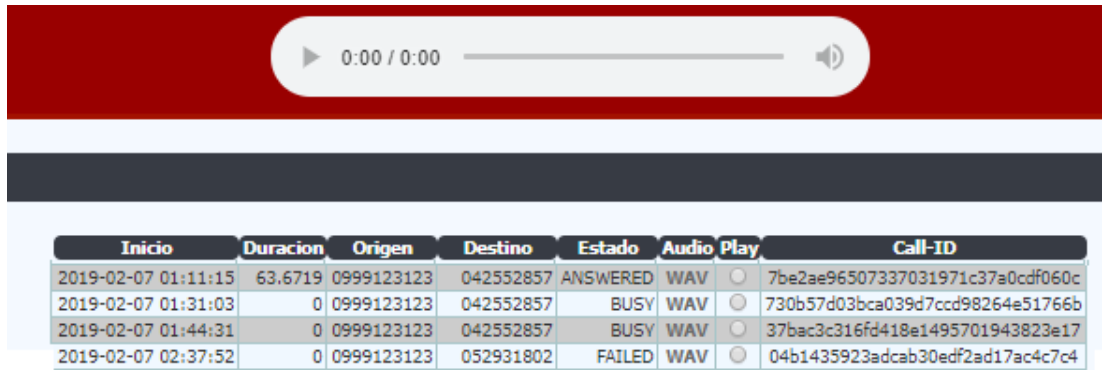
Figura 3. 35: Recarga de página.
Elaborado por: Autor.

3.6. Pruebas y resultados.

A continuación, se muestra las pruebas y resultados obtenidos en la simulación, para lo cual se realizan pruebas técnicas a través del carrier internacional de llamadas, este proveedor ofrece una ruta en la cual se completan las llamadas a destinos fijos de Ecuador, como configuración, existieron dos escenarios de prueba.

El primer escenario (figura 3.36) es una llamada exitosa al número 042552857 que pertenece a la red fija de CNT.

Uno de los campos importantes en la información expuesta es el Call ID de la llamada, un identificador único para cada llamada que permite obtener la señalización de esta en la aplicación Homer.



Inicio	Duracion	Origen	Destino	Estado	Audio	Play	Call-ID
2019-02-07 01:11:15	63.6719	0999123123	042552857	ANSWERED	WAV	<input type="radio"/>	7be2ae96507337031971c37a0cdf060c
2019-02-07 01:31:03	0	0999123123	042552857	BUSY	WAV	<input type="radio"/>	730b57d03bca039d7ccd98264e51766b
2019-02-07 01:44:31	0	0999123123	042552857	BUSY	WAV	<input type="radio"/>	37bac3c316fd418e1495701943823e17
2019-02-07 02:37:52	0	0999123123	052931802	FAILED	WAV	<input type="radio"/>	04b1435923adcab30edf2ad17ac4c7c4

Figura 3. 38: Muestra de resultados.

Elaborado por: Autor.

3.6.1. Análisis de señalización.

En la primera llamada de prueba, la cual se contesta, se logra establecer la comunicación con el proveedor mediante el “invite”, el cual obtiene como respuesta un 100 “trying”, posteriormente cuando el destino es localizado se recibe un 180 y un 183 “session in progress” que genera el tono audible de timbrado, una vez que se contesta la llamada por el destino se recibe un 200 “ok” y se abre el canal de audio, donde comienza el flujo RTP. Completa la llamada por parte del destino final, se envía un “bye”, el cual es aceptado por un 200 “ok” (figura 3.39).



Figura 3. 39: Análisis de señalización de la llamada de prueba.

Elaborado por: Autor.

3.6.2. Calidad de llamada.

3.6.2.1. Puntuación de opinión media.

MOS o Mean Opinion Score le da a VoIP la prueba de un valor numérico como una indicación de la calidad percibida de la voz recibida después de ser transmitida y comprimida usando códecs. Esta medición es el resultado de atributos de red subyacentes que actúan sobre el flujo de datos y es útil para predecir la calidad de la llamada y es una buena herramienta de prueba de VoIP para determinar los problemas que pueden afectar su calidad de VoIP y sus conversaciones. La tabla 3.1 muestra el rango de valores de MOS.

Tabla 3. 1: Comparación de valores de MOS

Rangos de valores MOS	
Muy satisfactorio	4.3 – 5.0
Satisfactorio	4.0 – 4.3
Algunos usuarios satisfechos	3.6 – 4.0
Muchos usuarios insatisfechos	3.1 – 3.6
Casi todos los usuarios insatisfechos	2.6 – 3.1
No recomendado	1.0 – 2.6

Elaborado por: Autor

3.6.2.2. Análisis de calidad.

Si bien es cierto la calidad de la llamada se puede determinar mediante el audio generado, también se puede establecer mediante el factor MOS, en nuestra llamada se nota que existe un factor MOS promedio de 4,40 y 0% de paquetes perdidos, lo cual indica de acuerdo con la escala de medición que la calidad es buena.

También se denota en base al parámetro jitter, la fluctuación del envío de la señal posee un promedio de 3.04 ms (el valor aceptable de jitter no debe sobrepasar los 30 ms), todos estos parámetros están indicados en la figura 3.40.

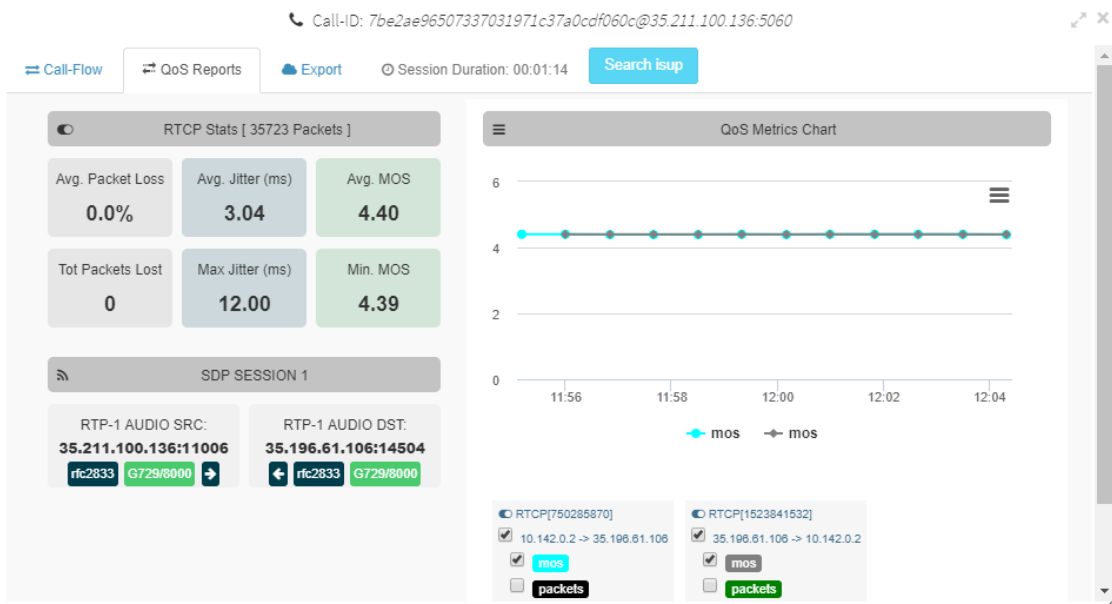


Figura 3. 40: Análisis de calidad.
Elaborado por: Autor.

Dentro del segundo escenario de prueba se obtiene una llamada sin respuesta por el usuario final, en este caso la llamada se libera con el código 480 “temporarily unavailable” (figura 3.41).



Figura 3. 41: Análisis de señalización del segundo escenario de prueba.
Elaborado por. Autor.

Al ser una llamada que no se contesta, no se produce flujo RTP por esta razón no existe información referente al MOS, tal como la figura 3.42 lo muestra.

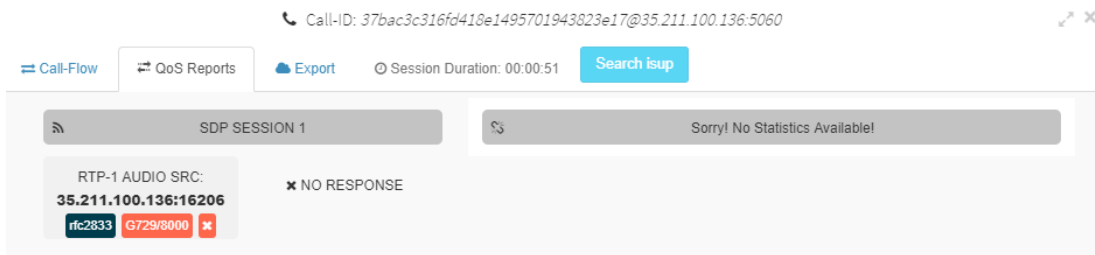


Figura 3. 42: Análisis de calidad dentro del segundo escenario de prueba.
Elaborado por: Autor.

3.6.2.3. Llamada fallida.

En este escenario se denota que la llamada fue rechazada por el proveedor, esto se origina por varias causas, como problemas de capacidad por parte del proveedor, congestión en la red del proveedor, problemas en el switch del proveedor o número invalido.

Como se observa en la señalización, se envía el “invite” al proveedor con la información del número de prueba, este responde con el 100 “trying” e inmediatamente rechaza la llamada con código 503 “service unavaible” (figura 3.43).

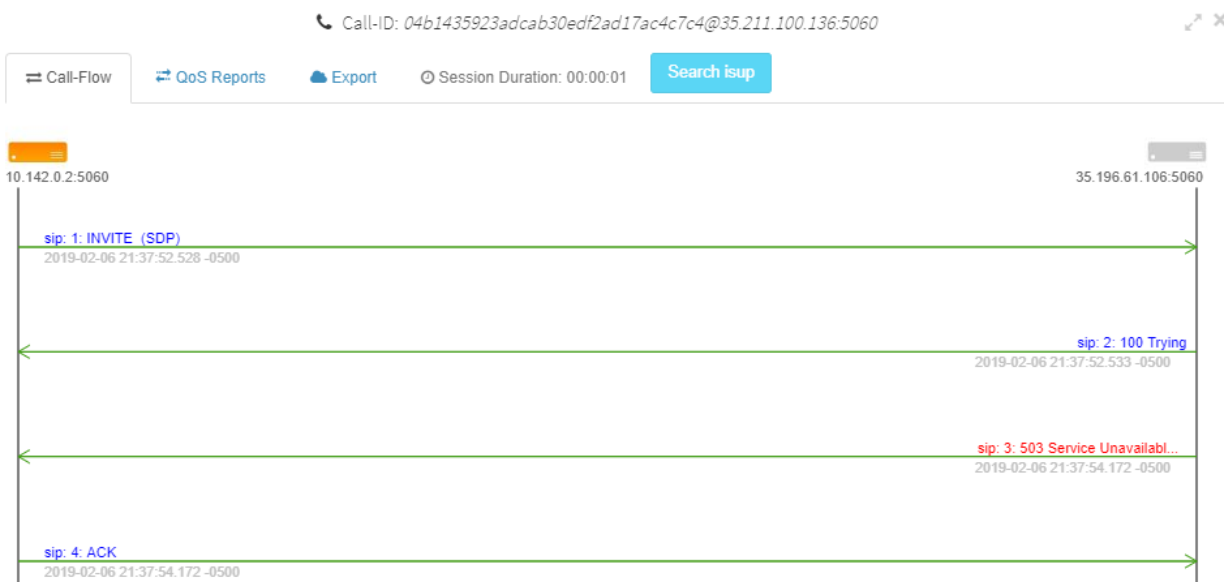


Figura 3. 43: Análisis de señalización del escenario de llamada perdida.
Elaborado por: Autor.

Al ser una llamada que se rechaza, no existe flujo RTP, razón por la cual no se puede cuantificar el MOS (figura 3.44)

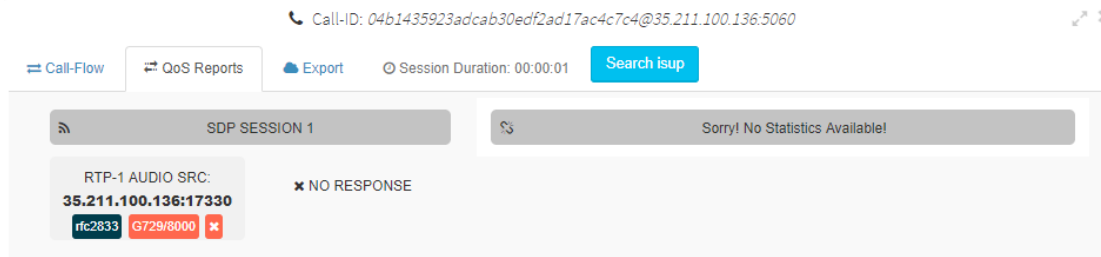


Figura 3. 44: Análisis de calidad dentro del escenario de llamada perdida.
Elaborado por: Autor.

3.6.3. Resultados.

La tabla 3.2 presenta los resultados obtenidos a través de los tres escenarios de prueba realizados anteriormente.

Tabla 3. 2: Resultados dentro de los 3 escenarios de prueba.

ESCENARIO DE PRUEBAS	PÉRDIDA DE PAQUETES PROMEDIO	JITTER PROMEDIO	MOS PROMEDIO
LLAMADA EXITOSA	0.00%	3.04	4.4
LLAMADA SIN RESPUESTA POR EL USUARIO FINAL	0	0	0
LLAMADA RECHAZADA POR EL PROVEEDOR	0	0	0

Elaborado por: Autor.

La tabla 3.2 expone los resultados obtenidos de las pruebas que se realizan bajo las características de los tres escenarios establecidos anteriormente, llamada exitosa, llamada sin respuesta por el usuario final y llamada rechazada por el proveedor. El primer caso exhibe claramente la superioridad que tiene el proceso, pues se obtuvieron resultados muy favorables, resultados arrojados por la aplicación evaluadora que este trabajo de titulación trata, con un valor de jitter excelente de 3.04 ms y una puntuación de opinión media de 4.4, valor que según la tabla 3.1 es muy satisfactorio. En el segundo y tercer escenario existe el rechazo o no respuesta a la llamada, lo que da lugar a la imposibilidad de ser examinada por el evaluador expuesto.

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.

4.1. Conclusiones.

- Esta herramienta brinda un alto grado de confiabilidad en los datos mostrados, pues los valores de MOS son coherentes con la calidad de audio obtenida, siendo esto de gran utilidad para empresas proveedoras de servicios de voz sobre IP que requieran automatizar el proceso de pruebas técnicas que actualmente son realizadas manualmente.
- El proveedor de telefonía IP al cual se le realizaron las pruebas técnicas de la ruta a destinos fijos de Ecuador presentó resultados satisfactorios con un alto valor de MOS, baja latencia de jitter y cero por ciento de pérdida de paquetes, lo cual indica que es un proveedor recomendado para la conexión del tráfico internacional de llamadas con destino a números fijos de Ecuador.
- Dentro de los escenarios de prueba “llamada sin respuesta por el usuario final” y “llamada rechazada por el proveedor” se obtuvieron resultados nulos para los parámetros “pérdida de paquete”, “jitter promedio” y “puntuación de opinión media”.

4.2. Recomendaciones.

- Modificar el código para lograr establecer mayor cantidad de llamadas simultáneas.
- Variar el código para generar la posibilidad de realizar llamadas a terminales móviles.
- Agregar audios pregrabados en varios idiomas para realizar pruebas a carriers internacionales de no habla-hispana.

Bibliografía

- Abdelnur, H., State, R., & Festor, O. (2015). Advanced fuzzing in the VoIP space. *Journal in Computer Virology*, 6(1), 57–64. <https://doi.org/10.1007/s11416-009-0123-7>
- Ahson, S., & Ilyas, M. (Eds.). (2009). *VoIP handbook: applications, technologies, reliability, and security*. Boca Raton: CRC Press.
- Alam, M. Z., Bose, S., Rahman, M. M., & Al-Mumin, M. A. (2007). Small Office PBX Using Voice Over Internet Protocol (VOIP). En *The 9th International Conference on Advanced Communication Technology* (pp. 1618–1622). Gangwon-Do, Korea: IEEE. <https://doi.org/10.1109/ICACT.2007.358679>
- Bourreau, M., & Doğan, P. (2014). Service-based vs. facility-based competition in local access networks. *Information Economics and Policy*, 16(2), 287–306. <https://doi.org/10.1016/j.infoecopol.2003.05.002>
- BullsEye Telecom. (2016). TDM vs.PBX vs. Vol: What Do These Terms Mean? Recuperado de <https://www.bullseyetelecom.com/blog/tdm-vs-pbx-vs-voi-what-do-these-terms-mean/>
- Butcher, D., Li, X., & Guo, J. (2014). Security Challenge and Defense in VoIP Infrastructures. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 37(6), 1152–1162. <https://doi.org/10.1109/TSMCC.2007.905853>
- Cai, L., Xiao, Y., (Sherman) Shen, X., Cai, L., & W. Mark, J. (2014). VoIP over WLAN: voice capacity, admission control, QoS, and MAC. *International Journal of Communication Systems*, 19(4), 491–508. <https://doi.org/10.1002/dac.801>
- CenturyLink Advanced Technology Solutions. (2016). The Business Benefits of VoIP.

- CounterPath USA. (2018). VoIP Solutions for Unified Communications & Collaboration (UC) | CounterPath. Recuperado el 10 de enero de 2019, de <https://www.counterpath.com/>
- Datarkar, T., Bobade, N. P., & Gaikwad, M. A. (2015). Voice over Internet Protocol (VOIP) Based On Asterisk, 4.
- Department of Business Administration, East West University, Bangladesh, Masudur Rahman, M., & Sarwar Islam, N. (2014). VoIP Implementation Using Asterisk PBX. *IOSR Journal of Business and Management*, 15(6), 47–53. <https://doi.org/10.9790/487X-1564753>
- Ergen, M., Ergen, S. C., & Varaiya, P. (2013). Throughput performance of a wireless VoIP model with packet aggregation in IEEE 802.11. En *IEEE Wireless Communications and Networking Conference, 2013. WCNC 2013*. (pp. 2235–2239). Las Vegas, NV, USA: IEEE. <https://doi.org/10.1109/WCNC.2006.1696643>
- Garuba, M., Li, J., & Yi, Z. (2013). Security in the New Era of Telecommunication: Threats, Risks and Controls of VoIP. En *Fifth International Conference on Information Technology: New Generations* (pp. 587–591). Las Vegas, NV, USA: IEEE. <https://doi.org/10.1109/ITNG.2008.222>
- Ghafarian, A., & Jackson, C. (2014). Securing Voice over Internet Protocol, 11.
- Ghossoon, W. (2017). Asterisk Open Source to Implement, 4.
- Habib, S., Nirmala, B., & Nirmala, M. (2018). Improving QoS of VoIP over WLAN (IQ-VW). *University of Colorado at Colorado Springs*.
- Hung, P. C. K., & Martin, M. V. (2016). Security Issues in VOIP Applications. En *2016 Canadian Conference on Electrical and Computer Engineering* (pp. 2361–2364). Ottawa, Ont.: IEEE. <https://doi.org/10.1109/CCECE.2006.277789>

- Imran, A., Qadeer, M. A., & Khan, M. (2015). Asterisk VoIP private branch exchange. En *2015 International Multimedia, Signal Processing and Communication Technologies* (pp. 217–220). Aligarh: IEEE. <https://doi.org/10.1109/MSPCT.2009.5164214>
- Jing Liu, & Zhisheng Niu. (2014). An adaptive receiver buffer adjust algorithm for VoIP applications considering voice characters. En *APCC/MDMC '14. The 2014 Joint Conference of the 10th Asia-Pacific Conference on Communications and the 5th International Symposium on Multi-Dimensional Mobile Communications Proceeding* (Vol. 2, pp. 597–601). Beijing, China: IEEE. <https://doi.org/10.1109/APCC.2004.1391784>
- Karapantazis, S., & Pavlidou, F.-N. (2015). VoIP: A comprehensive survey on a promising technology. *Computer Networks*, 53(12), 2050–2090. <https://doi.org/10.1016/j.comnet.2009.03.010>
- Kazemitabar, H., Ahmed, S., & Nisar, K. (2013). A comprehensive review on VoIP over Wireless LAN networks. *Computer Science Letters*.
- Korhonen, J., & Ye Wang. (2015). Effect of packet size on loss rate and delay in wireless links. En *IEEE Wireless Communications and Networking Conference, 2015* (Vol. 3, pp. 1608–1613). New Orleans, LA, USA: IEEE. <https://doi.org/10.1109/WCNC.2005.1424754>
- Kumar, A. (2013). An Overview of Voice Over Internet Protocol (VoIP). *RIVIER COLLEGE ONLINE ACADEMIC JOURNAL*.
- Lustosa, L. C. G., Souza, A. A. D. P., de A. Rodrigues, P. H., & Quinellato, D. G. (2008). ACME: An Automated Tool for Generating and Evaluating the Quality of VoIP Calls. En G. Pavlou, T. Ahmed, & T. Dagiuklas (Eds.), *Management of Converged Multimedia Networks and Services* (Vol. 5274, pp. 91–103). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-87359-4_9

- Masjedi, Y. (2013). The Benefits of Asterisk and VoIP. Recuperado el 10 de enero de 2019, de <https://www.nextiva.com/blog/the-benefits-of-asterisk-and-voip.html>
- Matango, F. (2016). Protocolo SIP. Recuperado el 11 de febrero de 2019, de <http://www.servervoip.com/blog/tag/sip-protocol/>
- McNeill, K., Liu, M., & Rodriguez, J. (2016). An Adaptive Jitter Buffer Play-Out Scheme to Improve VoIP Quality in Wireless Networks. En *MILCOM 2016* (pp. 1–5). Washington, DC, USA: IEEE. <https://doi.org/10.1109/MILCOM.2006.302119>
- Meggelen, J. V., Madsen, L., & Smith, J. (2014). *Asterisk: the future of telephony* (2nd ed). Sebastopol, CA: O'Reilly.
- Mintandjian, L., & Naylor, P. (2016). A study of echo in VoIP systems and synchronous convergence of the μ -law PNLMS algorithm. *EURASIP*.
- Nasr, M. E., & Napoleon, S. A. (2014). On improving voice quality degraded by packet loss in data networks. En *2014 IEEE Africon. 7th Africon Conference in Africa (IEEE Cat. No.04CH37590)* (Vol. 1, pp. 51–55). Gaborone, Botswana: IEEE. <https://doi.org/10.1109/AFRICON.2004.1406632>
- Ni, Q., Romdhani, L., & Turletti, T. (2014). A survey of QoS enhancements for IEEE 802.11 wireless LAN. *Wireless Communications and Mobile Computing*, 4(5), 547–566. <https://doi.org/10.1002/wcm.196>
- Packetizer Inc. (2019). Understanding VoIP - VoIP Protocols. Recuperado el 15 de enero de 2019, de https://www.packetizer.com/ipmc/papers/understanding_voip/voip_protocols.html
- Phithakkitnukoon, S., Dantu, R., & Baatarjav, E.-A. (2015). VoIP Security — Attacks and Solutions. *Information Security Journal: A Global*

Perspective, 17(3), 114–123.
<https://doi.org/10.1080/19393550802308618>

Qadeer, M. A., & Imran, A. (2008). Asterisk Voice Exchange: An Alternative to Conventional EPBX. En *2008 International Conference on Computer and Electrical Engineering* (pp. 652–656). Phuket, Thailand: IEEE. <https://doi.org/10.1109/ICCEE.2008.176>

Rango, F. D., Tropea, M., Fazio, P., & Marano, S. (2016). Overview on VoIP: Subjective and Objective Measurement Methods, 14.

Reynolds, R. J. B., & Rix, A. W. (2014). Quality VoIP — an engineering challenge. *BT Technology Journal*, 19(2), 23–32. <https://doi.org/10.1023/A:1011382801542>

Sahabudin, S., & Alias, M. Y. (2014). End-to end delay performance analysis of various codecs on VoIP Quality of Service. En *2014 IEEE 9th Malaysia International Conference on Communications (MICC)* (pp. 607–612). Kuala Lumpur, Malaysia: IEEE. <https://doi.org/10.1109/MICC.2009.5431426>

Sanneck, H. (2016). Selective Packet Prioritization for Wireless Voice over IP, 9.

Sonkar, S. K., Singh, R., Chauhan, R., & Singh, A. P. (2013). A Review Paper: Security on Voice over Internet Protocol from Spoofing attacks, 1(3), 8.

Stanton, R. (2016). Secure VoIP – an achievable goal. *Computer Fraud & Security*, 2016(4), 11–14. [https://doi.org/10.1016/S1361-3723\(06\)70333-5](https://doi.org/10.1016/S1361-3723(06)70333-5)

Swale, R. (2015). VoIP — Panacea or PIG's Ear? *BT Technology Journal*.

TechTarget Inc. (2019). Learning guide: VoIP protocols and standards. Recuperado el 15 de enero de 2019, de

<https://searchunifiedcommunications.techtarget.com/tutorial/Learning-guide-VoIP-protocols-and-standards>

Thorne, D. J. (2001). VoIP — The Access Dimension. *BT Technology Journal*, 19(2), 33–43. <https://doi.org/10.1023/A:1011386902451>

Xiuzhong Chen, Chunfeng Wang, Dong Xuan, Zhongcheng Li, Yinghua Min, & Wei Zhao. (2013). Survey on QoS management of VoIP. En *2013 International Conference on Computer Networks and Mobile Computing, 2003. ICCNMC 2003*. (pp. 69–77). Shanghai, China: IEEE Comput. Soc. <https://doi.org/10.1109/ICCNMC.2003.1243029>

Yong-feng, H., & Jiang-ling, Z. (2015). Implementation of ITU-T G. 729 speech codec in IP telephony gateway. *Wuhan University Journal of Natural Sciences*, 5(2), 159–163. <https://doi.org/10.1007/BF02827918>



DECLARACIÓN Y AUTORIZACIÓN

Yo, **Grijalva Arias, Leslie Elizabeth** con C.C: # 094156701-8 autor del Trabajo de Titulación: **Diseño e implementación de una herramienta generadora de llamadas automáticas para medir la calidad del servicio de un proveedor internacional de telefonía VoIP**, previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 11 de Marzo de 2018

f. _____

Nombre: Grijalva Arias, Leslie Elizabeth

C.C: 094156701-8

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Diseño e implementación de una herramienta generadora de llamadas automáticas para medir la calidad del servicio de un proveedor internacional de telefonía VoIP.		
AUTOR(ES)	GRIJALVA ARIAS, LESLIE ELIZABETH		
REVISOR(ES)/TUTOR(ES)	M. Sc. ZAMORA CEDEÑO, NÉSTOR ARMANDO		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería en Telecomunicaciones		
TÍTULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	11 de Marzo de 2019	No. DE PÁGINAS:	79
ÁREAS TEMÁTICAS:	Telefonía IP		
PALABRAS CLAVES/ KEYWORDS:	ASTERISK, VOIP, LLAMADA, HERRAMIENTA, PROGRAMACIÓN, TELEFONÍA.		
RESUMEN/ABSTRACT:	<p>Es posible que una infraestructura de comunicación VoIP deba servir a un número sustancial de usuarios con una calidad de voz aceptable. El presente trabajo de titulación investiga la idoneidad y calidad del servicio de proveedores internacionales de telefonía de voz sobre IP mediante el servidor PBX de Asterisk para proporcionar capacidades de comunicación con una calidad aceptable de Puntuación de Opinión Media (MOS), utilizado para evaluar la calidad de las llamadas de voz, Jitter promedio y Pérdida de Paquetes promedio, para un gran número de usuarios. Los resultados muestran la idoneidad de tres escenarios: llamada exitosa, llamada sin respuesta por el usuario final y llamada rechazada por el proveedor, para caracterizar la capacidad del servidor PBX de Asterisk. Los resultados experimentales han demostrado que el PBX de Asterisk puede manejar efectivamente gran cantidad de llamadas de voz, mientras que proporciona un MOS promedio superior a 4.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593-9-39730128	E-mail: leslie_grijalva@hotmail.com	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: M. Sc. Palacios Meléndez, Edwin Fernando		
	Teléfono: +593-9-67608298		
	E-mail: edwin.palacios@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			