

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN PROTOTIPO DE PBX USANDO SOFTWARE LIBRE

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN REDES Y TELECOMUNICACIONES**

ANTHONY ESTEBAN SUNTAXI ÑATO

anthony.suntaxi@epn.edu.ec

DIRECTOR: CARLOS ANDRÉS YUNGA SÁNCHEZ

carlos.yunga@epn.edu.ec

DMQ, febrero 2024

CERTIFICACIONES

Yo, ANTHONY ESTEBAN SUNTAXI ÑATO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Anthony Esteban Sntaxi Ñato

anthony.sntaxi@epn.ed.ec

estebansunta1502@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por ANTHONY ESTEBAN SUNTAXI ÑATO, bajo mi supervisión.

Carlos Andrés Yunga Sánchez

DIRECTOR

carlos.yunga@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Anthony Esteban Sntaxi Ñato

anthony.sntaxi@epn.edu.ec

estebansunta1502@gmail.com

CI: 1721850574

DEDICATORIA

En primer lugar, a Dios por guiarme por el camino del bien para poder culminar mis estudios, por darme salud y fuerza para no rendirme a mitad del camino.

A mis padres Luis y Rosa, por su sacrificio y por siempre estar en mis peores momentos alentándome y dándome su amor incondicional para poder cumplir mi meta.

A mi hermano Santiago, por ser un gran ejemplo y por hacerme entender que por más grande que sea el problema no debo rendirme y debo buscar una solución que me ayude a seguir adelante.

Esto es gracias a ustedes los amo mucho.

Anthony Esteban Suntaxi Ñato

AGRADECIMIENTO

A mi tutor de carrera, Ing. Leandro Pazmiño por guiarme de la mejor manera para poder finalizar mis estudios.

A mi tutor de tesis, Ing. Carlos Yunga por alentarme, guiarme y corregirme en todo este proceso para realizar un excelente trabajo.

A mis profesores, por brindarme sus conocimientos a lo largo de toda la carrera y formarme como estudiante.

A mi Ing. Pedro Catagña, por abrirme las puertas de su empresa y brindarme su confianza para poder aprender más cosas referentes a mi carrera.

Anthony Esteban Suntaxi Ñato

ÍNDICE DE CONTENIDOS

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDOS	V
RESUMEN	VIII
<i>ABSTRACT</i>	IX
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
1.4 Marco Teórico.....	2
Central Telefónica Privada (PBX)	2
VoIP	2
Telefonía IP	3
Códexs	3
G.722.....	3
WAV (<i>Waveform Audio File Format</i>)	4
Software	4
Softphone.....	4
Asterisk	4
Librerías de Asterisk.....	4
Sistema Operativo	5
Ubuntu	5
Hardware.....	5
Raspberry pi 4.....	5
Router	5

	<i>Patch cord</i> (cable de red)	6
	Tarjeta microSD	6
	Aplicaciones	6
	ZoiPer	6
2	METODOLOGÍA	7
3	RESULTADOS	8
3.1	Investigar las soluciones existentes de software libre para el despliegue de PBXs	8
	Análisis de plataformas de softwares libre	8
	Análisis del software libre Asterisk	8
	Análisis de software libre Elastix	9
	Análisis de software libre FreePBX	9
	Análisis de software libre Issabel.....	10
3.2	Seleccionar el hardware y software para el servidor y clientes acorde a los requerimientos establecidos.....	10
	Selección de hardware	10
	Selección del raspberry pi	10
	Selección de tarjeta microSD	11
	Selección del dispositivo para la comunicación de voz.....	12
	Selección de Software	13
	Selección de sistema Operativo Ubuntu.....	13
	Selección del software Asterisk.....	14
	Selección de Softphone a utilizarse.....	15
3.3	Configurar el servidor PBX basado software libre	17
	Booteo del sistema operativo para el raspberry pi 4	17
	Configuración de dirección estática	21
	Acceso al CLI de Ubuntu para la instalación de Asterisk	22
	Configuración de Asterisk para la central telefónica privada	29
3.4	Implementación del prototipo	33

Esquema del prototipo	33
Instalación y configuración del softphone seleccionado	34
Instalación de ZoiPer.....	34
Configuración de ZoiPer.....	35
Verificación de la comunicación de voz entre los 3 clientes.....	38
3.5 Realizar pruebas de funcionamiento del prototipo	40
Conexión y configuración para cada dispositivo.	40
Prueba de funcionamiento desde la extensión 1001 a 1002	41
Prueba de funcionamiento desde la extensión 1002 a 1001	41
Prueba de funcionamiento desde la extensión 1002 a 1003	42
Prueba de funcionamiento desde la extensión 1003 a 1001	43
4 CONCLUSIONES	44
5 RECOMENDACIONES.....	46
6 Referencias BIBLIOGRÁFICAS	47
7 ANEXOS.....	50
ANEXO I: Certificado de Originalidad	i
ANEXO II: Enlaces	ii
ANEXO III: Códigos Fuente	iii

RESUMEN

El siguiente proyecto de titulación trata sobre la implementación de una central telefónica privada (PBX), el cual se va a realizar mediante un raspberry pi 4 y una tarjeta microSD “booteada” en donde se instalará el sistema operativo Ubuntu, para instalar el software libre Asterisk que va a permitir la comunicación de voz entre varios usuarios.

En la primera sección, se redactan conceptos puntuales, donde se abarcan los objetivos principales y secundarios, para tener un mejor entendimiento de lo que va a tratar la implementación del prototipo, al igual que conceptos puntuales de los elementos que se van a utilizar en el proyecto.

En la segunda sección, se explica la metodología en donde se detallará brevemente todo el proceso del prototipo, garantizando el cumplimiento de los diferentes objetivos planteados en el proyecto.

En la tercera sección, se sustentará los resultados alcanzados, aquí se explica lo necesario que se utiliza para el prototipo. Se presentará el software al igual que hardware seleccionado que se utilizará para el desarrollo del prototipo PBX.

En la cuarta sección, se detallarán las diferentes conclusiones y recomendaciones plateadas de todo el proyecto, ya que aquí se expresa un resumen de las partes más destacadas de los resultados, al igual que las sugerencias para que se pueda mejorar la implementación del prototipo en futuros trabajos.

En las últimas secciones se encuentran todo el material bibliográfico, donde se acudió para tener conocimientos claros y poder redactar de una manera específica cada uno de los elementos planteados del prototipo.

PALABRAS CLAVE: *Asterisk, Raspberry pi 4, Tarjeta microSD, Telefonía IP, Ubuntu.*

ABSTRACT

The following degree project is about the implementation of a private telephone exchange (PBX), which will be carried out using a raspberry pi 4 and a “booted” microSD card where the Ubuntu operating system will be installed, to install the free software Asterisk that will allow voice communication between several users.

In the first section, specific concepts are written, which cover the main and secondary objectives, to have a better understanding of what the implementation of the prototype will deal with, as well as specific concepts of the elements that will be used in the project.

In the second section, the methodology is explained where the entire prototype process will be briefly detailed, guaranteeing compliance with the different objectives set in the project.

In the third section, the results achieved will be supported, here what is necessary to use for the prototype is explained. The software will be presented as well as selected hardware that will be used for the development of the PBX prototype.

In the fourth section, the different conclusions and recommendations of the entire project will be detailed, since here a summary of the most outstanding parts of the results is expressed, as well as suggestions so that the implementation of the prototype can be improved in future. jobs.

In the last section all the bibliographic material is found, where we went to have clear knowledge and to be able to write each of the proposed elements of the prototype in a specific way.

KEYWORDS: Asterisk, Raspberry pi 4, micro-SD card, IP Telephony, Ubuntu.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La comunicación mediante telefonía IP ha progresado a medida que pasa de tiempo, por lo que se dirigen a soluciones más flexibles y avanzadas para empresas y hogares. Los sistemas que se basan en tecnología de telefonía IP han sido la migración de los sistemas de telefonía tradicional, ya que una llamada al ser transmitida mediante una red de datos, ofrece una mejor eficiencia con una reducción de costos considerable.

Para implementar el prototipo se necesita de algunos puntos importantes, con los que debe contar la empresa o domicilio, ya que, para el desarrollo de la central telefónica privada, se tiene que contar con una conexión a internet ya sea inalámbrica o cableada, junto a dispositivos que ayuden a tener conexión de voz como teléfonos inteligentes.

Para esta ocasión se va a implementar un prototipo de una central telefónica privada en cualquier software libre, ya que nos va a permitir la comunicación entre diferentes usuarios en un lugar determinado. La comunicación se realizará mediante los teléfonos inteligentes con una aplicación "ZoiPer" usando direcciones IP. Se debe tomar en cuenta que nuestra central telefónica privada debe tener su propia dirección (IP estática), ya que esto nos va a permitir tener una conexión fija entre los teléfonos que estarán interconectados para la entrada y salida de llamadas.

Para el prototipo se utilizará un raspberry pi 4 y una tarjeta micro SD de 64GB, para que no se presente ningún inconveniente al momento de descargar los archivos y repositorios que va a necesitar Asterisk para el despliegue del servicio de telefonía, se necesitara de un router y cables ethernet para armar la estructura de la red que estará conectada al raspberry pi 4 y así poder dar conexión a nuestra central, ya que la conexión de datos se va a dar mediante direcciones IP que estén en el rango de la red.

Luego de configurar el software y los dispositivos inteligentes (celulares), se realizará diferentes pruebas de funcionamiento del prototipo, como la conexión de voz y la eficiencia que tendrá cada llamada al momento de realizarse entre los diferentes usuarios conectados.

1.1 Objetivo general

Implementar un prototipo de PBX usando software libre.

1.2 Objetivos específicos

- Investigar las soluciones existentes del software libre para el despliegue de PBXs.
- Seleccionar el hardware y software para el servidor y el software para los clientes acorde a los requerimientos establecidos.
- Configurar el servidor PBX basado en software libre.
- Implementar el prototipo.
- Realizar pruebas de funcionamiento del prototipo.

1.3 Alcance

Por medio del presente proyecto se va a realizar la implementación de un prototipo de PBX por medio de software libre. Esto permitirá aprovechar la infraestructura de red para desplegar el servicio de telefonía disminuyendo los costos que esto implique. El prototipo va a contar con las siguientes características:

- Implementación del servidor de telefonía privada.
- Instalación y configuración de los clientes para el servicio de telefonía.

1.4 Marco Teórico

Central Telefónica Privada (PBX)

Es una red telefónica que se conecta directamente hacia una red pública mediante líneas troncales. La red telefónica privada se puede implementar en hogares o empresas para que los usuarios se puedan comunicar de manera interna y externa, mediante distintos canales de comunicación de VoIP. Una gran ventaja de una PBX es que permite tener un número mayor de teléfonos que líneas telefónicas fijas. Brinda servicios como llamadas, grabaciones de llamadas, buzón de voz, etc [1].

VoIP

Es una tecnología que transmite comunicación de voz mediante internet, usando la red como un medio de transmisión, VoIP ha tenido grandes avances a lo largo del tiempo ya que por la gran popularidad que obtuvo se posicionó como telefonía tradicional para

empresas y hogares. Esta tecnología se utiliza inalámbricamente o por medios guiados (Cable Ethernet), ya que se utilizan en aplicaciones en la que el envío de audio se produce en tiempo real [2].

Telefonía IP

Es un servicio que aparece como un dilema para las comunicaciones de voz que surge como una opción accesible para las personas, prometiendo aplicaciones nuevas y tarifas más bajas. La Telefonía IP no solo transmite voz, sino también datos, ya que trabaja con conmutación de paquetes, lo que facilita el uso de redes informáticas que están basadas en el protocolo IP, para poder ejecutar llamadas telefónicas dentro de una red multiservicios o convergente.

Para realizar llamadas, el proveedor asigna un número al prestador de servicio y para eso debe acotar las recomendaciones de la E.164, es decir de la unión internacional de telecomunicaciones (UIT), con la finalidad de que el número sea permitido portar a cualquier sitio. Este servicio nos posibilita la realización y recepción de llamadas de otros usuarios, así como diferentes prestaciones por medio de la red de internet o mediante una red privada [3].

Códecs

La palabra "Código" se produce de una abreviatura de "CODificador-DECodificador" o "COMpresor-DECompresor", si nos referimos a la tecnología de la información un códec es un elemento que tiene como funcionalidad codificar y decodificar los datos digitales, con la finalidad de transmitir, almacenar o hacer una eficiente compresión de la información.

Los códecs referentes a la comunicación de voz y video, comprimen y descomprimen la información, proporcionando que la transmisión sea eficiente a través de las redes, ya que abrevia los datos necesarios para interpretar la señal. Dentro de VoIP se los conoce como "Vocoders". Por lo general los códecs se usan en emisiones de los medios de comunicación y también para videoconferencias, ya que la voz se decodifica para poder transmitir por cualquier red IP [4]. El códec, el formato y la red de voz que se utilizó para la implementación de la central telefónica son los siguientes:

G.722

Fue desarrollado por la UIT, el siguiente estándar se maneja comúnmente en los sistemas de telefonía IP, ya que la calidad de audio que proporciona es eficiente, es decir comparte calidad en alta definición. Trabaja en tasas de bits muchas más altas a

comparación de versiones antiguas, por lo que ofrece una reproducción de voz más clara, ya que opera en los 48,56 y 64 Kbits/s [5].

WAV (Waveform Audio File Format)

Este formato de audio fue creado por IBM y Microsoft, surgió en el año de 1995. Es un tipo de archivo que guarda muestras en un rango de 8 a 16 bits de audio digital. También presenta tres frecuencias de muestro de 11.025, 33.05 y 44.1 kHz. Su desventaja es que es un archivo que ocupa cantidades considerables de disco duro, ya que la calidad que ofrece es muy buena. Este archivo con el pasar del tiempo se convirtió en un estándar para las grabaciones de música en Cd's [6].

Software

Softphone

Es una herramienta con funcionalidades similares a un teléfono celular, se basa en la tecnología VoIP con la finalidad de recibir y realizar llamadas, teniendo en cuenta que la comunicación que se realice hacia otros softphones, deben estar trabajando sobre el mismo protocolo SIP. Los softphones se caracterizan por su flexibilidad al momento de ser configurados. Existe una gran diversidad de softphones en el mercado como: ZoiPer, 3cx, MicroSIP, entre otras. con la única diferencia que algunos modelos son libres y otros son de paga [7].

Asterisk

Es el software para implementar una central telefónica privada (PBX), mediante una plataforma Unix o Linux, que se conecta a la red telefónica pública conmutada (PSTN), ya que brinda una estructura física para realizar y recibir llamadas. Asterisk brinda una conexión entre VoIP y la PSTN en tiempo real. Es denominada como una App de código abierto que tiene licencia "GPL" la cual implemento Marc Spencer, se usa para desarrollar soluciones de sistemas de comunicaciones, ya que desarrolla una plataforma para las diferentes aplicaciones de telefonía, incluyendo los servicios VoIP [8].

Librerías de Asterisk

Sip.conf: es un archivo con la funcionalidad de configurar los canales SIP cual conlleva distintos parámetros dentro de la configuración SIP para Asterisk, ya que define variables generales como los servidores SIP, clientes y la estructura. Dentro del archivo se configuran los usuarios que estarán conectados a Asterisk y también se hará una configuración para el encaminamiento de las llamadas [9].

Extension.conf: es un archivo con la finalidad de configurar comportamiento de las conexiones que se van a realizar en la central telefónica privada mediante la IP, es decir se va a configurar el plan de marcado, los parámetros del archivo es gestionar, controlar y por último encaminar cada llamada que se va a realizar a las diferentes extensiones tanto en llamadas entrantes como en llamadas salientes dentro del sistema Asterisk [9].

Sistema Operativo

Ubuntu

El Sistema Operativo Ubuntu es una de las más grandes distribuciones que tiene Linux. Ubuntu empezó enfocado en un proyecto, con la finalidad de tener una distribución que no sea tan complicada de manejar, es decir que sea completa e innovadora. Es por esto que con las actualizaciones nuevas los desarrolladores han tenido un manejo mucho más sencillo, ya que cada versión tiene aplicaciones de administración más completas, por lo que Ubuntu hoy en día lo puede manejar cualquier usuario ya que es un sistema operativo con interfaz gráfica [10].

Hardware

Raspberry pi 4

Es un elemento conocido como un mini computador con un costo accesible en comparación a una computadora normal, en otras palabras, se lo puede describir como un ordenador de un tamaño reducido. Este dispositivo fue elaborado en el Reino Unido en el 2011 por una fundación llamada raspberry pi, existen varios modelos dentro del mercado como el raspberry pi 3 y 4 modelo B, raspberry pi Zero W, raspberry pi 400 y Raspberry pi Zero 2W. Los modelos van dependiendo desde la capacidad de trabajo que maneje cada uno, es decir, si es un minicomputador con un gran rendimiento va a tener un mayor costo que alguno con un rendimiento bajo [11].

Router

El router es un dispositivo muy conocido en la actualidad por su capacidad de brindar acceso a internet. Son dispositivos que los proveedores de servicio a internet (ISP) facilitan a los usuarios en sus residencias para poder dar servicio de conexión de internet. Un router indica el camino más óptimo por donde deber ir los mensajes desde un dispositivo hacia otro. Los routers asignan direcciones IP de forma dinámica a los dispositivos para que se conecten a su red, dando así el acceso a internet para realizar cualquier ejecución [12].

Patch cord (cable de red)

Es un cable el cual conlleva interiormente cuatro pares de cable con conectores Rj45 en los dos extremos, fue elaborado para facilitar las interconexiones entre dispositivos dentro de una red LAN. El patch cord es utilizado para transferir datos a velocidades muy elevadas desde un dispositivo de red hacia otro. El cable UTP usado para los patch cord es el cable de categoría 5e ya que nos pueden brindar una velocidad de hasta 100Mbps, ya que para llegar a tener una categoría deben cumplir diferentes estándares internacionales de fabricación, lo que permite que sea esencial para poder trabajar en la red de área local [13].

Tarjeta microSD

La tarjeta microSD es un dispositivo en el cual se puede almacenar diferente tipo de información como música, videos, aplicaciones, imágenes, etc. A este tipo de tarjeta se las comercializan de tres maneras diferentes, de la cual lo único diferente es la dimensión de la misma. La tarjeta más grande se la denomina como "SD estándar", a la mediana se la conoce como "miniSD" y a la más pequeña como "microSD". Poseen 9 pines, uno es para enviar comandos, otro es de reloj, otros cuatro son para el almacenamiento de datos y el resto es para su alimentación [14].

Aplicaciones

ZoiPer

Es una aplicación que facilita que las computadoras y los equipos móviles que se conecten al servidor Asterisk, mediante una red de internet, la dirección IP, el numero de la extensión que se le asigna en el "sip.conf" y la contraseña asignada para cada extensión. Se debe tener en cuenta que para el funcionamiento correcta de la aplicación se necesita configurar una IP estática ya que esta nos va a permitir que no se corrompa la conexión automática con la central telefónica privada (PBX) [15].

2 METODOLOGÍA

De acuerdo al plan de trabajo asignado, se llevará a cabo el proyecto de manera ordenada según los puntos establecidos, los mismos que serán explicados detalladamente con la finalidad de dar un fácil entendimiento al lector.

Como primer punto se realiza un análisis profundo del software libre que se va a seleccionar, con la finalidad de montar una central telefónica privada (PBX), es decir se va a revisar ciertos parámetros como los protocolos que maneja o qué tipo de aplicaciones puede soportar, ya que para este proyecto se va a utilizar el softphone Zoiper como medio de comunicación entre los usuarios.

En el punto dos, se selecciona el hardware y software para el servidor y los clientes, elegimos Asterisk como el software, ya que cumple con todas las condiciones para una central telefónica privada. En la selección de hardware se escogió un raspberry pi y una tarjeta microSD para levantar el servicio de telefonía IP en tres celulares en donde se montará la aplicación ZoiPer, para poder conectarnos a la central con el propósito de brindar una perspectiva más auténtica del prototipo.

En el punto tres, se lleva a cabo las configuraciones de servidor PBX. Para esto en el dispositivo raspberry pi 4 se coloca una tarjeta microSD booteada con el sistema Ubuntu ya instalado, en donde se va a instalar el software libre Asterisk para levantar la central telefónica, para esto se necesitará de una red de internet la cual estará conectada directamente mediante un cable ethernet al raspberry pi.

En el punto cuatro, se implementará el prototipo con todas las configuraciones correspondientes para empezar a realizar las pruebas de funcionamientos con los software y hardware seleccionados.

Por último, se procede a dar funcionamiento a nuestra central telefónica privada y realizar las pruebas necesarias para que el prototipo funcione de manera correcta. Se realizó las llamadas respectivas entre los tres teléfonos para verificar la conexión de voz y ver si no se presentan ningún error al momento de su funcionamiento y en caso de que existan poder solucionarlos de la mejor manera, para garantizar la calidad del plan de trabajo.

3 RESULTADOS

A continuación, se va a presentar todo el procedimiento realizado para el correcto funcionamiento de la central telefónica privada. Se presentarán las características principales que tiene cada hardware y software para la implementación. Una vez seleccionado los elementos, se explicará el desarrollo de cada paso a seguir para el levantamiento del servicio en la raspberry pi 4.

Se realizará un diseño en el cual el raspberry pi 4 será presentado como una pequeña central telefónica y una tarjeta microSD la cual actuara como un cerebro principal en donde serán guardado toda la información y a su vez en donde los teléfonos celulares estarán configurados con su extensión y clave respectiva. El diseño nos brindará una comunicación de voz interna mediante una red de internet para tener acceso a las llamadas. Al final se presentaras distintas pruebas de su correcto funcionamiento, con la finalidad de mostrar que el sistema sea eficiente.

3.1 Investigar las soluciones existentes de software libre para el despliegue de PBXs

Acorde a los análisis realizados de los softwares libres, identificamos todos los puntos necesarios para optar por una central telefónica de alta gama, con la finalidad de brindar a los usuarios una calidad de audio eficiente con un costo tolerable.

Análisis de plataformas de softwares libre

Análisis del software libre Asterisk

Dentro del análisis de Asterisk detallamos puntos importantes para implementar la central telefónica privada ya que es compatible con líneas de telefonía IP y líneas de telefonía tradicionales. Asterisk es un software flexible y es por eso que cada empresa puede configurarlo de acuerdo a sus necesidades.

Adicional tiene funciones como correo de voz, música en espera, transferencia de llamadas, entre otras. Una ventaja de Asterisk es que tiene compatibilidad con un sin número de protocolos, pero en este caso se va a utilizar el protocolo SIP, ya que facilita la comunicación con distintos dispositivos celulares (**Figura 3.1**).



Figura 3.1. Logo de Asterisk [8]

Análisis de software libre Elastix

Elastix es una distribución de Asterisk, con la diferencia que su representación es con interfaz gráfica se usuario similar a “FreePBX”, sus características se asemejan con Asterisk, es por eso que facilita las configuraciones y gestiones de los sistemas de telefonía, ya que es representada gráficamente en una página web que brinda una breve configuración en las extensiones, troncales, reglas de enrutamiento, etc (**Figura 3.2**) [16].



Figura 3.2. Logo de Elastix [16]

Análisis de software libre FreePBX

Es un software que maneja interfaz gráfica mediante una página web como “Elastix”, se integra con Asterisk y su fácil configuración y administración la resalta de los demás sistemas de telefonía. Como se mencionó anteriormente, tiene un fácil manejo mediante su interfaz web, al momento de administrar las troncales y extensiones. Cuenta con un panel de control para poder monitorear el estado en el que se encuentra el sistema, al igual que módulos adicionales con la finalidad de ampliar otras funcionalidades que conlleva FreePBX (**Figura 3.3**) [17].



Figura 3.3. Logo de FreePBX [17]

Análisis de software libre Issabel

Es un software que se creó en el año 2016 con la finalidad de evitar las pérdidas de los avances que tenía Elastix durante años. es un software de telefonía IP de código abierto, al igual que otros Elastix se basa en Asterisk, ya que su función es montar servidores PBX IP, correo electrónico, mensajería instantánea, entre otros. Algunas de las utilizadas de este software libre es la grabación de llamadas, llamadas en espera, IVR flexible y configurables, herramientas para crear lotes de extensiones, etc (**Figura 3.4**) [18].



Figura 3.4. Logo de Issabel [18]

3.2 Seleccionar el hardware y software para el servidor y clientes acorde a los requerimientos establecidos

Posteriormente, se llevará a cabo la selección de los hardware y software, considerando todos los requisitos que se necesitarán para el despliegue de la central PBX, tanto para los clientes como para el software libre. El objetivo de este proyecto es garantizar la prestación de un central PBX óptima en conexiones de voz durante las llamadas planificadas.

Selección de hardware

De acorde al sistema PBX planteado, se necesita un hardware en donde se pueda levantar el servicio de telefonía, se tiene diferentes elementos para la creación del siguiente prototipo, desde un computador en donde se puede montar en una máquina virtual hasta en un dispositivo como un raspberry pi 4. Para la selección de los dispositivos se hizo un análisis para ver lo más factible para el levantamiento de la PBX.

Selección del raspberry pi

Se selecciono el raspberry pi 4 por las características que se van a presentar a continuación, ya que cumple con todos los parámetros para el desarrollo de una central telefónica privada. Por concepto general un raspberry pi 4 es denominada como una

mini computadora personal como se aprecia en la **Figura 3.5**. Contiene cuatro puertos USB, una entrada para colocar la tarjeta microSD, dos ranuras V8 para colocar un cable HDMI al monitor mediante un adaptador, una salida para audio, una entrada para cable Giga Ethernet y una entradas tipo C para poder alimentar del raspberry pi 4.

Hay que tener en cuenta que los componentes que conlleva el raspberry pi 4 varían según el modelo. Este dispositivo tiene un sin número de aplicaciones en donde se le utiliza, pero en esta ocasión lo vamos a usar como una central telefónica privada aprovechando todos sus elementos para poder implementarlo según los objetivos propuestos en el plan [11].

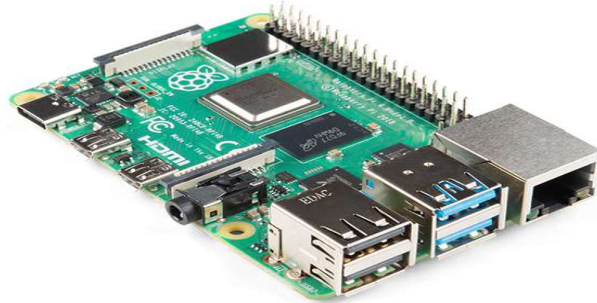


Figura 3.5. Raspberry pi 4 modelo B [11]

Las características que tiene el dispositivo:

- CPU de 1,5 (GHz).
- Conexión HDMI en dos pantallas con una calidad de has 4K
- Conexión WIFI
- Bluetooth 5.0
- Memoria de 8 (GB)

Estas son las características por las que vamos a utilizar un raspberry pi 4 ya que nos proporciona una gran velocidad y memoria para implementar sin problemas la central telefónica privada.

Selección de tarjeta microSD

Como se observa en la **Figura 3.6**. Se seleccionó una tarjeta microSD clase 10 de 64 (GB), ya que brinda más espacio para almacenar cualquier tipo de información, este elemento se utilizó para el booteo del sistema operativo Ubuntu en el cual se va a

desarrollar la implementación de la central telefónica, ya que tiene una velocidad de lectura de hasta 120 (MB/s) y una velocidad de escritura de 25 (MB/s) [19].



Figura 3.6. Tarjeta microSD clase 10 de 64GB [19]

Algunas aplicaciones de la tarjeta microSD:

- Almacenamiento extra para teléfonos celulares y tabletas.
- Se utiliza para bootear sistemas operativos de Linux, Windows, etc.
- Almacenar videos en alta definición.
- Almacenamiento de fotos de alta definición para cámaras digitales.
- Se utiliza en reproductores de música para almacenar una amplia biblioteca.

Selección del dispositivo para la comunicación de voz

Se seleccionó el dispositivo celular, ya que comúnmente se utiliza para la comunicación móvil entre diferentes usuarios. Este dispositivo inteligente se conecta de manera inalámbrica hacia las redes de internet, es por eso que pasa a formar parte de nuestra central telefónica, ya que se puede descargar un sin número de apps, es por ello que para la PBX se va a utilizar un softphone que se puede descargar en dicho dispositivo.

En este caso se va a descargar la aplicación llamada “Zoiper” como se está representando en la **Figura 3.7**, para poder conectarnos a la central, ya que para lo único que necesita es la extensión y la contraseña del usuario creado en Asterisk para poderse conectar a la central telefónica privada, es decir es un complemento más que necesita la central para poder tener comunicación de voz.

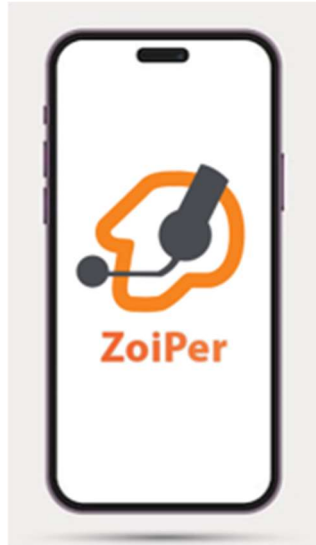


Figura 3.7. Smartphone con ZoiPer

Selección de Software

Acorde al plan de tesis, para el desarrollo de la PBX se necesita de un software libre en donde será montada la central telefónica privada con la finalidad de tener comunicación de voz entre distintos usuarios, es por eso que según los estudios realizados de los diferentes softwares que existen para poder levantar este tipo de servicio, se escogió Asterisk ya que es un software completo que contiene lo necesario para que el presente proyecto se lleve a cabo.

Selección de sistema Operativo Ubuntu

Se tomo en cuenta el siguiente sistema operativo, ya que nos ofrece estabilidad, esto es esencial para nuestra central telefónica privada debido a que debe funcionar de forma continua. Ubuntu es una plataforma de código abierto muy popular para implementaciones de centrales telefónicas. Este sistema operativo nos ofrece un entorno seguro para las gestiones de las comunicaciones telefónicas que son sensibles (**Figura 3.8**) [10].



Figura 3.8. Logo de Ubuntu [10]

Características de Ubuntu:

- Ofrece escalabilidad, estabilidad y confiabilidad a la central telefónica.
- Brinda soporte para la central telefónica.
- Es seguro para la implementación de una central telefónica.
- Es compatible con Asterisk.
- Es fácil para la instalación y configuración de Asterisk

Selección del software Asterisk

Para seleccionar el software Asterisk realizamos estudios de todos los parámetros que esta plataforma nos puede brindar, con la finalidad de que la central telefónica privada cuente con una excelente calidad, al momento de realizar las llamadas desde un usuario hacia otro, a continuación, se describirán algunos parámetros importantes de Asterisk:

- **Características y funcionalidades de Asterisk**

De acuerdo a las necesidades de la implementación se escogió el software Asterisk, ya que permite conexión y gestión de llamadas tanto internas como externas, dentro de Asterisk se puede manejar numerosas líneas telefónicas y extensiones.

Se maneja el protocolo SIP ya que es el más compatible con Asterisk a comparación de otros protocolos. Por otro lado, brinda el desarrollo de sistemas de respuesta de voz interactiva (IVR), para tener interacción con otros usuarios que están conectados a la red. Tiene servicios extras como grabaciones de llamadas ya que es una característica para la calidad del servicio [8].

- **Requisitos breves para el levantamiento de Asterisk**

Para la implementación de Asterisk se utilizó un raspberry pi 4 y una tarjeta micro SD de 64 (GB) ya que juntos llegan a velocidades desde 50 a 400 (Mbps) lo que es suficiente para nuestro sistema operativo, es por esto que al momento de instalar Asterisk para la central telefónica, no se presenta ningún error porque nos da una velocidad aceptable y a su vez un almacenamiento para descargar sin ningún problema los protocolos y archivos como el "sip.conf" y "extensions.conf", ya que vienen mejorados para la comunicación de voz, con la finalidad de no presentar errores como actualizaciones anteriores de Asterisk [8].

- **Actualizaciones y desarrollo continuo de Asterisk**

Las ventajas de las actualizaciones con el pasar del tiempo, es la mejora del rendimiento del software, ya que en cada actualización corrige errores que

presentaban versiones anteriores, mejora la seguridad para proteger al sistema de telefonía ya que se maneja a través de una red de internet. En cada versión actual agregan nuevas características para una configuración más accesible, mejorando la capacidad y la flexibilidad de la central telefónica privada [8].

Selección de Softphone a utilizarse

Se estudio los siguientes softphones, ya que son compatibles con la plataforma de Asterisk. Pero para nuestra central solo se va a escoger uno, comparando las características y viendo que softphone es más eficiente, previamente el softphone escogido es ZoiPer ya que tiene configuraciones más completas a comparación de los otros.

ZoiPer

Es un softphone compatible con diversas plataformas como Android, Linux, Windows, etc. Nos brinda una interfaz muy interactiva que nos facilita la ejecución de llamadas de VoIP. Esta aplicación es conocida por su fácil configuración, esto permite que los usuarios puedan conectarse de forma inmediata a los servicios de telefonía que este brinda y que son basadas en Asterisk. También dispone de diversas funciones avanzadas como mensajería instantánea, transferencia de llamadas y conferencias (**Figura 3.9**) [15].



Figura 3.9. Logo de ZoiPer [15]

Características de ZoiPer:

- Compatible con diferentes plataformas.
- Brinda una interfaz intuitiva.
- Tiene compatibilidad con la plataforma SIP.
- Buena calidad de audio.
- Es segura.
- Su configuración es rápida.

3CX

Es una aplicación que está basada en software para las comunicaciones unificadas. Al igual que ZoiPer tiene compatibilidad con diferentes plataformas como Linux, Windows. La nube, etc. Esta aplicación brinda funciones avanzadas sobre video conferencia y telefonía IP.

3CX tiene una interfaz gráfica muy fácil de usar ya que permite una administración muy eficiente de su sistema. Ofrece una gran escalabilidad, adaptándose a todas las necesidades mínimas de grandes corporativos y empresas. El modelo de licencia que es basado en extensiones, tiende a reducir costos y simplificar las gestiones (**Figura 3.10**) [20].



Figura 3.10. Logo de 3CX [20]

Características de 3CX:

- Compatible con pocas plataformas
- Brinda comunicación unificada
- Brinda interfaz intuitiva
- Es segura
- Tiene escalabilidad

MicroSIP

Es una plataforma que está basada para Windows y Linux, es de código abierto y ofrece soluciones ligeras, por lo que es fácil utilizarla para las comunicaciones de VoIP. Se distingue de las demás, ya que la interfaz que maneja esta aplicación es sencilla y rápida de configurar, es por ello que los que más le dan uso a MicroSIP son personas principiantes.

Es compatible con el protocolo SIP, por lo que se le hace fácil integrarse a los dispositivos de telefonía que están basados en ese estándar. Ofrece diversas funciones como transferencia de archivos, mensajería instantánea y llamadas de voz. A pesar de que no es una aplicación muy eficiente, MicroSIP brinda una buena calidad de audio durante las llamadas (**Figura 3.11**) [20].



Figura 3.11. Logo de MicroSIP [20]

Características de MicroSIP:

- Cuenta con una interfaz sencilla
- Tiene una configuración rápida
- Compatible con el protocolo SIP
- Calidad de audio eficiente.

3.3 Configurar el servidor PBX basado software libre

Una vez seleccionado el hardware, se pasa a la instalación y configuración para levantar el servicio de la central telefónica privada (PBX). A continuación, se van a detallar cada paso realizado en el sistema operativo Ubuntu para instalar y configurar la central.

Booteo del sistema operativo para el raspberry pi 4

Una vez descargado el programa “Raspberry Pi Imager”, se procede a realizar el booteo del sistema operativo que vamos a utilizar para poder instalar nuestro software libre, en este caso vamos a utilizar Ubuntu (ver **Figura 3.12**).



Figura 3.12. Pantalla principal de Raspberry Pi Imager

Se empieza seleccionando el modelo del dispositivo Raspberry Pi en el que se va a correr el sistema operativo Ubuntu en este caso el dispositivo que se va a utilizar es un Raspberry pi 4 modelo B (**Figura 3.13**).

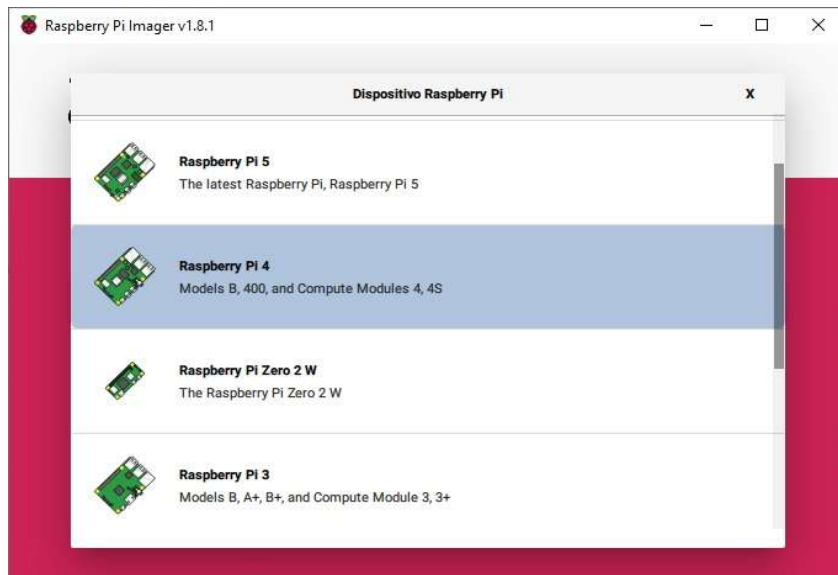


Figura 3.13. Selección del dispositivo

Luego se selecciona el sistema operativo con el que se va a trabajar, después se dirige al apartado “elegir SO”, se desplegará una pantalla en donde se muestra diferentes sistemas operativos que se pueden utilizar, en este caso se escoge la opción “Other general-purpose OS”, como se ve en la **Figura 3.14**.

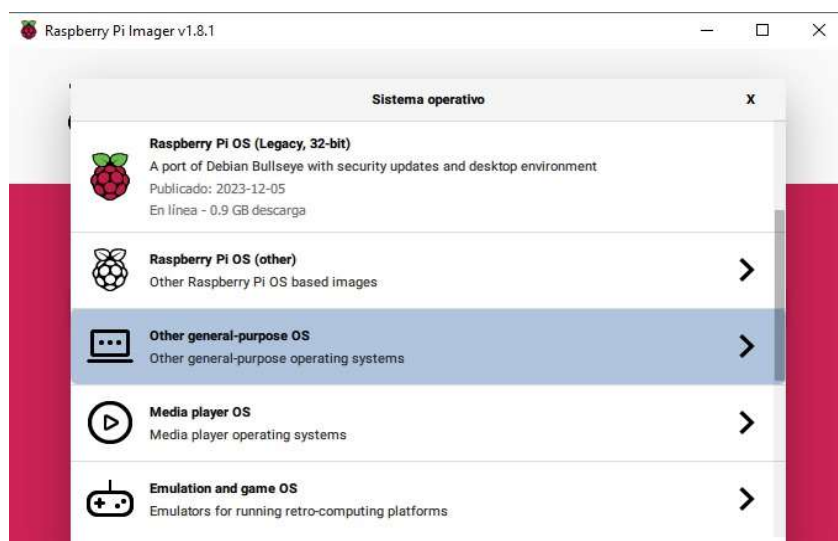


Figura 3.14. Sección de los diferentes sistemas operativos

Una vez adentro en el apartado mencionado anteriormente y se desplegará otra ventana en donde se va a encontrar el icono de Ubuntu como se ve en la **Figura 3.15**. Es decir, el sistema operativo en donde se va a desarrollar la instalación de Asterisk.

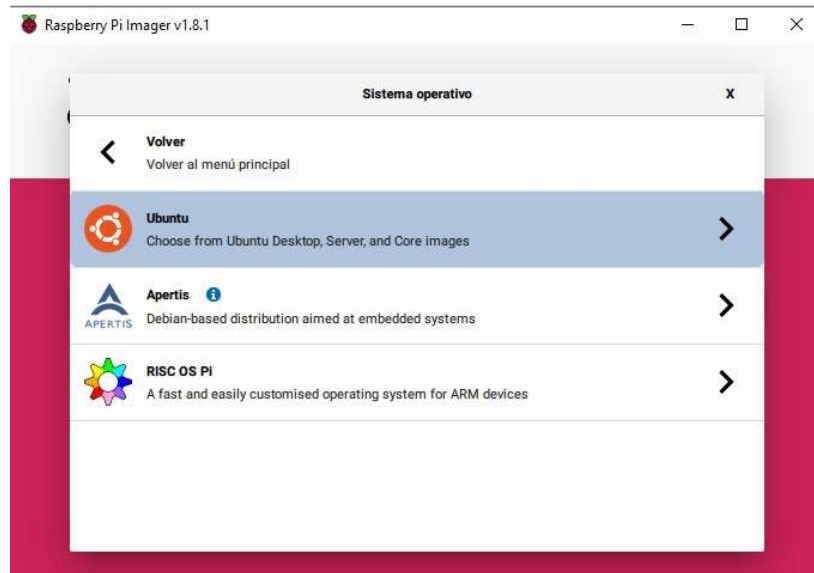


Figura 3.15. Selección del sistema operativo Ubuntu

Se entra en el apartado "Ubuntu" y se escoge la versión de Ubuntu que vamos a utilizar, hay que tener en cuenta que las versiones más recientes son mejores, ya que van a presentar menos errores que versiones anteriores de tal manera que al utilizar una versión actual, da una mejor calidad de servicio a nuestra central telefónica privada. En este caso se escoge la versión que se muestra en la **Figura 3.16**.

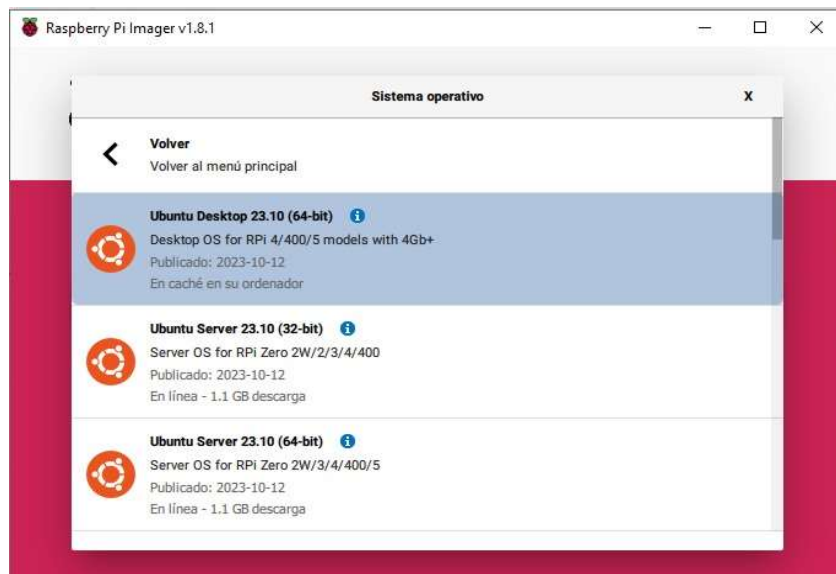


Figura 3.16. Selección de la versión de Ubuntu

A continuación, se selecciona el almacenamiento en el cual se va a realizar el booteo, en esta sección hay que tener en cuenta que la tarjeta microSD que vayan a usar debe ser mucho mayor a 8 (GB) de memoria, ya que este sistema operativo pesa 8.8 (GB), es por esto que para este caso se optó por montar el sistema operativo en una tarjeta microSD de 64GB, como muestra en la **Figura 3.17**.

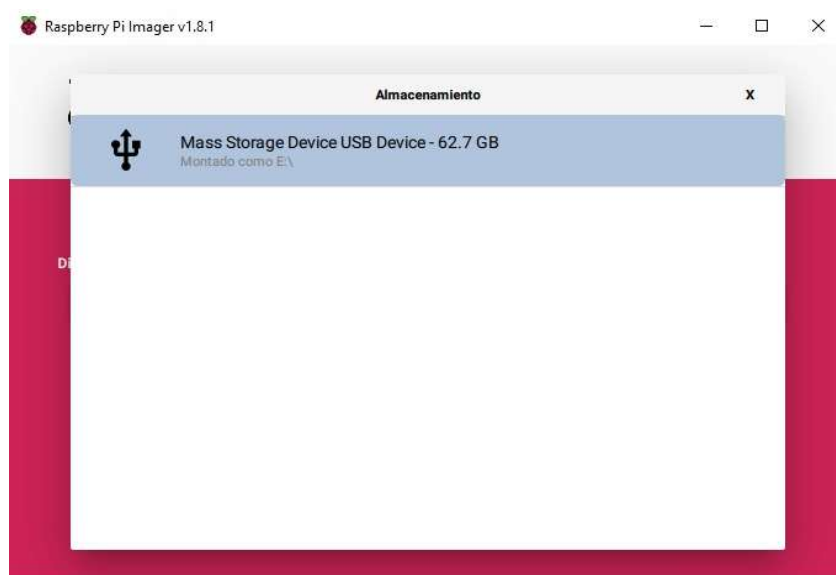


Figura 3.17. Selección de almacenamiento

Luego le damos en “Siguiente”, se mostrará una ventana en donde nos menciona que se borran todos los datos existentes de la tarjeta de “Mass Storage Device USB Device”, presionamos “SI” (ver **Figura 3.18**).



Figura 3.18. Ventana de advertencia para iniciar el booteo

Finalmente, empezara a realizarse el booteo de Ubuntu en la tarjeta microSD (**Figura 3.19**).



Figura 3.19. Inicio del booteo del sistema operativo

Configuración de dirección estática

Una vez ya booteado el sistema operativo se procede a arrancarlo en el raspberry pi 4, para poder instalar y configurar todo lo que se necesite para que Ubuntu funcione de manera correcta. Luego se dirige al CLI, pero antes de empezar con las configuraciones, se tiene que dirigir al apartado de ajustes del sistema operativo, dar click en la red que se está conectada y se nos desplegara la ventana como se observa en la **Figura 3.20**, nos dirigimos a la parte de IPv4 y cambiamos a una dirección estática ya que por defecto el sistema operativo viene con direccionamiento dinámico (la dirección IP estática se

debe colocar según el rango que se muestre en la red de internet a la que está conectada).

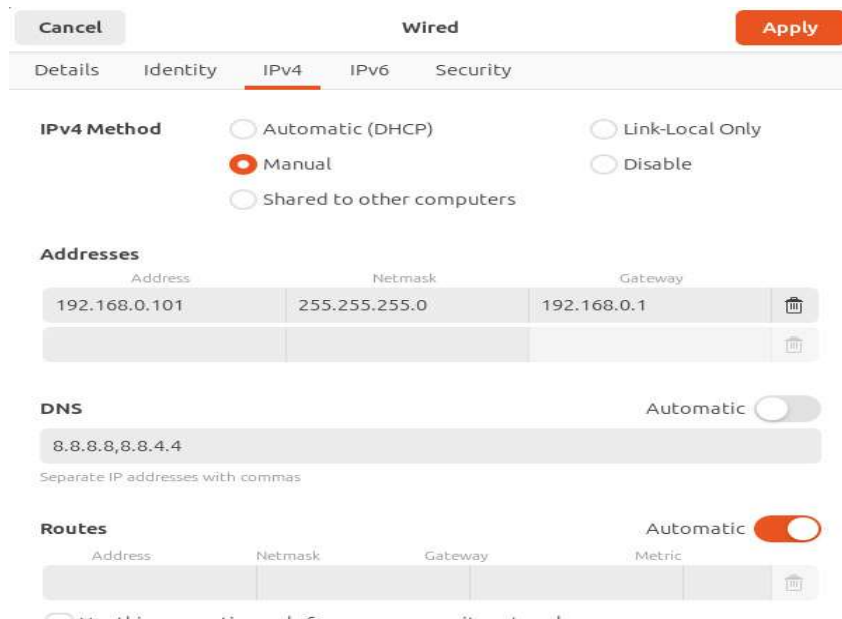


Figura 3.20. Configuración de dirección estática

Acceso al CLI de Ubuntu para la instalación de Asterisk

Dentro del CLI se debe actualizar todos los paquetes antes de descargar el software libre Asterisk, ya que esto ayudara a que se descarguen los archivos actualizados del software libre, con la finalidad de que la central telefónica sea óptima y eficiente con la versión de Ubuntu instalada, es por eso que se utiliza los comandos: “sudo apt-get update” y el “sudo apt-get upgrade”, como se observan en la **Figura 3.21** y la **Figura 3.22**.

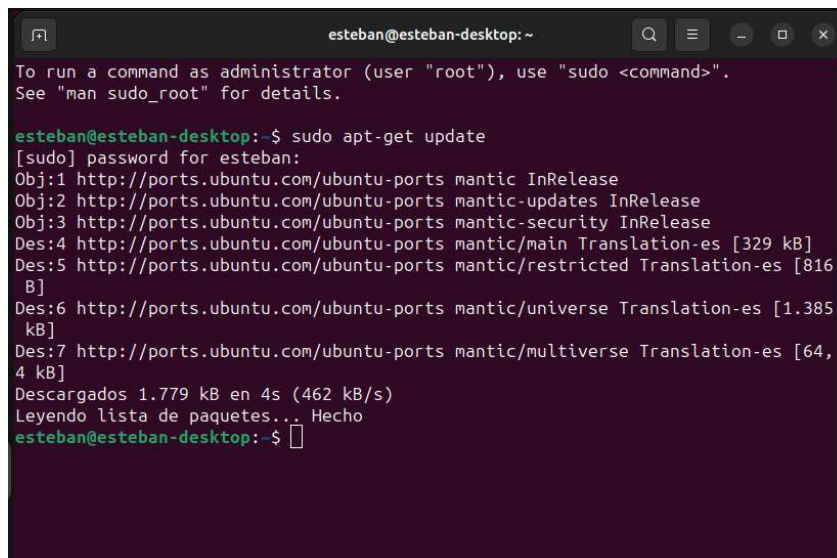


Figura 3.21. Actualización del update


```
esteban@esteban-desktop: ~  
Taking backup of pifi-40.dtbo.  
Installing new pifi-40.dtbo.  
Taking backup of midi-uart0-pi5.dtbo.  
Installing new midi-uart0-pi5.dtbo.  
Taking backup of iqaudio-dacplus.dtbo.  
Installing new iqaudio-dacplus.dtbo.  
Taking backup of i2c5.dtbo.  
Installing new i2c5.dtbo.  
Taking backup of cm-swap-i2c0.dtbo.  
Installing new cm-swap-i2c0.dtbo.  
Taking backup of ads1015.dtbo.  
Installing new ads1015.dtbo.  
Taking backup of README.  
Installing new README.  
Processing triggers for libc-bin (2.38-1ubuntu6) ...  
esteban@esteban-desktop:~$ sudo apt-get upgrade  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
The following packages have been kept back:  
  linux-image-raspi ubuntu-advantage-tools  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
esteban@esteban-desktop:~$
```

Figura 3.22. Actualización del upgrade

En el caso de no tener un monitor, se puede descargar y activar el servicio SSH como se visualiza en la **Figura 3.23**, ya que nos ofrece una conexión segura mediante conexión remota. Para poder llevar a cabo la conexión es necesario tener instalado el “Putty” para poder acceder al CLI mediante la IP estática propuesta.

```
esteban@esteban-desktop: ~  
esteban@esteban-desktop:~$ sudo apt install openssh-server  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  ncurses-term openssh-sftp-server ssh-import-id  
Suggested packages:  
  molly-guard monkeysphere ssh-askpass  
The following NEW packages will be installed:  
  ncurses-term openssh-server openssh-sftp-server ssh-import-id  
0 upgraded, 4 newly installed, 0 to remove and 2 not upgraded.  
Need to get 737 kB of archives.  
After this operation, 6229 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://ports.ubuntu.com/ubuntu-ports mantic/main arm64 openssh-sftp-server arm64 1:9.3p1-1ubuntu3 [36.5 kB]  
Get:2 http://ports.ubuntu.com/ubuntu-ports mantic/main arm64 openssh-server arm64 1:9.3p1-1ubuntu3 [418 kB]  
Get:3 http://ports.ubuntu.com/ubuntu-ports mantic/main arm64 ncurses-term all 6.4+20230625-2 [272 kB]  
Get:4 http://ports.ubuntu.com/ubuntu-ports mantic/main arm64 ssh-import-id all 5.11-0ubuntu1 [10.1 kB]  
Fetched 737 kB in 2s (319 kB/s)  
Preconfiguring packages ...
```

Figura 3.23. Instalación del SSH

Mediante los comandos que se evidencian en la **Figura 3.24**, se puede activar y ver si el servicio SSH ya está levantado para poder tener una conexión remota, ya que como lo mencione anteriormente, esto se hace solo en caso de que no se tenga un monitor para poder conectar al Raspberry pi.

```
esteban@esteban-desktop: ~  
esteban@esteban-desktop:~$ sudo systemctl start ssh  
esteban@esteban-desktop:~$ sudo systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: enabled)  
   Drop-In: /etc/systemd/system/ssh.service.d  
           └─00-socket.conf  
   Active: active (running) since Sun 2023-11-26 16:48:23 -05; 2s ago  
   TriggeredBy: ● ssh.socket  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
   Process: 11071 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
   Main PID: 11072 (sshd)  
     Tasks: 1 (limit: 3903)  
    Memory: 1.3M  
       CPU: 89ms  
   CGroup: /system.slice/ssh.service  
           └─11072 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Nov 26 16:48:23 esteban-desktop systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...  
Nov 26 16:48:23 esteban-desktop sshd[11072]: Server listening on :: port 22.  
Nov 26 16:48:23 esteban-desktop systemd[1]: Started ssh.service - OpenBSD Secure Shell server.  
esteban@esteban-desktop:~$ sudo systemctl enable ssh  
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable ssh
```

Figura 3.24. Levantamiento del servicio SSH

A continuación, se procede a instalar el software Asterisk en el que se va a levantar la Central telefónica privada (PBX). Para realizar las interconexiones entre los teléfonos inteligentes y poder tener servicio de voz entre los usuarios, tal como se evidencia en la **Figura 3.25**.

```
esteban@esteban-desktop: ~  
esteban@esteban-desktop:~$ sudo apt-get install asterisk  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  asterisk-config asterisk-core-sounds-en asterisk-core-sounds-en-gsm asterisk-modules  
  asterisk-moh-opsound-gsm freetds-common libc-client2007e libcodec2-1.2 libgmime-3.0-0 libgsm1  
  libiksemel3 liblua5.1-0 libneon27 libodbc2 libopencore-amrnb0 libopencore-amrwb0 libopusfile0  
  libportaudio2 libpq5 libradcli4 libresample1 libsox-fmt-alsa libsox-fmt-base libsox3 libspandsp2  
  libspeexdsp1 libsrtp2-1 libsybdb5 libunbound8 liburiparser1 libvo-amrwbenc0 mlock sox  
Suggested packages:  
  asterisk-dahdi asterisk-dev asterisk-doc asterisk-oooh323 asterisk-opus uw-mailutils  
  odbc-postgresql tdsodbc libsox-fmt-all  
The following NEW packages will be installed:  
  asterisk asterisk-config asterisk-core-sounds-en asterisk-core-sounds-en-gsm asterisk-modules  
  asterisk-moh-opsound-gsm freetds-common libc-client2007e libcodec2-1.2 libgmime-3.0-0 libgsm1  
  libiksemel3 liblua5.1-0 libneon27 libodbc2 libopencore-amrnb0 libopencore-amrwb0 libopusfile0  
  libportaudio2 libpq5 libradcli4 libresample1 libsox-fmt-alsa libsox-fmt-base libsox3 libspandsp2  
  libspeexdsp1 libsrtp2-1 libsybdb5 libunbound8 liburiparser1 libvo-amrwbenc0 mlock sox  
0 upgraded, 34 newly installed, 0 to remove and 2 not upgraded.  
Need to get 22.3 MB of archives.  
After this operation, 70.1 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://ports.ubuntu.com/ubuntu-ports mantic/universe arm64 asterisk-config all 1:20.4.0~dfsg~
```

Figura 3.25. Instalación de Asterisk

Se verifica la versión de Asterisk que se descargó en el sistema operativo Ubuntu, para así empezar a instalar los paquetes necesarios para levantar la central telefónica privada (ver **Figura 3.26**).

```
esteban@esteban-desktop: ~  
$ asterisk -V  
Asterisk 20.4.0-dfsg+-cs6.13.40431414-2  
esteban@esteban-desktop: ~
```

Figura 3.26. Versión de Asterisk

Como se presenta en la **Figura 3.27**, el comando “apt-cache search asterisk”, se puede observar todos los paquetes de códecs que Asterisk ofrece. El idioma que trae por defecto en los códecs de respuesta de las llamadas es en inglés, es por eso que se instalan los paquetes en español.

```
esteban@esteban-desktop: ~  
$ apt-cache search asterisk  
debian-goodies - Small toolbox-style utilities for Debian systems  
asterisk - Open Source Private Branch Exchange (PBX)  
asterisk-config - Configuration files for Asterisk  
asterisk-core-sounds-en - asterisk PBX sound files - US English  
asterisk-core-sounds-en-g722 - asterisk PBX sound files - en-us/g722  
asterisk-core-sounds-en-gsm - asterisk PBX sound files - en-us/gsm  
asterisk-core-sounds-en-wav - asterisk PBX sound files - en-us/wav  
asterisk-core-sounds-es - asterisk PBX sound files - Spanish  
asterisk-core-sounds-es-g722 - asterisk PBX sound files - es-mx/g722  
asterisk-core-sounds-es-gsm - asterisk PBX sound files - es-mx/gsm  
asterisk-core-sounds-es-wav - asterisk PBX sound files - es-mx/wav  
asterisk-core-sounds-fr - asterisk PBX sound files - Canadian French  
asterisk-core-sounds-fr-g722 - asterisk PBX sound files - fr-ca/g722  
asterisk-core-sounds-fr-gsm - asterisk PBX sound files - fr-ca/gsm  
asterisk-core-sounds-fr-wav - asterisk PBX sound files - fr-ca/wav  
asterisk-core-sounds-it - asterisk PBX sound files - Italian  
asterisk-core-sounds-it-g722 - asterisk PBX sound files - it-it/g722  
asterisk-core-sounds-it-gsm - asterisk PBX sound files - it-it/gsm  
asterisk-core-sounds-it-wav - asterisk PBX sound files - it-it/wav  
asterisk-core-sounds-ru - asterisk PBX sound files - Russian  
asterisk-core-sounds-ru-g722 - asterisk PBX sound files - ru-ru/g722  
asterisk-core-sounds-ru-gsm - asterisk PBX sound files - ru-ru/gsm  
asterisk-core-sounds-ru-wav - asterisk PBX sound files - ru-ru/wav
```

Figura 3.27. Paquetes de Asterisk

Se procede a la descarga de los códecs en español, para así poder comprimir y descomprimir los datos de audio, con la finalidad de reducir el tamaño de los archivos para facilitar el almacenamiento, la transmisión y ahorrar ancho de banda. Esto es el punto esencial para la transmisión dentro de la PBX, al igual que el “prompt”, ya que dentro de la PBX se los utiliza como sistemas de respuesta de voz interactiva (IVR), (ver **Figura 3.28**).

```
esteban@esteban-desktop:~$ sudo apt-get install asterisk-prompt-es asterisk-core-sounds-es asterisk-core-sounds-es-gsm asterisk-core-sounds-es-wav asterisk-core-sounds-es-g722
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'asterisk-core-sounds-es' instead of 'asterisk-prompt-es'
The following NEW packages will be installed:
  asterisk-core-sounds-es asterisk-core-sounds-es-g722 asterisk-core-sounds-es-gsm
  asterisk-core-sounds-es-wav
0 upgraded, 4 newly installed, 0 to remove and 2 not upgraded.
Need to get 39.8 MB of archives.
After this operation, 48.7 MB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports mantic/universe arm64 asterisk-core-sounds-es-gsm all 1.6.1-1 [3005 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports mantic/universe arm64 asterisk-core-sounds-es-g722 all 1.6.1-1 [13.0 MB]
Get:3 http://ports.ubuntu.com/ubuntu-ports mantic/universe arm64 asterisk-core-sounds-es-wav all 1.6.1-1 [23.8 MB]
Get:4 http://ports.ubuntu.com/ubuntu-ports mantic/universe arm64 asterisk-core-sounds-es all 1.6.1-1 [19.8 kB]
Fetched 39.8 MB in 8s (4945 kB/s)
Selecting previously unselected package asterisk-core-sounds-es-gsm.
(Reading database ... 125717 files and directories currently installed.)
Preparing to unpack .../asterisk-core-sounds-es-gsm_1.6.1-1_all.deb ...
```

Figura 3.28. Instalación de paquetes de Asterisk

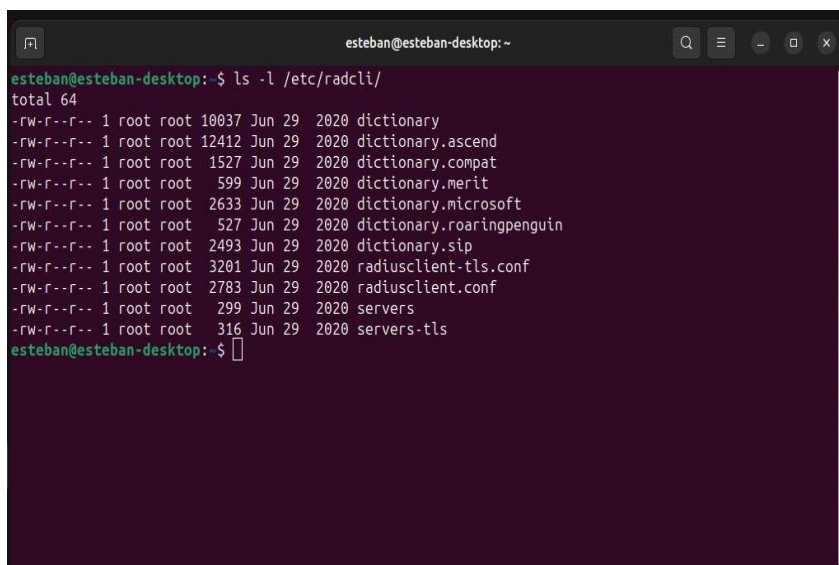
Una vez descargado los códecs en español se verifica que el sistema de Asterisk esté funcionando de manera correcta, pero como se puede observar en la **Figura 3.29**, existe un error del “radcli”, ya que está carpeta está en una dirección que el sistema no puede reconocer, es decir no se encuentra dentro lo de los archivos de Asterisk, por lo que se procede a corregir ese error.

```
esteban@esteban-desktop:~$ service asterisk status
● asterisk.service - Asterisk PBX
   Loaded: loaded (/lib/systemd/system/asterisk.service; enabled; preset: enabled)
   Active: active (running) since Sun 2023-11-26 17:03:15 -05; 15min ago
     Docs: man:asterisk(8)
  Main PID: 13514 (asterisk)
    Tasks: 69 (limit: 3903)
   Memory: 47.2M
      CPU: 20.057s
   CGroup: /system.slice/asterisk.service
           └─13514 /usr/sbin/asterisk -g -f -p -U asterisk
             └─13515 astcanary /var/run/asterisk/alt.asterisk.canary.tweet.tweet.tweet 13514

Nov 26 17:03:12 esteban-desktop systemd[1]: Starting asterisk.service - Asterisk PBX...
Nov 26 17:03:14 esteban-desktop asterisk[13514]: radcli: rc_read_config: rc_read_config: can't open>
Nov 26 17:03:14 esteban-desktop asterisk[13514]: radcli: rc_read_config: rc_read_config: can't open>
Nov 26 17:03:15 esteban-desktop systemd[1]: Started asterisk.service - Asterisk PBX.
líneas 1-16/16 (END)
```

Figura 3.29. Verificación del servicio Asterisk

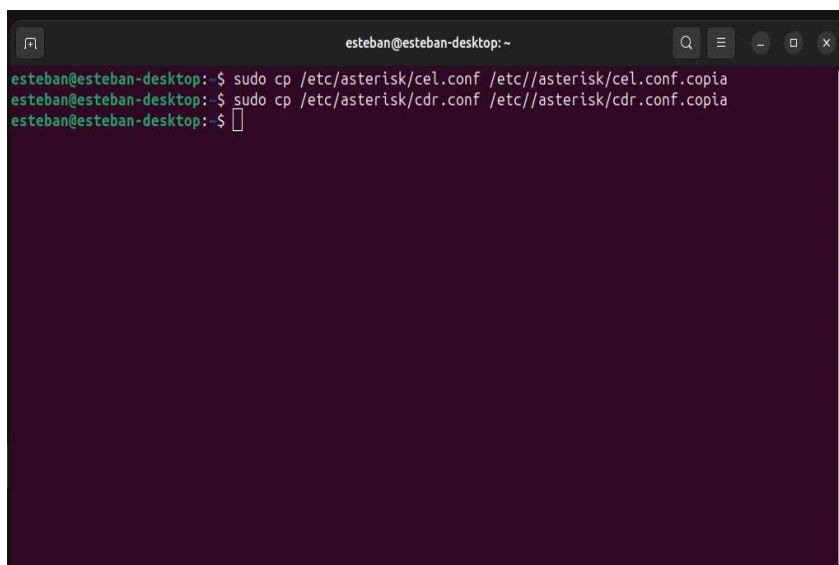
Se debe buscar el archivo que está dando problemas y se puede notar que efectivamente se encuentra en otra dirección, es decir no está con los archivos de Asterisk como se observar en la siguiente **Figura 3.30**, lo que se procede hacer es ubicar de manera correcta el archivo para que así Asterisk lo reconozca y funcione de manera correcta.



```
esteban@esteban-desktop: ~  
esteban@esteban-desktop:~$ ls -l /etc/radcli/  
total 64  
-rw-r--r-- 1 root root 10037 Jun 29 2020 dictionary  
-rw-r--r-- 1 root root 12412 Jun 29 2020 dictionary.ascend  
-rw-r--r-- 1 root root 1527 Jun 29 2020 dictionary.compat  
-rw-r--r-- 1 root root 599 Jun 29 2020 dictionary.merit  
-rw-r--r-- 1 root root 2633 Jun 29 2020 dictionary.microsoft  
-rw-r--r-- 1 root root 527 Jun 29 2020 dictionary.roaringpenguin  
-rw-r--r-- 1 root root 2493 Jun 29 2020 dictionary.sip  
-rw-r--r-- 1 root root 3201 Jun 29 2020 radiusclient-tls.conf  
-rw-r--r-- 1 root root 2783 Jun 29 2020 radiusclient.conf  
-rw-r--r-- 1 root root 299 Jun 29 2020 servers  
-rw-r--r-- 1 root root 316 Jun 29 2020 servers-tls  
esteban@esteban-desktop:~$
```

Figura 3.30. Dirección de la carpeta radcli

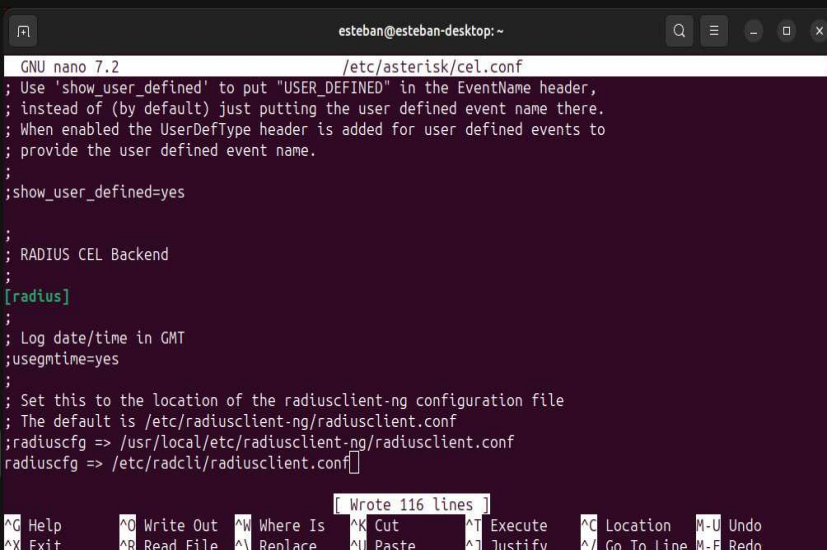
Para poder corregir este problema, se dirige a los archivos que contiene el “radcli”, los cuales son el archivo “cel.conf” y el archivo “cdr.conf” para poder cambiar su dirección, pero antes de eso es importante hacer una copia de cada uno para no alterar las configuraciones del archivo principal como se presenta en la **Figura 3.31**.



```
esteban@esteban-desktop: ~  
esteban@esteban-desktop:~$ sudo cp /etc/asterisk/cel.conf /etc//asterisk/cel.conf.copia  
esteban@esteban-desktop:~$ sudo cp /etc/asterisk/cdr.conf /etc//asterisk/cdr.conf.copia  
esteban@esteban-desktop:~$
```

Figura 3.31. Dirección de los archivos radcli

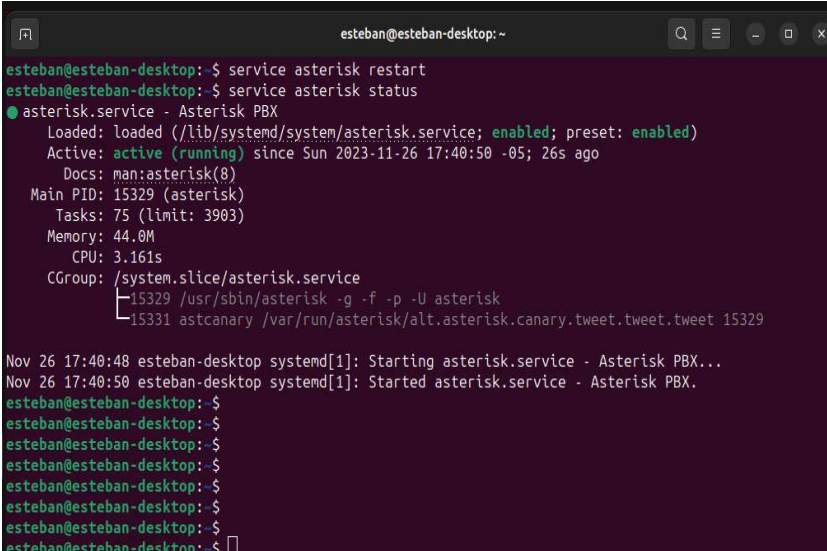
De acuerdo a lo ilustrado en la **Figura 3.32**, se ingresa a cada uno de los archivos para colocar la dirección correcta en donde se encuentran los otros archivos de Asterisk, una vez corregida la dirección se procede a guardar y salir del directorio.



```
GNU nano 7.2 /etc/asterisk/cel.conf
; Use 'show_user_defined' to put "USER_DEFINED" in the EventName header,
; instead of (by default) just putting the user defined event name here.
; When enabled the UserDefType header is added for user defined events to
; provide the user defined event name.
;
;show_user_defined=yes
;
; RADIUS CEL Backend
;
[radius]
;
; Log date/time in GMT
;usegmt=yes
;
; Set this to the location of the radiusclient-ng configuration file
; The default is /etc/radiusclient-ng/radiusclient.conf
;radiuscfg => /usr/local/etc/radiusclient-ng/radiusclient.conf
radiuscfg => /etc/radcli/radiusclient.conf
Wrote 116 lines
^G Help      ^O Write Out ^M Where Is  ^K Cut      ^T Execute  ^C Location M-U Undo
^X Exit      ^R Read File ^N Replace  ^U Paste    ^J Justify  ^_ Go To Line M-E Redo
```

Figura 3.32. Ingreso a los archivos radcli

Una vez corregido la dirección de los archivos del “radcli”, se verifica nuevamente si el servicio no presenta ningún error y esta levantado correctamente, pero antes de eso se realiza un “restart” para que el servicio arranque nuevamente desde cero y cómo se puede observar en la **Figura 3.33** el servicio de Asterisk esta levantado de manera correcta.



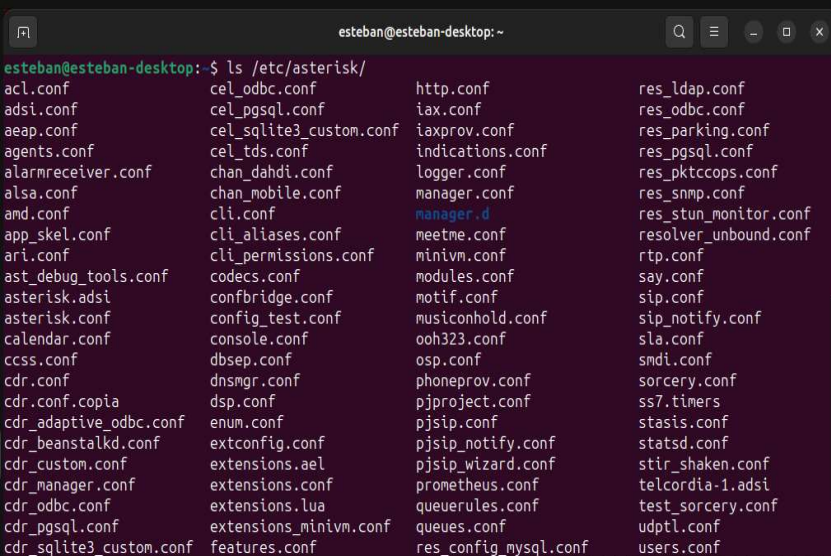
```
esteban@esteban-desktop:~$ service asterisk restart
esteban@esteban-desktop:~$ service asterisk status
● asterisk.service - Asterisk PBX
   Loaded: loaded (/lib/systemd/system/asterisk.service; enabled; preset: enabled)
   Active: active (running) since Sun 2023-11-26 17:40:50 -05; 26s ago
     Docs: man:asterisk(8)
   Main PID: 15329 (asterisk)
    Tasks: 75 (limit: 3903)
   Memory: 44.0M
      CPU: 3.161s
   CGroup: /system.slice/asterisk.service
           └─15329 /usr/sbin/asterisk -g -f -p -U asterisk
             └─15331 astcanary /var/run/asterisk/alt.asterisk.canary.tweet.tweet.tweet 15329

Nov 26 17:40:48 esteban-desktop systemd[1]: Starting asterisk.service - Asterisk PBX...
Nov 26 17:40:50 esteban-desktop systemd[1]: Started asterisk.service - Asterisk PBX.
esteban@esteban-desktop:~$
esteban@esteban-desktop:~$
esteban@esteban-desktop:~$
esteban@esteban-desktop:~$
esteban@esteban-desktop:~$
esteban@esteban-desktop:~$
esteban@esteban-desktop:~$
```

Figura 3.33. Verificación del servicio Asterisk

Una vez ya levantado el servicio correctamente se procede a ver los directorios creados en Asterisk, se verifica los más importantes que son el “sip.conf” y “extensions.conf”, ya

que son los directorios esenciales en donde se va a crear nuestro plan de marcado para poder conectar a los usuarios y poder tener comunicación de voz (**Figura 3.34**).

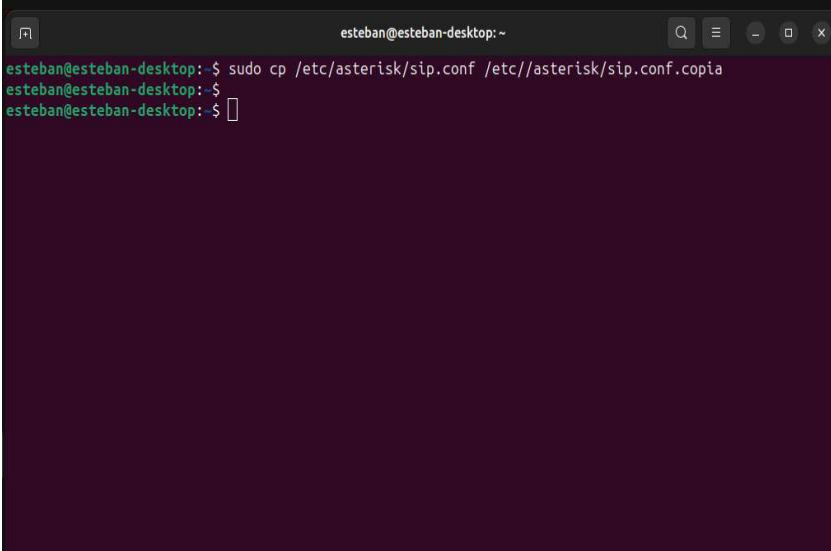


```
esteban@esteban-desktop: ~  
$ ls /etc/asterisk/  
acl.conf          cel_ldap.conf      http.conf          res_ldap.conf  
adsi.conf         cel_odbc.conf      iax.conf           res_odbc.conf  
aeap.conf         cel_pgsql.conf     iaxprov.conf      res_parking.conf  
agents.conf       cel_sqlite3_custom.conf iaxproto.conf     res_pgsql.conf  
alarmreceiver.conf chan_dahdi.conf    logger.conf        res_pktccops.conf  
alsa.conf         chan_mobile.conf   manager.conf       res_smp.conf  
amd.conf          cli.conf           manager.d           res_stun_monitor.conf  
app_skel.conf     cli_aliases.conf  meetme.conf        resolver_unbound.conf  
ari.conf          cli_permissions.conf minivm.conf         rtp.conf  
ast_debug_tools.conf codecs.conf         modules.conf        say.conf  
asterisk.adsi     confbridge.conf   motif.conf          sip.conf  
asterisk.conf     config_test.conf  musiconhold.conf   sip_notify.conf  
calendar.conf     console.conf       ooh323.conf         sla.conf  
ccss.conf         dbsep.conf         osp.conf            smdi.conf  
cdr.conf          dnsmgr.conf        phoneprov.conf      sorcery.conf  
cdr.conf.copia    dsp.conf           pjproject.conf      ss7_timers  
cdr_adaptive_odbc.conf enum.conf           pjsip.conf          stasis.conf  
cdr_banstalkd.conf extconfig.conf     pjsip_notify.conf  statsd.conf  
cdr_custom.conf   extensions.ael     pjsip_wizard.conf  stir_shaken.conf  
cdr_manager.conf  extensions.conf    prometheus.conf     telcordia-1.adsi  
cdr_odbc.conf      extensions.lua     queuerules.conf     test_sorcery.conf  
cdr_pgsql.conf     extensions_minivm.conf queues.conf          udptl.conf  
cdr_sqlite3_custom.conf features.conf       res_config_mysql.conf users.conf
```

Figura 3.34. Servicio de Asterisk

Configuración de Asterisk para la central telefónica privada

En este apartado como se observa en la **Figura 3.35**, se configura en directorio “sip.conf”, pero antes de esto se realiza una copia para no afectar al archivo inicial como ya se lo realizo anteriormente. Aquí se registra los terminales y también los usuarios que se podrán conectar a la central telefónica.

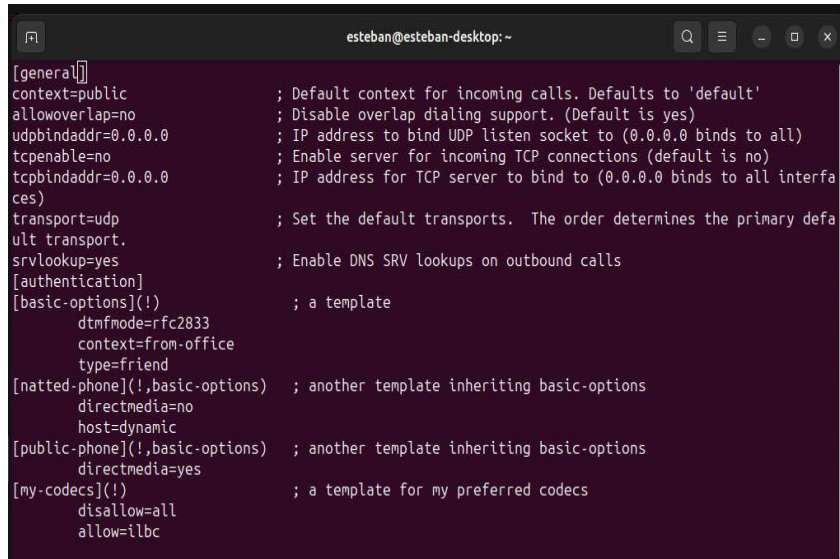


```
esteban@esteban-desktop: ~  
$ sudo cp /etc/asterisk/sip.conf /etc//asterisk/sip.conf.copia  
esteban@esteban-desktop: ~  
$  
esteban@esteban-desktop: ~
```

Figura 3.35. Dirección del directorio sip.conf

Una vez creado una copia de nuestro “sip.conf”, se ingresa con el comando “sudo vi /etc/asterisk/sip.conf” ya que permite eliminar las líneas comentadas y vacías, para que no se dificulte al momento de configurar nuestro directorio, ya que el texto que contiene

es demasiado grande e incómoda al momento de buscar la sección que debemos configurar, después de haber hecho los respectivos cambios se guarda y sale (ver **Figura 3.36**).



```
esteban@esteban-desktop: ~
[general]
context=public           ; Default context for incoming calls. Defaults to 'default'
allowoverlap=no         ; Disable overlap dialing support. (Default is yes)
udpbindaddr=0.0.0.0     ; IP address to bind UDP listen socket to (0.0.0.0 binds to all)
tcpenable=no           ; Enable server for incoming TCP connections (default is no)
tcpbindaddr=0.0.0.0     ; IP address for TCP server to bind to (0.0.0.0 binds to all interfaces)
transport=udp          ; Set the default transports. The order determines the primary default transport.
srvlookup=yes          ; Enable DNS SRV lookups on outbound calls
[authentication]
[basic-options](!)     ; a template
    dtmfmode=rfc2833
    context=from-office
    type=friend
[natted-phone](!,basic-options) ; another template inheriting basic-options
    directmedia=no
    host=dynamic
[public-phone](!,basic-options) ; another template inheriting basic-options
    directmedia=yes
[my-codecs](!)         ; a template for my preferred codecs
    disallow=all
    allow=ilbc
```

Figura 3.36. Directorio vi de sip.conf

Luego de haber comprimido la cantidad de palabras en el directorio se ingresa con el comando “nano /etc/asterisk/sip.conf” en nuestro directorio “sip.conf” y se agregan las siguientes líneas, como se ve en la **Figura 3.37**:

- **qualify=yes**; la función de “qualify” es que va a monitorear las conexiones entre los teléfonos VoIP.
- **language=es**; el “language” es el idioma que tendrán por defecto los usuarios dentro de la PBX.
- **disallow=all**; el “disallow” es para desactivar todos los codificadores, para poder activar los codificadores.
- **allow=alaw, ulaw**; el “allow” permite que los codificadores se ordenen de manera preferencial (ulaw es para Sudamérica).


```
GNU nano 7.2 /etc/asterisk/sip.conf *
[general]
context=public ; Default context for incoming calls. Defaults to 'default'
allowoverlap=no ; Disable overlap dialing support. (Default is yes)
udpbindaddr=0.0.0.0 ; IP address to bind UDP listen socket to (0.0.0.0 binds to all)
tcpenable=no ; Enable server for incoming TCP connections (default is no)
tcpbindaddr=0.0.0.0 ; IP address for TCP server to bind to (0.0.0.0 binds to all interf>
transport=udp ; Set the default transports. The order determines the primary def>
srvlookup=yes ; Enable DNS SRV lookups on outbound calls

qualify=yes
language=es
disallow=all
allow=alaw, ulaw

[authentication]
[basic-options](!) ; a template
    dtmfmode=rfc2833
    context=from-office
    type=friend
[natted-phone](!,basic-options) ; another template inheriting basic-options

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location ^U Undo
^X Exit ^R Read File ^M Replace ^V Paste ^J Justify ^_ Go To Line ^E Redo
```

Figura 3.37. Configuración del directorio sip.conf

A continuación, se crear los usuarios, pero antes de eso se debe tener en cuenta una plantilla para poder agregar toda la configuración de los usuarios que se van a conectar a la central telefónica, la plantilla va a tener los siguientes parámetros:

- **Type=friend:** es para que pueda enviar y recibir las llamadas de otras extensiones.
- **Host=Dynamic:** es para el equipo se registre como cliente con una dirección IP.
- **Context=pbx:** es un contexto predefinido que se va usar en el directorio “extensions.conf”.

Luego de tener la plantilla ya creada, se procede a configurar cada uno de los usuarios que se van a conectar a la central telefónica, en este caso vamos a contar con 3 usuarios, los usuarios van a contar con las siguientes características (**Figura 3.38**):

- Nombre del usuario.
- Nombre de la plantilla a la que pertenece cada usuario.
- Número de extensión del usuario.
- La clave para cada extensión.
- El puerto de salida de la llamada que por defecto es el 5061.

```

GNU nano 7.2 /etc/asterisk/sip.conf *
[usuario](!)
type=friend
host=dynamic
context=pbx

; Extensi3n 1001
[ext1001](usuario)
username=Esteban
secret=12345678
port=5061

; Extensi3n 1002
[ext1002](usuario)
username=Luis
secret=12345678
port=5061

; Extensi3n 1003
[ext1003](usuario)
username=Anthony

```

Figura 3.38. Creaci3n de la plantilla para la PBX

Se ingresa a la consola de Asterisk con el comando "Asterisk -rvvv" para verificar todo lo que va pasa al momento de configurar los softphones o hacer alg3n cambio en la central telef3nica privada.

Se debe tener en cuenta que en esta consola se puede verificar la IP que le asigna Asterisk a cada dispositivo que se va a conectar, las extensiones que tienen y si el servicio est3 "OK".

Para verificar este proceso, se ingresa con el comando "sip show users" o "sip show peers", dentro de la consola. Una vez ya en este punto, se seleccionarn la aplicaci3n que tenga compatibilidad con el software para ya empezar a dar funcionamiento a la central telef3nica privada (PBX), como se ilustra en la **Figura 3.39**.

```

esteban@esteban-desktop:~$ sudo asterisk -rvvv
Asterisk 20.4.0-dfsg+cs6.13.40431414-2, Copyright (C) 1999 - 2022, Sangoma Technologies Corporation
and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 20.4.0-dfsg+cs6.13.40431414-2 currently running on esteban-desktop (pid = 153
29)
esteban-desktop*CLI> sip show peers
Name/username      Host                Dyn Forcerport Conedia  ACL Port
  Status  Description
ext1001/Esteban    192.168.0.107      D Auto (No) No      5599
1 OK (8 ms)
ext1002/Luis      192.168.0.106      D Auto (No) No      4450
4 OK (30 ms)
ext1003/Anthony   (Unspecified)     D Auto (No) No      0
UNKNOWN
3 sip peers [Monitored: 2 online, 1 offline Unmonitored: 0 online, 0 offline]
esteban-desktop*CLI>

```

Figura 3.39. Ingreso a la consola de Asterisk

3.4 Implementación del prototipo

En este punto se realiza la explicación de la instalación y configuración del softphone seleccionado. Esto ayudará a tener comunicación entre los usuarios conectados a nuestra central y a su vez a realizar las pruebas de funcionamiento requeridas para dar un servicio eficiente al usuario. El softphone es uno de los elementos básicos de la estructura ya que es la herramienta necesaria para la entrada y salida de llamadas que se van a realizar.

Esquema del prototipo

En el siguiente apartado se muestra la estructura general de todos los elementos de software y hardware que posee el prototipo de la central telefónica privada, en la **Figura 3.40** se exhibe el esquema del proyecto. Se puede mirar que el Raspberry Pi 4 es el que se encarga de operar todo el sistema.

Este posee la tarjeta micro SD en donde está cargado el sistema operativo para poder levantar nuestra central, opcional a esto está interconectado con un monitor mediante un cable HDMI para poder configurar mediante comandos todo lo que necesitamos para configurar Asterisk, para que esto se pueda realizar correctamente, se conecta a un router el cual va a dar la salida a internet para nuestro raspberry pi 4.

Se necesita de una IP que nos da la red de internet de manera dinámica para poder interconectar los teléfonos celulares a la central telefónica privada. Finalmente se conectan los teléfonos celulares (Cada uno con la aplicación ZoiPer) a la central telefónica para que se active el servicio de voz, es decir cada dispositivo va a tener una dirección IP asignada con la cual se va a identificar dentro del sistema.

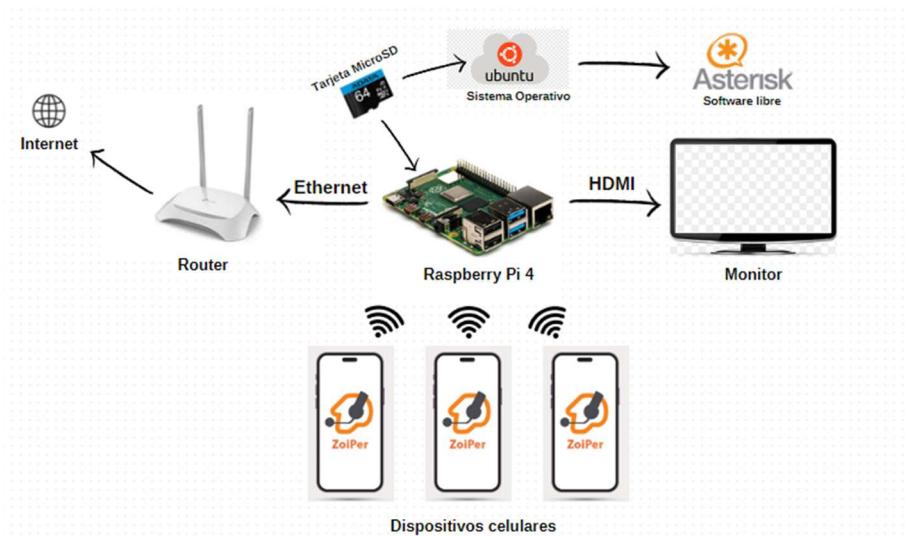


Figura 3.40. Esquema del prototipo

La idea de todo este sistema, es que se puede tener servicio de voz entre los dispositivos conectados, con la finalidad de realizar llamadas con una excelente calidad. Cuando una persona cuente con la aplicación en su celular, podrá tener llamadas ilimitadas dentro del zona de cobertura de la red de internet a la que está conectada.

Se debe tener en cuenta que la central telefónica está configurada para funcionar en una sola red con una sola IP, es decir que, si el dispositivo está conectado en otra red con diferente dirección IP no va a funcionar la central, es por eso que el sistema está ligado a funcionar de manera correcta para una sola red con una IP estática que ayuda a que los dispositivos se conecten automáticamente.

Instalación y configuración del softphone seleccionado

En el siguiente punto se va a explicar detalladamente la instalación y configuración del softphone para cada teléfono inteligente. Para este proyecto se va a trabajar con ZoiPer ya que es un softphone fácil de configurar que cuenta con todos los requisitos para la comunicación de voz.

Instalación de ZoiPer

La instalación de ZoiPer es muy sencilla, es decir, esta aplicación se la puede encontrar en la Playstore como una versión Lite como se observa en la **Figura 3.41**, ya que permite manejar solamente una extensión, en caso de se requiera tener más extensiones y manejar algunas otras opciones dentro de la aplicación se deberá pagar por el servicio.



Figura 3.41. Instalación de ZoiPer

Como se puede observar en la **Figura 3.42**, ZoiPer ya está instalado en nuestro teléfono inteligente, se debe tomar en cuenta que para este proyecto se va a utilizar la versión “Lite” del softphone, ya que ofrece una buena calidad para la comunicación de voz y es más que suficiente para realizar las pruebas necesarias de nuestra central telefónica privada.

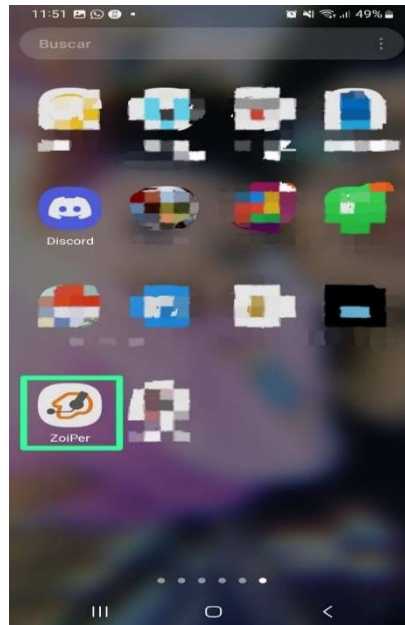


Figura 3.42. Verificación de la aplicación instalada

Configuración de ZoiPer

Se procede a configurar los softphone, aquí es donde va a servir que nuestro sistema operativo este con una dirección estática ya que los softphones necesita de una IP estática para poder tener una conexión fija, por lo que debe estar siempre en la misma red que el sistema operativo en el que se está levantando la central telefónica. Primero se abre la aplicación ZoiPer y se coloca en la primera ventana la extensión junto a la dirección IP de la central y su contraseña (**Figura 3.43**).



Figura 3.43. Datos de la cuenta de ZoiPer

En la **Figura 3.44**, se coloca una de las extensiones que previamente fueron creadas en Asterisk



Figura 3.44. Extensión del usuario

En la **Figura 3.45**, se coloca la dirección IP estática que pusimos inicialmente a la central telefónica.



Figura 3.45. IP del sistema PBX

Una vez realizado este proceso, se espera que la cuenta se active automáticamente por medio de la IP, para ya poder realizar las llamadas a los otros usuarios que están conectados a la central telefónica como se aprecia en la **Figura 3.46**.

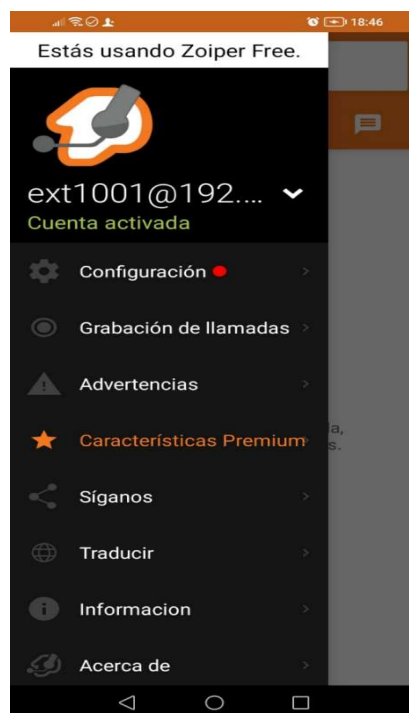


Figura 3.46. Cuenta activa de ZoiPer

Se verifica nuevamente en la consola y se observa que los dispositivos ya estén activados con su nombre, la extensión y su IP correspondiente. Como se puede

observar en la **Figura 3.47**, ya están activos los dispositivos que se configuro con la aplicación ZoiPer.

```
esteban@esteban-desktop: ~
[Nov 26 18:42:53] NOTICE[16595]: chan_sip.c:25011 handle_response_peerpoke: Peer 'ext1002' is now Re
achable. (7ms / 2000ms)
  Unregistered SIP 'ext1002'
  Registered SIP 'ext1001' at 192.168.0.107:55991
[Nov 26 18:42:57] NOTICE[16595]: chan_sip.c:25011 handle_response_peerpoke: Peer 'ext1001' is now Re
achable. (8ms / 2000ms)
  Unregistered SIP 'ext1001'
  Registered SIP 'ext1002' at 192.168.0.106:44504
  Registered SIP 'ext1001' at 192.168.0.107:55991
  Using SIP RTP CoS mark 5
[Nov 26 18:43:31] NOTICE[16595][C-00000001]: chan_sip.c:26828 handle_request_invite: Call from 'Este
ban' (192.168.0.107:55991) to extension '1002' rejected because extension not found in context 'pbx'
.
esteban-desktop*CLI> sip show peers
Name/username      Host                               Dyn Forcerport Comedia   ACL Port
Status            Description
ext1001/Esteban    192.168.0.107                     D Auto (No)  No        5599
1 OK (8 ms)
ext1002/Luis       192.168.0.106                     D Auto (No)  No        4450
4 OK (30 ms)
ext1003/Anthony    (Unspecified)                     D Auto (No)  No         0
UNKNOWN
3 sip peers [Monitored: 2 online, 1 offline Unmonitored: 0 online, 0 offline]
esteban-desktop*CLI> 
```

Figura 3.47. Consola con los dispositivos conectados

Verificación de la comunicación de voz entre los 3 clientes

Luego de a ver configurado los softphones, se realiza una primera prueba entre los teléfonos, pero hay un problema y es que todavía no sale las llamadas, ya que se necesita aun configurar el directorio de las extensiones. Para eso se tiene que hacer una copia como en el directorio “sip.conf” y así poder configurar el directorio “extensions.conf” (**Figura 3.48**).

```
esteban@esteban-desktop: ~$ sudo mv /etc/asterisk/extensions.conf /etc/asterisk/extension.conf.copia
[sudo] password for esteban:
```

Figura 3.48. Dirección del directorio extension.conf

Aquí lo que se va a colocar es la extensión creada, la prioridad 1 por donde van a salir las llamadas, se va agregar el dial para la ejecución de las llamadas de entrada y salida, se utilizara el protocolo SIP para llamar a cada una de las extensiones como se ve en la **Figura 3.49**, es decir con esta configuración ya se podrá realizar las llamadas de manera correcta. Estas configuraciones son requisitos esenciales que nos van ayudar a tener una comunicación de voz eficiente en llamadas entrantes y salientes de la central.


```

GNU nano 7.2 /etc/asterisk/extensions.conf
[pbx]
exten => 1001,1,Dial(SIP/ext1001)
exten => 1002,1,Dial(SIP/ext1002)
exten => 1003,1,Dial(SIP/ext1003)

```

Figura 3.49. Configuración en el directorio extension.conf

En este apartado como se ilustra en la **Figura 3.50**. Se hace una recarga en la consola nuevamente con el comando “dialplan reload”, con la finalidad de ver qué cambios se obtuvieron.

```

esteban-desktop*CLI> sip show peers
Name/username      Host                               Dyn Forcerport Conedia  ACL Port
Status            Description
ext1001/Esteban    192.168.0.107                     D Auto (No)  No      5599
1 OK (8 ms)
ext1002/Luis       192.168.0.106                     D Auto (No)  No      4450
4 OK (30 ms)
ext1003/Anthony    (Unspecified)                    D Auto (No)  No       0
UNKNOWN
3 sip peers [Monitored: 2 online, 1 offline Unmonitored: 0 online, 0 offline]
esteban-desktop*CLI> dialplan reload
Dialplan reloaded.
Including switch 'Lua/' in context 'default'
Including switch 'Lua/' in context 'public'
Including switch 'Lua/' in context 'demo'
Including switch 'Lua/' in context 'local'
Including switch 'DUNDi/e164' in context 'ael-dundi-e164-switch'
Time to scan old dialplan and merge leftovers back into the new: 0.001872 sec
Time to restore hints and swap in new dialplan: 0.000027 sec
Time to delete the old dialplan: 0.000409 sec
Total time merge_contexts_delete: 0.002308 sec
pbx_config successfully loaded 50 contexts (enable debug for details).
esteban-desktop*CLI>

```

Figura 3.50. Reinicio de la consola Asterisk

Por último, se verifica que en nuestra consola los softphones estén conectados y se procede a realizar las pruebas del funcionamiento del prototipo para verificar que el servicio este en óptimas condiciones de la central telefónica privada (ver **Figura 3.51**). Se debe tomar en cuenta que cada dispositivo que esté conectado a nuestra central telefónica privada, debe estar “OK” ya que esto indica que el dispositivo está listo para poder realizar llamadas dentro de la misma red en el cual está conectado el raspberry pi 4.

```
esteban@esteban-desktop:~$ ssh -X esteban@esteban-desktop
esteban@esteban-desktop:~$ ssh -X esteban@esteban-desktop
3 sip peers [Monitored: 2 online, 1 offline Unmonitored: 0 online, 0 offline]
esteban-desktop*CLI> dialplan reload
Dialplan reloaded.
  Including switch 'Lua/' in context 'default'
  Including switch 'Lua/' in context 'public'
  Including switch 'Lua/' in context 'demo'
  Including switch 'Lua/' in context 'local'
  Including switch 'DUNDI/e164' in context 'ael-dundi-e164-switch'
Time to scan old dialplan and merge leftovers back into the new: 0.001872 sec
Time to restore hints and swap in new dialplan: 0.000027 sec
Time to delete the old dialplan: 0.000409 sec
Total time merge_contexts delete: 0.002308 sec
pbx_config successfully loaded 50 contexts (enable debug for details).
esteban-desktop*CLI> sip show peers
Name/username      Host                               Dyn Forcerport Comedia  ACL Port  Status  Description
-----
ext1001/Esteban    192.168.0.107                     D Auto (No) No        55991   OK (10 ms)
ext1002/Luis       192.168.0.106                     D Auto (No) No        44504   OK (15 ms)
ext1003/Anthony    (Unspecified)                    D Auto (No) No         0       UNKNOWN
-----
3 sip peers [Monitored: 2 online, 1 offline Unmonitored: 0 online, 0 offline]
esteban-desktop*CLI> 
```

Figura 3.51. Verificación de la consola Asterisk

3.5 Realizar pruebas de funcionamiento del prototipo

Por último, se empieza con las pruebas de funcionamiento, ya que aquí se valida si la central PBX este funcionamiento correctamente. Todos los resultados obtenidos serán utilizados para subsanar los errores en caso de que se presente alguno, garantizando que el prototipo cumpla con todos los puntos que se establecieron.

Conexión y configuración para cada dispositivo

Como se puede observar en la **Figura 3.52**. Los dispositivos ya están listos para ser configurados con su respectiva extensión y dirección IP para empezar con las pruebas de llamada para los tres clientes. Se puede observar el raspberry pi 4 conectado al router con un cable ethernet hacia su puerto RJ45 y alimentándose con su respectivo cargador, de igual manera se puede observar el router conectado, ya que necesitamos de una red para que nuestra central telefónica pueda funcionar, es decir necesitamos de una dirección estática para cada dispositivo.

Se debe tener en cuenta que los dispositivos a conectarse deben estar conectado en la misma red que el raspberry pi 4, caso contrario no se podrán conectar a la central telefónica privada. La configuración en la aplicación ZoiPer se debe realizar como ya se explicó antes para los 3 teléfonos con su extensión correspondiente para que se puedan conectar sin ningún problema.



Figura 3.52. Central telefónica privada PBX

Prueba de funcionamiento desde la extensión 1001 a 1002

En la siguiente **Figura 3.53**, se puede observar la llamada saliente desde el dispositivo con la extensión 1001 a el dispositivo con la extensión 1002. La prueba de funcionamiento de los siguientes dispositivos funciona de manera correcta ya que la conexión en la que están los teléfonos inteligentes es la misma.

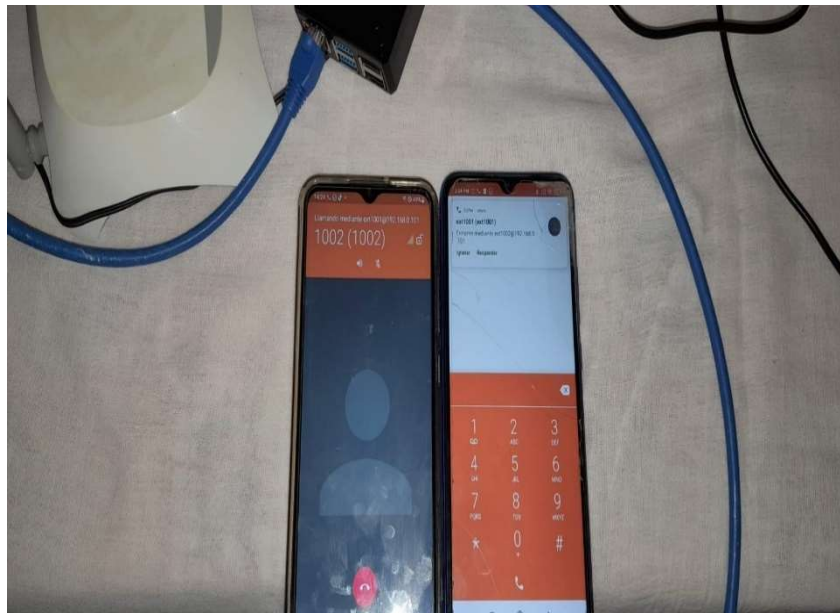


Figura 3.53. Llamada saliente desde la extensión 1001 a 1002

Prueba de funcionamiento desde la extensión 1002 a 1001

En la **Figura 3.54**, se puede observar la llamada saliente desde el dispositivo 1002 hacia el dispositivo 1001, las llamadas se están realizando sin ningún problema entre los

dispositivos celulares con la aplicación ZoiPer. Las llamadas se devolvieron desde un celular hacia otro con la finalidad de verificar que tanto el teléfono con la extensión 1001, 1002 y 1003 podían realizar llamadas sin ningún inconveniente.

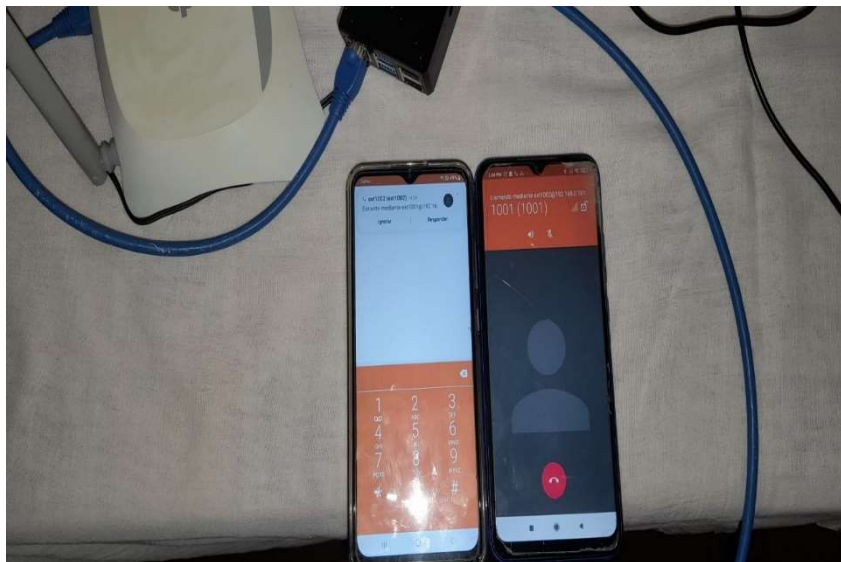


Figura 3.54. Llamada saliente desde la extensión 1002 a 1001

Prueba de funcionamiento desde la extensión 1002 a 1003

En la **Figura 3.55**, al igual que las anteriores pruebas de funcionamiento aquí se muestra la llamada saliente desde el dispositivo con la extensión 1001 hacia el dispositivo con la extensión 1003.

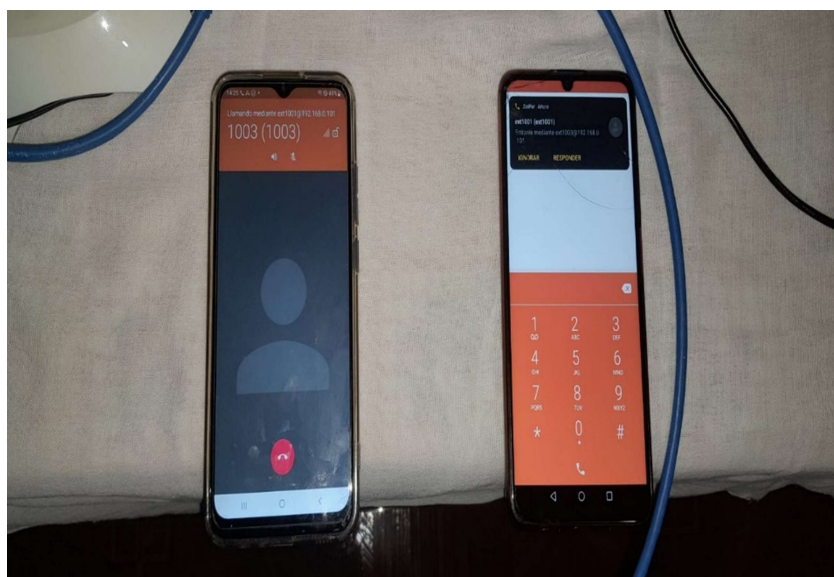


Figura 3.55. Llamada saliente desde la extensión 1002 a 1003

Prueba de funcionamiento desde la extensión 1003 a 1001

De igual manera se puede observar que en la **Figura 3.56**, se realiza la llamada saliente desde el dispositivo con la extensión 1003 hacia el dispositivo con la extensión 1001.

Con las pruebas de funcionamiento ya realizadas se puede constatar que la central telefónica está funcionando de manera correcta tanto en las llamadas entrantes como en las salientes de todos los usuarios que cuenten con un softphone y estén dentro de la misma red.

Las pruebas de voz se van a indicar de mejor manera en el video que estará en el link al final del trabajo escrito, ahí se mostrará la comunicación de voz que se va a tener entre los tres usuarios, ya que mediante imágenes no será posible indicar el funcionamiento de voz.

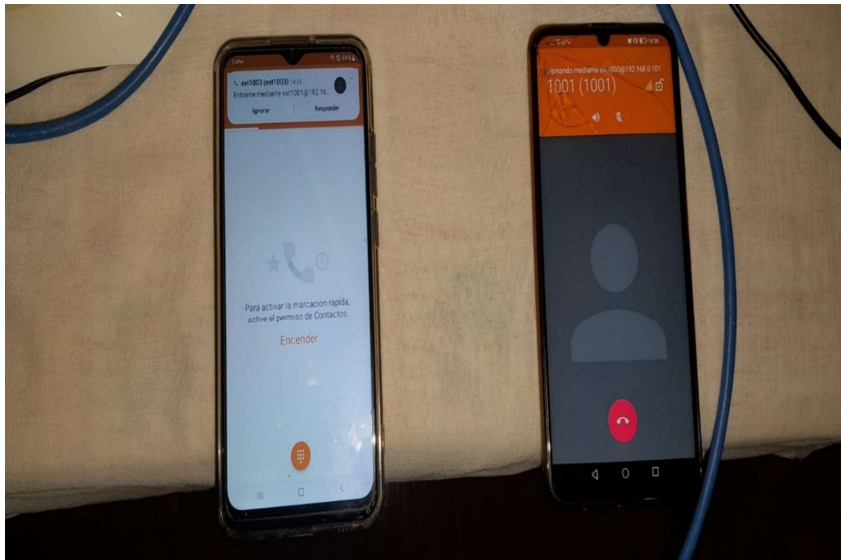


Figura 3.56. Llamada saliente desde la extensión 1003 a 1001

4 CONCLUSIONES

- La implementación de la central telefónica privada (PBX), se ha destacado como una herramienta esencial en este proyecto, ya que hoy en día la infraestructura de las comunicaciones permite una gestión eficiente en las llamadas entrantes y salientes, sin la necesidad de optar por infraestructuras robustas que existen en el mercado, generando un ahorro significativo de recursos para su implementación.
- Las investigaciones realizadas sobre softwares libres para el despliegue de la central telefónica privada indican un panorama donde no solo ofrecen opciones económicas viables, sino también ofrecen herramientas personalizables y flexibles que permiten que los sistemas de comunicación se adapten según sus necesidades específicas. Es por eso que, de todos los softwares investigados, se optó por trabajar con Asterisk ya que es un software muy interactivo que no solo nos ofrece compatibilidad con telefonía tradicional y líneas de telefonía IP, sino también servicios como: correo de voz, música en espera, transferencia de llamadas, entre otras.
- La selección de hardware y software para la central telefónica privada, se muestra como un paso estratégico y crítico, ya que son herramientas esenciales para un despliegue exitoso y muy eficiente, por lo que nos brindan rendimientos óptimos a largos plazos, es por eso que las herramientas seleccionadas como ZoiPer, Ubuntu, Asterisk, raspberry pi 4, etc. Brindan escalabilidad, adaptabilidad y seguridad, es decir factores claves que nos definirán el éxito constante de la central telefónica privada.
- Mediante la configuración de Asterisk se obtuvo como resultado la PBX la cual está basado en el protocolo SIP para la comunicación entre usuarios. Para llegar a ello se configuraron los directorios “sip.conf” y “extension.conf”, en donde se creó el plan de marcado para así poder realizar llamadas eficientes entre los clientes que se conecten a la PBX.
- Para la representación del prototipo, se tomó al raspberry pi 4 para que por medio del sistema operativo se pueda crear la centralita, en donde la tarjeta MicroSD es el cerebro general de la PBX, en donde se realizaron todas las configuraciones para brindar servicio de telefonía a los 3 clientes.
- Mediante las pruebas realizadas se pudo corroborar que el prototipo está funcionando de manera óptima y eficiente. Se pudo verificar que las llamadas entrantes y salientes se están realizando de manera correcta sin ningún

problema, al igual que la comunicación de voz entre los 3 clientes, gracias al formato de audio WAV, los códecs y los estándares que se están utilizando como G.722, que nos ayudan a tener una óptima comunicación de voz.

5 RECOMENDACIONES

- Realizar un análisis comparativo de diferentes soluciones de software y hardware para el diseño de la central telefónica privada (PBX), considerando diferentes aspectos como la fácil instalación, escalabilidad y la compatibilidad con los dispositivos que formaran parte de la PBX, para el correcto funcionamiento del sistema.
- Se debe trabajar con una dirección estática para el sistema, ya que ayudara a la rápida interconectividad entre los dispositivos celulares con el raspberry pi 4, esto se llevará a cabo con la finalidad de no estar configurando la dirección IP cada vez que se quiera ingresar a la central telefónica privada, en otras palabras, no se debe manejar la PBX con direccionamiento dinámico.
- Escoger una MicroSD clase 10 con suficiente espacio, ya que ayuda a no tener problemas al momento de realizar la instalación del sistema operativo Ubuntu en donde se va realizar la instalación del software Asterisk con todos sus requisitos necesarios para la PBX.
- Tener en cuenta que los software y hardware seleccionados abarquen el tema de seguridad y privacidad para poder implementar la central telefónica privada, con la finalidad de proteger la información personal y poder prevenir las posibles amenazas de cualquier intruso, ya que se debe tener en cuenta que el sistema funciona mediante una dirección IP que nos asigna la red en la que se está trabajando.
- En caso de no contar con un monitor, es recomendable mediante un protocolo SSH, el cual nos va a permitir trabajar de manera remota junto a una aplicación llamada "Putty", esto facilita la interacción con el CLI al momento de configurar la central telefónica privada.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] C. E. A. Samaniego, «Implementación de una central telefónica PBX basada en ASTERISK,» 2011. [En línea]. Available: <https://dspace.uazuay.edu.ec/bitstream/datos/2091/1/08204.pdf>. [Último acceso: 20 Diciembre 2023].
- [2] S. Karapantazis, «Computer Networks,» ScienceDirect, 13 Agosto 2009. [En línea]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S1389128609001200>. [Último acceso: 20 Diciembre 2023].
- [3] A. N. C. MEDINA, «Diseño de un sistema de telefonía IP basado en software libre e integración con la red de datos; como alternativa de comunicación de voz sobre el protocolo IP entre dependencias del Gobierno Autónomo Descentralizado Municipal de San Miguel de Ibarra,» 25 Marzo 2013. [En línea]. Available: <http://repositorio.utn.edu.ec/bitstream/123456789/1778/1/RED%20023%20TESI%20S.pdf>. [Último acceso: 23 Diciembre 2023].
- [4] V. C. M. L. Paola Renata Montenegro Cantos, «Análisis y evaluación para la selección de codecs de VoIP,» 01 Diciembre 2017. [En línea]. Available: <https://dspace.uazuay.edu.ec/bitstream/datos/2271/1/05787.pdf>. [Último acceso: 23 Diciembre 2023].
- [5] A. Marín García, «Evaluación experimental de QoE/QoS en redes inalámbricas 802.11,» 17 Diciembre 2014. [En línea]. Available: <https://repositorio.upct.es/bitstream/handle/10317/4285/pfc5792.pdf?sequence=1&isAllowed=y>. [Último acceso: 25 Diciembre 2023].
- [6] C. A. N. Hidalgo, «COMPRESIÓN DE AUDIO EN BASE A LA APLICACIÓN DE TÉCNICAS DE CUANTIFICACIÓN NO LINEAL PARA DETERMINAR EL EFECTO SONORO DE SU REPRODUCCIÓN EN LOS SERES HUMANOS.,» Septiembre 2010. [En línea]. Available: <https://repositorio.uta.edu.ec/bitstream/123456789/71/1/t551m.pdf>. [Último acceso: 25 Diciembre 2023].
- [7] B. G. d. V. Ordovás, «Desarrollo e implementación de un sistema de VoIP basado en Asterisk y PBX,» 2012. [En línea]. Available: <https://e->

archivo.uc3m.es/bitstream/handle/10016/16947/TFG_Borja_Garcia_de_Vinuesa_Ordovas.pdf. [Último acceso: 31 Enero 2024].

- [8] G. C. J. E. Guerrero Cueva Juan Pablo, «Definición de un procedimiento de pruebas para definir la capacidad, disponibilidad y QoS de un servidor Asterisk,» 09 Noviembre 2010. [En línea]. Available: <https://www.dspace.espol.edu.ec/bitstream/123456789/13543/1/D-42838.pdf>. [Último acceso: 23 Diciembre 2023].
- [9] J. G. Isaac, «MANUAL DE CONFIGURACION DE ASTERISK CON SIP TRUNKING DE VOZTELECOM (OIGAA DIRECT),» 8 Diciembre 2008. [En línea]. Available: https://d1wqtxts1xzle7.cloudfront.net/35998465/Manual_Asterisk_Oigaa-libre.pdf?1418949958=&response-content-disposition=inline%3B+filename%3DManual_Oigaa_Asterisk_MANUAL_DE_CONFIGUR.pdf&Expires=1703371704&Signature=ToBcwnhNVAHP4rRb2btKTN~Lp8n8Ftoub01DoLq5. [Último acceso: 23 Diciembre 2023].
- [10] P. M. Juliá, «Software libre, Linux y Ubuntu,» [En línea]. Available: <https://digitum.um.es/digitum/bitstream/10201/29185/1/Software%20libre,%20Linux%20y%20Ubuntu.pdf>. [Último acceso: 12 Diciembre 2023].
- [11] S. Á. M. M. E. G. Daniel Chora Garcia, «Informática y sistemas,» *Revista de tecnologías de la informática y las telecomunicaciones*, vol. II, nº 1, pp. 74-82, 2018.
- [12] C. V. Miranda, *Redes Tekenáticas*, Madrid: Paraninfo, 2015.
- [13] G. M. E. J. Sánchez Rodríguez Héctor Andrés, «Rack de pruebas Redes LAN,» 2014. [En línea]. Available: https://repository.uniminuto.edu/bitstream/10656/3495/1/TR_SanchezRodriguezHectorAndres_2014.pdf. [Último acceso: 23 Diciembre 2023].
- [14] R. C. Martín, «Desarrollo de una librería para microcontrolador dsPIC33F para la lectura de tarjetas microSD en un dispositivo inercial,» [En línea]. Available: <https://core.ac.uk/download/pdf/41810713.pdf>. [Último acceso: 23 Diciembre 2023].
- [15] C. J. P. R. F. E. Ñ. R. J. L. A. A. Jhonson Hailer Araujo Garcia1, «Implementación de una red celular GSM mediante software OPENBTS,» 1 Abril 2019. [En línea].

Available: <http://journal.upao.edu.pe/PuebloContinente/article/view/1256/1088>.
[Último acceso: 23 Diciembre 2023].

- [16] Anónimo, «Elastix Logo.jpg,» voip.ms, 10 Junio 2019. [En línea]. Available: https://wiki.voip.ms/w/index.php?title=File:Elastix_Logo.jpg&filetimestamp=20190610154352. [Último acceso: 25 Diciembre 2023].
- [17] FreePBX, «FreePBX - Let Freedom Ring,» FreePBX, 16 Diciembre 2019. [En línea]. Available: <https://www.freepbx.org/>. [Último acceso: 26 Diciembre 2023].
- [18] masIP, «¿Qué es Issabel y cómo funciona?,» masIP, 23 Octubre 2019. [En línea]. Available: <https://www.masip.es/blog/que-es-issabel/>. [Último acceso: 26 Diciembre 2023].
- [19] Anónimo, «MICRO SD 64GB CLASE 10,» 330ohms, [En línea]. Available: <https://www.330ohms.com/products/micro-sd-64gb-clase-10>. [Último acceso: 26 Diciembre 2023].
- [20] Anónimo, «Compare 3CX and MicroSIP,» Anónimo, [En línea]. Available: <https://www.g2.com/compare/3cx-vs-microsip>. [Último acceso: 15 1 2024].

7 ANEXOS

La lista de los **Anexos** se muestra a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces

ANEXO III. Conjunto de datos extensos

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 21 de febrero de 2024

De mi consideración:

Yo, CARLOS ANDRÉS YUNGA SÁNCHEZ, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN PROTOTIPO DE PBX USANDO SOFTWARE LIBRE asociado a la IMPLEMENTACIÓN DE UN PROTOTIPO DE PBX USANDO SOFTWARE LIBRE elaborado por la estudiante ANTHONY ESTEBAN SUNTAXI ÑATO de la carrera en Tecnología Superior en Redes y Telecomunicaciones, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 7%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

LINK

[3. Reporte de Turnitin Anthony Suntaxi Tic 2023B.pdf](#)

Atentamente,

Carlos Andrés Yunga Sánchez

Técnico Docente

Escuela de Formación de Tecnólogos

ANEXO II: Enlaces



Anexo II.I Código QR de la implementación y pruebas de funcionamiento

Link: [Anthony Suntaxi Video Final.mp4](#)

ANEXO III: Códigos Fuente