

Redes de Telecomunicaciones

Práctica 2: Implementación de un servidor VoIP

Franklin Mauricio Gómez López, Juan Diego Tenesaca Illares

Universidad de Cuenca, Facultad de Ingeniería,
Av. 12 de Abril y Agustín Cueva, EC010112, Cuenca, ECU
`franklin.gomez@ucuenca.edu.ec`, `juan.tenesaca@ucuenca.edu.ec`

Resumen En este documento se presenta el proceso detallado de implementación de un servidor SIP mediante el sistema operativo de Issabel en una máquina virtual de VirtualBox, el cual fue troncalizado con otro servidor SIP que emplea Asterisk para ofrecer servicios de telefonía IP. Además, se llevó a cabo un análisis de los protocolos presentes en la comunicación, utilizando la herramienta Wireshark, con el fin de comprender el funcionamiento de la telefonía IP y los protocolos que se emplean en este tipo de comunicación. Este proyecto tiene como objetivo principal entender el funcionamiento de la telefonía IP, así como profundizar en los aspectos técnicos que intervienen en la implementación de un servidor SIP y la troncalización con otros servidores.

Keywords: VoIP · Issabel · RTP · SIP · Softphones · Wireshark · Troncalización

1. Introducción

La telefonía tradicional ha evolucionado enormemente en los últimos años, especialmente con el avance de las tecnologías de la información y las comunicaciones. La telefonía IP, o VoIP (Voz sobre Protocolo de Internet), es un ejemplo de esta evolución, ya que permite la transmisión de voz a través de Internet utilizando diferentes protocolos y tecnologías.

En la telefonía IP, la voz se convierte en datos y se transmite por la red en paquetes. Esta tecnología permite una mayor flexibilidad, ya que se pueden realizar llamadas desde cualquier lugar con conexión a Internet, lo que significa un ahorro considerable en costos de telefonía. Además, permite una mayor integración de servicios como correo de voz, videoconferencia, mensajería instantánea, entre otros.

En este informe se exponen los pasos seguidos para brindar un servicio de telefonía IP y está dividido en varias secciones. En la primera sección se encuentra la Introducción, que proporciona una visión general del tema a tratar. La segunda sección, es la del Marco Teórico, en donde se exponen los conceptos y se describe los programas y herramientas empleadas para levantar un servicio de VoIP.

En la tercera sección, se encuentra el Metodología y Desarrollo, sección en la que se presenta la instalación y configuración de Issabel y los clientes. Como cuarta sección se tiene el Análisis de Resultados, en esta sección se emplea Wireshark para capturar y

analizar los paquetes enviados durante una llamada, se analizan los mensajes SIP que intervienen en la llamada y también se analizan los paquetes RTP que transportan los datos de audio. Las conclusiones obtenidas se presentan en la siguiente sección. Finalmente, el informe se cierra con la sección de referencias, donde se incluyen todas las fuentes consultadas.

2. Marco Teórico

2.1. Tecnología VoIP

Aunque VoIP puede definirse de forma abreviada como una tecnología que aprovecha el protocolo TCP/IP para ofrecer conversaciones de voz, lo cierto es que es mucho más que esto. VoIP puede ser usada para reemplazar la telefonía tradicional en un entorno empresarial, en un pequeño negocio o en casa, o simplemente para añadir ventajas a un sistema de telefonía tradicional [1].

La tecnología VoIP permite la emisión de voz en paquetes IP a través de redes de datos, como Internet, lo que une dos mundos que antes estaban separados: la transmisión de voz y la de datos. Con VoIP, la voz es transportada en paquetes encapsulados para su transmisión sin necesidad de una infraestructura telefónica convencional. Esto permite el desarrollo de una red homogénea para enviar todo tipo de información, incluyendo voz, video y datos [2].

La utilización de una única red para la transmisión de voz y datos tiene muchas ventajas, como obtener mayores beneficios para proveedores de servicios de telefonía y datos y ahorrar en gastos de infraestructura y mantenimiento. Las llamadas telefónicas tradicionales requieren una gran inversión en infraestructura para conectar centralitas entre sí con cableado, fibra óptica, satélites de telecomunicación o cualquier otro medio. En cambio, las llamadas telefónicas VoIP comprimen la voz y la envían en paquetes de datos por una línea, lo que permite que diferentes llamadas e incluso diferentes tipos de datos viajen juntos sin necesidad de líneas dedicadas ni desaprovechamiento del ancho de banda [2].

La VoIP ofrece la flexibilidad de utilizar una arquitectura distribuida o centralizada, ambas con sus propios beneficios y desventajas. El enfoque centralizado puede obstaculizar futuras innovaciones tecnológicas debido a que todo está concentrado en un solo punto, mientras que la arquitectura distribuida puede ser más compleja. Sin embargo, independientemente del enfoque elegido, la VoIP proporciona una gran flexibilidad en la forma en que se distribuyen y utilizan los recursos de comunicación, en la figura 1.

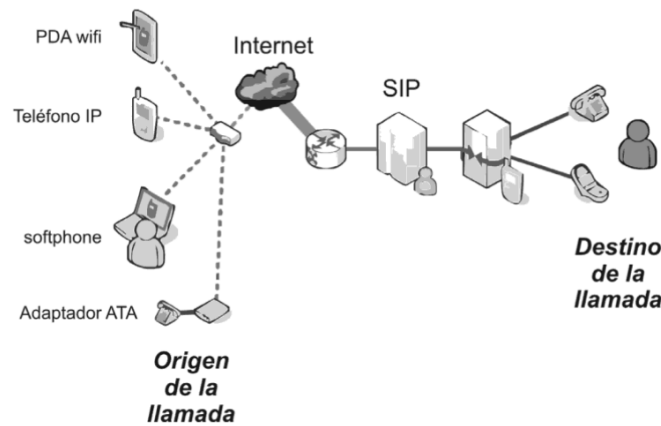


Figura 1: Arquitectura de la telefonía VoIP

2.2. Sistemas de telefonía IP

Los sistemas de telefonía IP se dividen en tres categorías principales: PBX IP, centralita virtual y servicio de telefonía en la nube. Los PBX IP son sistemas de telefonía empresarial que utilizan el protocolo IP para la transmisión de llamadas y ofrecen funciones avanzadas como el enrutamiento de llamadas, la grabación de llamadas y la integración de correo de voz. Asterisk es un ejemplo popular de PBX IP de código abierto, y su distribución Issabel es una opción popular para aquellos que buscan una solución de comunicaciones unificadas basada en Asterisk. Las centralitas virtuales son soluciones basadas en la nube que ofrecen funciones similares a las de un PBX IP, pero con la ventaja de que se pueden administrar y escalar fácilmente. Los servicios de telefonía en la nube son sistemas de telefonía que se ejecutan completamente en la nube y eliminan la necesidad de hardware o infraestructura local. Estos servicios a menudo se pagan mediante suscripción y ofrecen una fácil escalabilidad y flexibilidad [3].

Issabel PBX Issabel es una solución de comunicaciones unificadas que combina características como PBX IP, correo electrónico, mensajería instantánea, fax y otras funciones colaborativas. Cuenta con una interfaz web y ofrece la posibilidad de tener un centro de llamadas. Tras la adquisición de Elastix por parte de 3CX, Issabel surgió para continuar el desarrollo de PBX con el apoyo de una comunidad de expertos, empresas y colaboradores [4].

Issabel ha sido diseñado para proporcionar una gran capacidad de llamadas simultáneas utilizando la infraestructura de red actual. Esta solución se integra con la red de telefonía pública, ofreciendo comunicación VoIP dentro y fuera de la oficina. Issabel permite mantener anexos remotos en otras ciudades o países conectándolos por internet,

así como interconectar sucursales en ubicaciones remotas a través del acceso a internet, lo que reduce significativamente los costos de comunicación entre ellas [5]. Además, Issabel permite utilizar computadoras como teléfonos a través de Softphone, o teléfonos tradicionales mediante adaptadores especiales, todo sobre la red LAN existente [4].

2.3. Protocolo SIP

El protocolo SIP es utilizado para establecer llamadas de voz o video en vivo a través de una red IP. Se utiliza como un protocolo de señalización para crear, modificar y finalizar sesiones que involucran uno o más participantes. Estas sesiones pueden ser establecidas como llamadas telefónicas de dos vías o como conversaciones entre múltiples participantes, también conocidas como conferencias. Este protocolo ha permitido la creación de un conjunto de servicios que antes parecían imposibles, como la telefonía IP, los mensajes instantáneos, la transmisión de voz y video, entre otros [4].

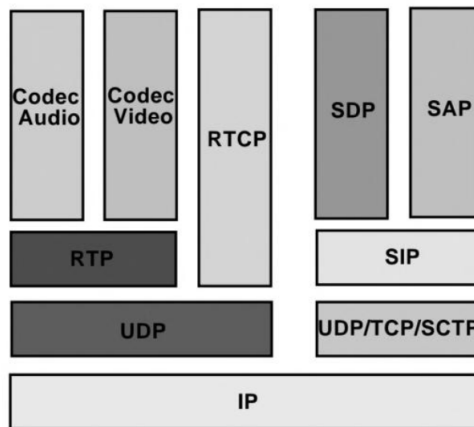


Figura 2: Arquitectura del protocolo SIP

Solicitudes SIP: El funcionamiento de este protocolo se basa en un conjunto de peticiones y respuestas que permiten la comunicación entre los dispositivos. En la figura 3, se pueden observar las distintas peticiones del protocolo SIP. Cada una de estas peticiones tiene una función específica dentro del protocolo, lo que permite la creación y finalización de sesiones multimedia de manera eficiente y efectiva en las redes IP.

INVITE	Es la petición SIP que se envía a un usuario cuando queremos establecer con él una comunicación, una llamada.
ACK	Esta petición es enviada por el usuario origen que envió la petición INVITE para hacer saber al usuario destino que su respuesta 200 OK ha sido recibida. Es el momento en que ambos pueden empezar a enviar tráfico Media.
BYE	Para finalizar la conexión, la comunicación entre los dos usuarios establecida anteriormente con INVITE.
CANCEL	Se utiliza para cancelar una petición, por ejemplo INVITE, que se encuentra en progreso. Por ejemplo si el teléfono destino está sonando pero aún no ha sido descolgado y el teléfono origen cuelga, se envía un CANCEL a diferencia de un BYE que se enviaría si el teléfono destino hubiera sido descolgado previamente y por tanto la comunicación establecida unos instantes.
OPTIONS	Un UA puede enviar peticiones OPTIONS a un UAS para solicitar cierta información sobre este.
REGISTER	Un UAC envía peticiones REGISTER a un servidor de registro-localización para informar de la posición actual en la que se encuentra en un momento determinado. Esto hace posible que el UAC pueda ser localizado haciendo uso de su misma dirección user@dominio sin importar donde el UAC se encuentre físicamente.

Figura 3: Peticiones del protocolo SIP

Respuestas SIP: Después de haber recibido e interpretado un requerimiento SIP, el destinatario de este requerimiento devuelve una respuesta SIP. Existen seis clases de respuestas:

- 1xx = respuestas informativas, tal como 180, la cual significa teléfono sonando.
- 2xx = respuestas de éxito
- 3xx = respuestas de redirección
- 4xx = errores de solicitud
- 5xx = errores de servidor
- 6xx = errores globales

3. Metodología - Desarrollo

Para la realización de esta práctica se empleó Issabel como plataforma de comunicaciones, ya que esta distribución de software libre incluye la funcionalidad de telefonía IP, en este caso se instaló este sistema como una máquina virtual y como *software* de virtualización se empleó VirtualBox en su versión 6.1.42 y se empleó la versión 20.20 de Issabel.

3.1. Instalación y configuración de Issabel

Como se mencionó anteriormente, en esta práctica se utilizó una máquina virtual con el sistema operativo Issabel. Para empezar, es necesario descargar la imagen ISO del sistema y crear la máquina virtual. Durante este proceso, es crucial configurar correctamente la red de la máquina virtual. Para que el servicio de Issabel sea accesible desde cualquier dispositivo, la máquina virtual debe estar conectada a un adaptador puente, como se muestra en la figura 4.

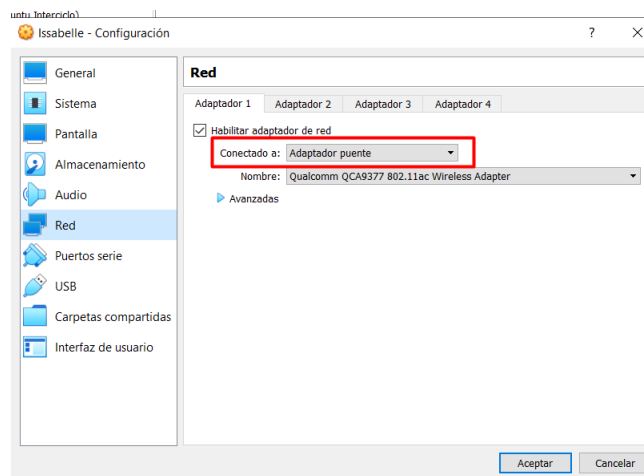


Figura 4: Configuración de red de la máquina virtual

Una vez arrancada la máquina con la imagen ISO, se mostrará una pantalla similar a la mostrada en la figura 5, donde se puede seleccionar el idioma principal del sistema operativo.

Una vez seleccionado el lenguaje, se configurará la fecha y hora, así como el teclado y la selección del origen y destino de la instalación, como se muestra en la figura 6.

Una vez seleccionados estos parámetros ya se puede iniciar la instalación del sistema y mientras se realiza se debe crear una contraseña para el usuario root, que permitirá iniciar el servicio de Issabel, en la figura 7 se muestra la forma en la que se crea esta contraseña, en esta práctica la contraseña root es *nand74ls00*.

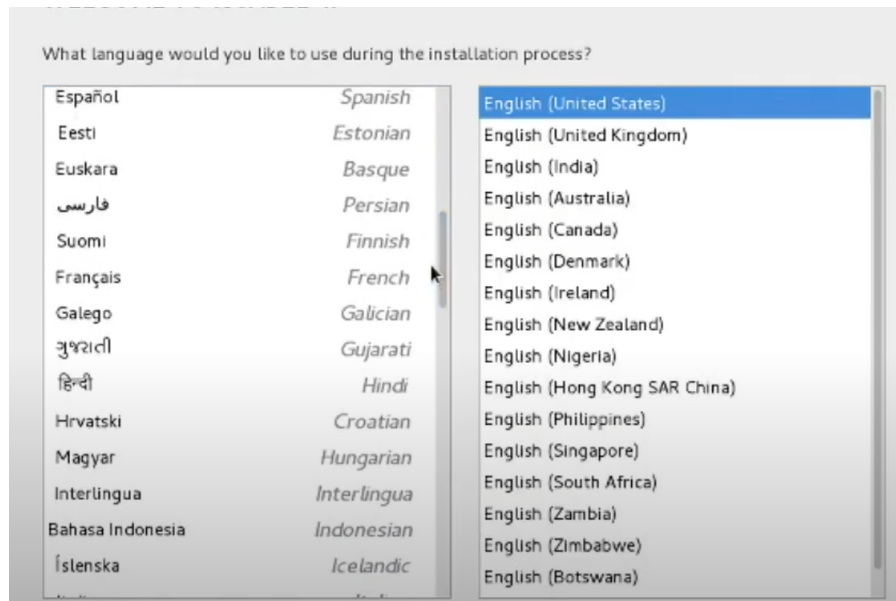


Figura 5: Selección del lenguaje principal de Issabel



Figura 6: Instalación de Issabel

Una vez que se ha terminado de instalar el sistema, el siguiente paso es configurarlo, para ello se debe crear una contraseña para la base de datos de usuarios, como se muestra en la figura 8, la contraseña empleada en esta práctica es *nand74ls00*. Lo siguiente es asignar una contraseña para el usuario *admin*, la cual es necesaria para acceder al *dashboard* de Issabel, en la figura 9 se muestra la creación de esta contraseña, la cual para motivos de esta práctica es *admin123*.

Una vez finalizada la configuración solo basta con iniciar sesión con el usuario root para levantar el sistema de Issabel, como se observa en la figura 10.

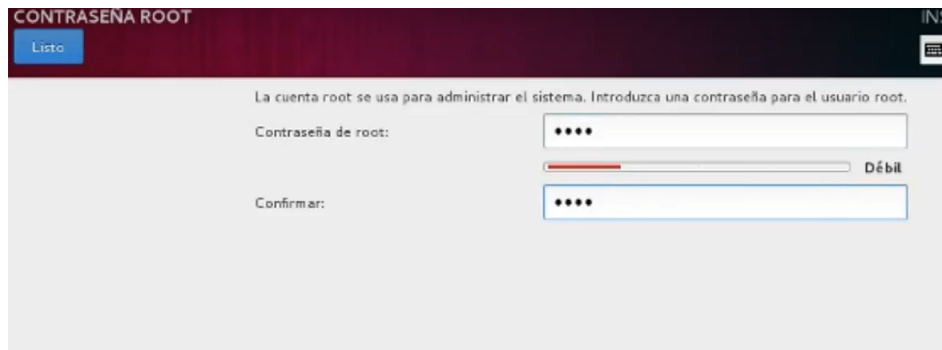


Figura 7: Creación de la contraseña para el usuario root

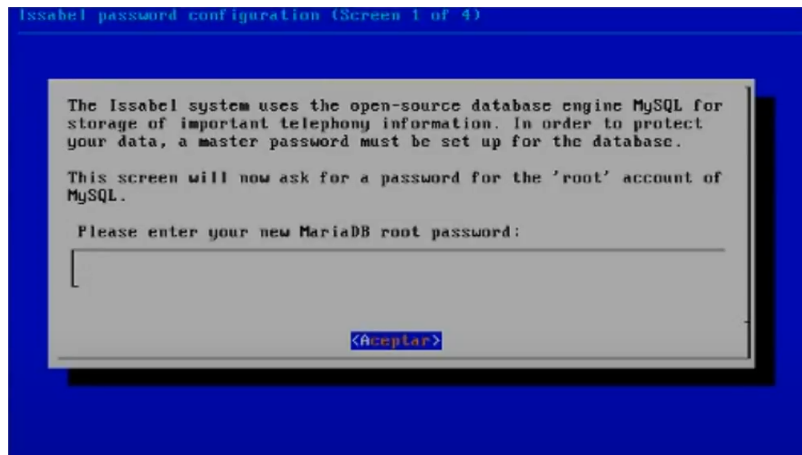


Figura 8: Contraseña para la base de datos

Figura 9: Contraseña para el usuario *admin*

```

Issabel 4
Kernel 3.10.0-1062.el7.x86_64 on an x86_64

issabel login: root
Password:
Login incorrect

issabel login: root
Password:
Last failed login: Mon Apr  3 23:20:14 -05 2023 on tty1
There were 4 failed login attempts since the last successful login.
Last login: Mon Apr  3 23:18:58 on

  0 0 0   Issabel is a product meant to be configured through a web browser.
  0 0 0   Any changes made from within the command line may corrupt the system
  0 0 0   configuration and produce unexpected behavior; in addition, changes
  0      made to system files through here may be lost when doing an update.

To access your Issabel System, using a separate workstation (PC/MAC/Linux)
Open the Internet Browser using the following URL:

https://192.168.211.129

Your opportunity to give back: http://www.patreon.com/issabel

System load:  0.12 (1min) 0.11 (5min) 0.06 (15min)      Uptime:  5 min
Asterisk:     Asterisk 16.7.0                        Active Calls: 0
Memory:       [=====>-----] 31% 305/972M
Usage on /:   [=====>-----] 18% 2,9/18G
Swap usage:   0.0%
SSH logins:   1 open sessions
Processes:    123 total, 85 yours

[root@issabel ~]#

```

Figura 10: Inicio de sesión en Issabel

La gran ventaja de usar Issabel como servidor VoIP, es que cuenta con una interfaz gráfica accesible desde cualquier navegador, lo que facilita la adición de extensiones y la creación de troncales, para acceder a ella basta con ingresar la dirección IP del servidor de Issabel y se presentará una pantalla como la mostrada en la figura 11, en la cual se debe ingresar con el usuario *admin*, para acceder a la interfaz del servidor. En primera instancia se mostrará un resumen del estado y rendimiento del servidor, como se muestra en la figura 12.



Figura 11: Inicio de sesión en el panel de control de Issabel

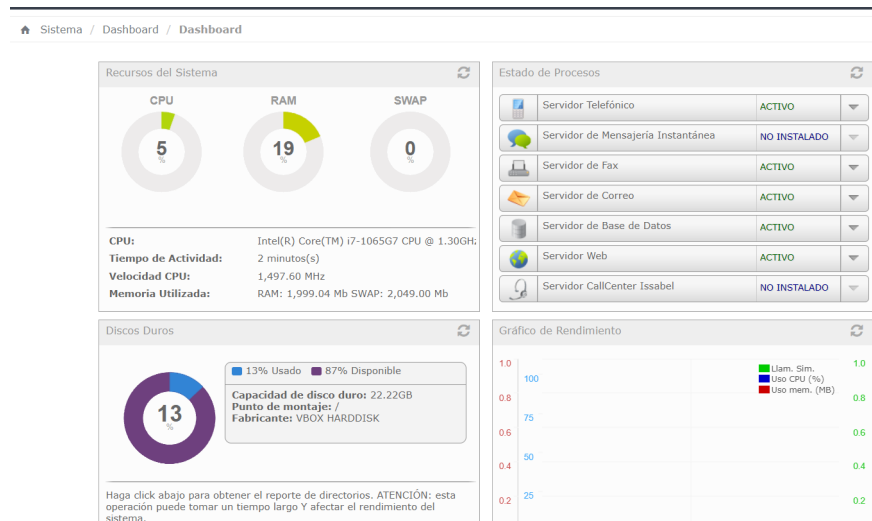


Figura 12: Dashboard de Issabel

Creación de extensiones: Una vez que el servidor ha sido configurado correctamente, es necesario agregar las extensiones correspondientes para que los usuarios puedan acceder al servicio. En Issabel, este proceso se lleva a cabo a través de la configuración de PBX, tal como se muestra en la figura 13. Al seleccionar la opción *Añadir extensión* y hacer clic en *Enviar*, se puede crear una nueva extensión, como se muestra en 14.

Una vez creada la extensión, es importante configurarla correctamente para que pueda ser utilizada por los usuarios. En la figura 15, se muestra un ejemplo de cómo se configura una extensión en Issabel. En este caso, la nueva extensión es la 2002 y se le asigna el nombre de *JD*. Además, es necesario asignar una contraseña para que la aplicación cliente pueda conectarse a esta extensión.

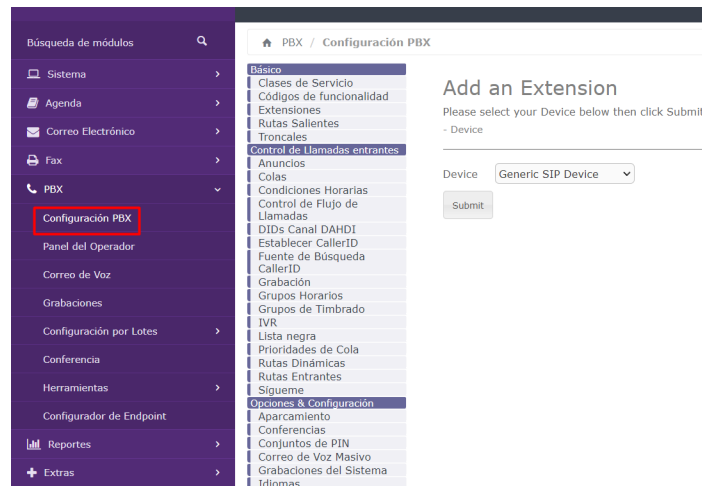


Figura 13: Configuración PBX

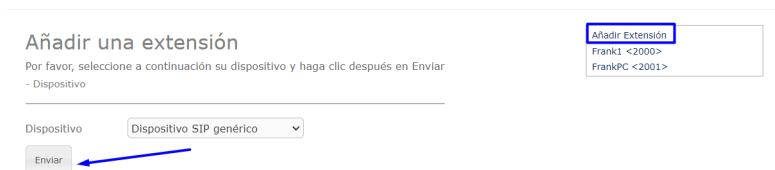


Figura 14: Adición de nueva extensión

Add SIP Extension

- Añadir extensión

Extensión del usuario

Nombre para mostrar

CID Num Alias

Alias SIP

- Opciones de la extensión

CID saliente

Asterisk Dial Options ☐

Ring Time

Call Forward Ring Time

Outbound Concurrency Limit

Llamada en espera

Internal Auto Answer

Call Screening

Pinless Dialing

CID de emergencia

- Assigned DID/CID

Figura 15: Configuración de la nueva extensión

Creación de troncales: Como se mencionó anteriormente, para esta practica se conectan dos servidores y para establecer la conexión entre los servidores Issabel y Asterisk, es necesario crear una troncal que permita a los usuarios de Issabel llamar a los usuarios de Asterisk. En Issabel, la creación de troncales se lleva a cabo desde la configuración PBX.

En la figura 16 se muestra la configuración realizada en este caso, se observa que es necesario especificar la dirección IP del servidor Asterisk, que en este caso es **192.168.0.110**, también se debe definir la conexión como *amistosa* (*type=friend*) para permitir la llamada de cualquier usuario que pertenezca al segundo servidor.

A continuación, se procedió a crear una ruta de salida que permitiera a los usuarios de Issabel comunicarse con los usuarios de Asterisk. En la figura 17 se muestra la configuración de dicha ruta de salida, en la cual se definió el prefijo 7 para realizar llamadas hacia el servidor Asterisk. Esto significa que, para llamar a un usuario de Asterisk, es necesario marcar primero el número siete. Por último, se conectó la ruta de salida con la troncal creada anteriormente, lo cual se logró seleccionando la troncal en la configuración de la ruta de salida.

Opciones & Configuración

- Aparcamiento
- Conferencias
- Conjuntos de PIN
- Correo de Voz Masivo
- Grabaciones del Sistema
- Idiomas
- Intercom y Paginación
- Música en Espera
- Otras Aplicaciones
- Otros Destinos
- Acceso Remoto
- DISA
- Devolver Llamada
- Opciones Avanzadas
- Administrar Correo de Voz
- Destinos Personalizados
- Escribir en Queue Log
- Extensiones Personalizadas
- Información de Asterisk
- Inyección de Dialplan
- Mensajes de Congestión
- Registros Asterisk
- Ajustes
- Clases de Servicio (admin)
- Configuraciones AMI
- Configuraciones Avanzadas
- Configuraciones TAX
- Configuraciones Logger
- Configuraciones SIP
- Opcion
- IssabelPBX sin Embeber

+ Agregar Más Patrones de Marcado

Limpiar todos los Campos

Asistente de reglas de marcación: (seleccione uno)

Prefijo de marcación externa:

Opciones salientes

Nombre de la línea troncal: diego

Detalles del par:
context=from-internal
host=192.168.0.110
type=friend

Opciones entrantes

Contexto del usuario: diego2

Detalles del usuario:


Registro

Cadena de registro:

Enviar cambios Duplicar Troncal

Figura 16: Creación de línea troncal en Issabel

Editar ruta

 Eliminar ruta IsaatoAste

Route Settings

Nombre de la ruta:

Route CID:

Contraseña de la ruta:

Route Type: ☐ Emergencia ☐ Intra-Company

¿Música en espera?

Time Group:

Route Position:

Additional Settings

Call Recording:

PIN Set:

Dial Patterns that will use this Route

() + [0XX] /

(prepend) + prefix | match pattern / CallerID

+ Agregar Más Patrones de Marcado

Asistente de reglas de marcación: (seleccione uno)

Export Dialplans as CSV:

Trunk Sequence for Matched Routes

0

1

Figura 17: Creación de ruta de salida en Issabel

3.2. Configuración de los clientes:

Como cliente o *softphone* se empleó el software Zoiper tanto para PC como su versión móvil, en la figura 18 se muestra como se configura el cliente, basta con colocar la dirección IP del servidor, el nombre de la extensión asignada y la clave secreta o contraseña de cada extensión.

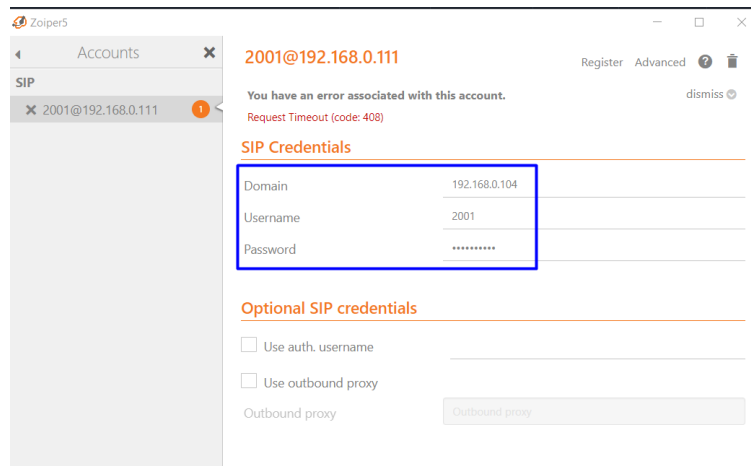


Figura 18: Configuración de Zoiper en Windows

3.3. Funcionamiento del servidor VoIP:

Para probar el correcto funcionamiento del servicio de telefonía IP se realizó llamadas entre diferentes extensiones, en la figura 19 se muestra una llamada entre la extensión de 2001 de Issabel y la extensión 098 de Asterisk, en esta figura también se observa que para llamar al segundo servidor se emplea el prefijo que se mostró en la figura 17.

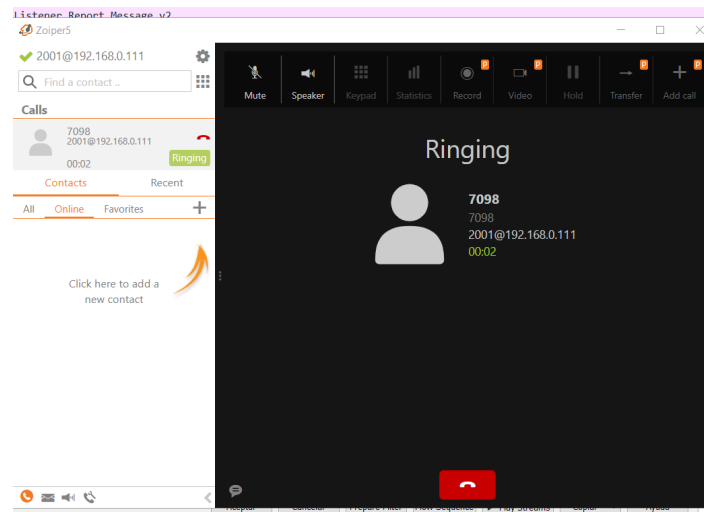


Figura 19: Realización de una llamada desde la extensión 2001 de Issabel hacia la extensión 098 de Asterisk

4. Análisis de Resultados

4.1. Análisis de protocolos con Wireshark

Análisis del protocolo SIP: Para establecer las llamadas de voz a través de la red IP, Issabel utiliza el protocolo SIP como protocolo de señalización, para crear, modificar y finalizar sesiones y, como se mencionó en la sección del marco teórico, SIP se basa en un conjunto de peticiones y respuestas que permiten la comunicación entre dispositivos, en esta sección se analizará como Issabel utiliza estas peticiones y respuestas para la comunicación de voz, para ello se empleó el software Wireshark para capturar los paquetes transmitidos durante una llamada IP.

En la tabla 1 se muestra las condiciones en las que se realizó la prueba de telefonía IP entre un remitente que utiliza el servidor Asterisk con la dirección IP 192.168.0.110 y la extensión *ext098*, y un destinatario que utiliza el servidor Issabel con la dirección IP 192.168.0.104 y la extensión *2001*.

Tabla 1: Condiciones de la prueba de telefonía IP (VoIP)

	Información del remitente	Información del destinatario
Servidor	Asterisk	Issabel
Dirección IP del servidor	192.168.0.110	192.168.0.104
Nombre de la extensión	ext098	2001

En la figura 20 se muestran varios paquetes capturados con Wireshark, se resaltan las peticiones *OPTIONS* y la respuesta *200 OK*. En el protocolo SIP, un cliente emplea esta petición para solicitar información al servidor sobre este. Mientras que las respuestas que pertenecen al grupo *2xx* corresponden a respuestas que informan del éxito de una petición SIP, en este caso se observa una respuesta *200 OK*.

En la figura 21 se muestra a detalle el paquete capturado de la petición *OPTIONS*. Se observa que primero se detalla de qué tipo de mensaje SIP se trata, en este caso de una petición *OPTIONS*. El siguiente campo de interés que se observa es el de **Via**, en donde se almacena cada uno de los elementos por los que va pasando la petición. El siguiente campo es el de **Max-Forwards**, que indica el número máximo de saltos permitidos a la petición para llegar a su destino, en el caso de este paquete este campo tiene el valor de 70.

Después se encuentra el campo **From**, que indica la entidad origen que envió la petición SIP. Después se tiene **To**, que hace referencia a la AOR (*Address of Record*) de destino de la petición.

El siguiente campo de interés es el de **Call-ID**, el cual es un identificador único y global. La combinación de las etiquetas indicadas en **To**, **From** junto con el *Call-ID* definen e identifican de manera única un diálogo SIP entre dos extremos.

Cseq es un contador de peticiones pertenecientes a un mismo diálogo.

Contact, en él se indica la SIP URI de la forma *usuario@direcciónIP:puerto*.

De igual forma, en la figura 20 se muestra a detalle el paquete capturado de la respuesta *200 OK*, se observa que posee los mismo campos descritos anteriores, con la única diferencia de que el emisor es el servidor Issabel y el receptor el servidor Asterisk y que en este paquete no existe el campo de **Max-Forwards**.

No.	Time	Source	Destination	Protocol	Length	Info
1...	1238.4628...	192.168.0.110	192.168.0.104	SIP	603	Request: OPTIONS sip:192.168.0.104
1...	1239.4744...	192.168.0.110	192.168.0.104	SIP	603	Request: OPTIONS sip:192.168.0.104
1...	1240.4562...	192.168.0.110	192.168.0.104	SIP	603	Request: OPTIONS sip:192.168.0.104
1...	1241.4554...	192.168.0.110	192.168.0.104	SIP	603	Request: OPTIONS sip:192.168.0.104
1...	1251.3331...	192.168.0.104	192.168.0.110	SIP	554	Status: 200 OK
1...	1251.3334...	192.168.0.104	192.168.0.110	SIP	554	Status: 200 OK
1...	1251.3335...	192.168.0.104	192.168.0.110	SIP	554	Status: 200 OK
1...	1251.3338...	192.168.0.104	192.168.0.110	SIP	554	Status: 200 OK

Figura 20: Paquetes capturados con Wireshark, solicitud *OPTIONS* y respuesta *200 OK*

<ul style="list-style-type: none"> Session Initiation Protocol (OPTIONS) <ul style="list-style-type: none"> Request-Line: OPTIONS sip:192.168.0.104 SIP/2.0 <ul style="list-style-type: none"> Method: OPTIONS Request-URI: sip:192.168.0.104 <ul style="list-style-type: none"> Request-URI Host Part: 192.168.0.104 [Resent Packet: False] Message Header <ul style="list-style-type: none"> Via: SIP/2.0/UDP 192.168.0.110:5060;branch=z9hG4bK5d7191ef <ul style="list-style-type: none"> Transport: UDP Sent-by Address: 192.168.0.110 Sent-by port: 5060 Branch: z9hG4bK5d7191ef Max-Forwards: 70 From: "asterisk" <sip:asterisk@192.168.0.110>;tag=as3e165097 <ul style="list-style-type: none"> SIP from display info: "asterisk" SIP from address: sip:asterisk@192.168.0.110 <ul style="list-style-type: none"> SIP from address User Part: asterisk SIP from address Host Part: 192.168.0.110 SIP from tag: as3e165097 To: <sip:192.168.0.104> <ul style="list-style-type: none"> SIP to address: sip:192.168.0.104 <ul style="list-style-type: none"> SIP to address Host Part: 192.168.0.104 Contact: <sip:asterisk@192.168.0.110:5060> <ul style="list-style-type: none"> Contact URI: sip:asterisk@192.168.0.110:5060 <ul style="list-style-type: none"> Contact URI User Part: asterisk Contact URI Host Part: 192.168.0.110 Contact URI Host Port: 5060 Call-ID: Saf6de4138bd7a0159ba78e673da194b@192.168.0.110:5060 [Generated Call-ID: Saf6de4138bd7a0159ba78e673da194b@192.168.0.110:5060] CSeq: 102 OPTIONS <ul style="list-style-type: none"> Sequence Number: 102 Method: OPTIONS User-Agent: Asterisk PBX 18.10.0-dfsg+~cs6.10.40431411-2 Date: Thu, 06 Apr 2023 19:56:01 GMT Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE Supported: replaces, timer Content-Length: 0 	<p>Información del remitente</p> <p>Información del destinatario</p>
--	--

Figura 21: Análisis del paquete de la solicitud *OPTIONS*

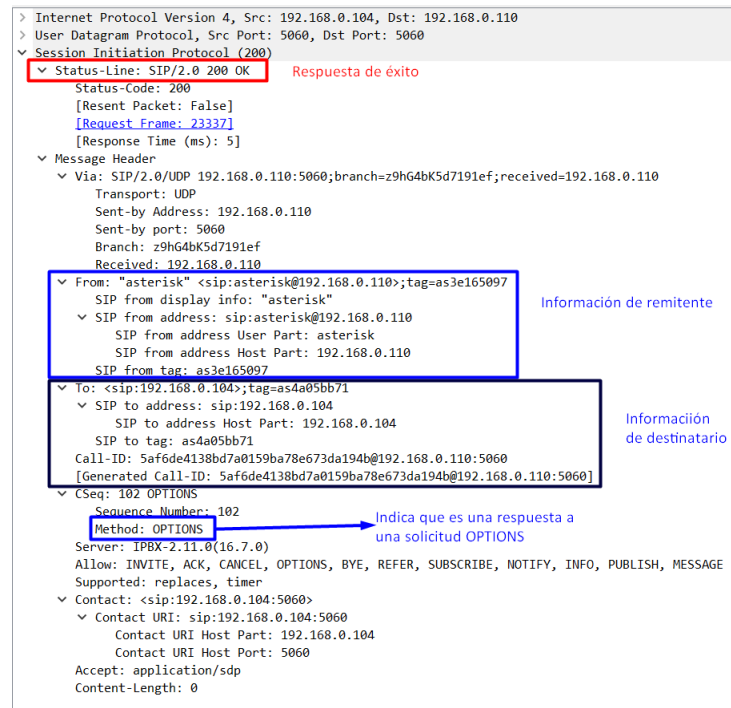


Figura 22: Análisis del paquete de la respuesta 200 OK

En la figura 23 se muestran los paquetes capturados con Wireshark cuando el servidor Issabel envía peticiones *REGISTER* al servidor de Asterisk. En esta figura también se observan las respuestas enviadas desde Asterisk, las cuales corresponden a los errores 401 y 403.

En la figura 24 se muestra a detalle el paquete capturado de la petición *REGISTER*. Esta petición es enviada por un cliente a un servidor de registro para informar la posición actual en la que se encuentra en un momento determinado, con el objetivo de que el cliente pueda ser localizado usando su misma dirección.

De igual forma, en la figura 25 se muestra el paquete de la respuesta 401, la cual se emplea para indicar que el cliente no está autorizado.

En la figura 26 se muestra una respuesta de error 403, la cual indica que el cliente está prohibido para el servidor.

1...	1323.9157...	192.168.0.104	192.168.0.110	SIP	554 Status: 200 OK
1...	1324.1045...	192.168.0.104	192.168.0.110	SIP	441 Request: REGISTER sip:192.168.0.110 (1 binding)
1...	1324.1308...	192.168.0.110	192.168.0.104	SIP	589 Status: 401 Unauthorized
1...	1324.1318...	192.168.0.104	192.168.0.110	SIP	602 Request: REGISTER sip:192.168.0.110 (1 binding)
1...	1324.2039...	192.168.0.110	192.168.0.104	SIP	510 Status: 403 Forbidden
1...	1325.4193...	192.168.0.110	192.168.0.104	SIP	603 Request: OPTIONS sip:192.168.0.104
1...	1325.4203...	192.168.0.104	192.168.0.110	SIP	554 Status: 200 OK

Figura 23: Paquetes capturados con Wireshark, solicitud *REGISTER* y respuestas de error *401* y *403*

```
> Frame 12251: 441 bytes on wire (3528 bits), 441 bytes captured (3528 bits) on interface \Device\NPF-
> Ethernet II, Src: Chongqin_8f:6b:a3 (4c:eb:bd:8f:6b:a3), Dst: 38:d5:7a:5e:fb:49 (38:d5:7a:5e:fb:49)
> Internet Protocol Version 4, Src: 192.168.0.104, Dst: 192.168.0.110
> User Datagram Protocol, Src Port: 5060, Dst Port: 5060
v Session Initiation Protocol (REGISTER)
  v Request-Line: REGISTER sip:192.168.0.110 SIP/2.0
    Method: REGISTER
    v Request-URI: sip:192.168.0.110
      Request-URI Host Part: 192.168.0.110
      [Resent Packet: False]
  v Message Header
    v Via: SIP/2.0/UDP 192.168.0.104:5060;branch=z9hG4bK044f40e3
      Transport: UDP
      Sent-by Address: 192.168.0.104
      Sent-by port: 5060
      Branch: z9hG4bK044f40e3
      Max-Forwards: 70
    v From: <sip:diego@192.168.0.110>;tag=as29739002
      v SIP from address: sip:diego@192.168.0.110
        SIP from address User Part: diego
        SIP from address Host Part: 192.168.0.110
        SIP from tag: as29739002
    v To: <sip:diego@192.168.0.110>
      v SIP to address: sip:diego@192.168.0.110
        SIP to address User Part: diego
        SIP to address Host Part: 192.168.0.110
        Call-ID: 505cc5e03ab13ba64c68bd115cd07abe@127.0.0.1
        [Generated Call-ID: 505cc5e03ab13ba64c68bd115cd07abe@127.0.0.1]
    v CSeq: 102 REGISTER
      Sequence Number: 102
      Method: REGISTER
      Supported: replaces, timer
      User-Agent: IPBX-2.11.0(16.7.0)
      Expires: 120
    v Contact: <sip:s@192.168.0.104:5060>
      v Contact URI: sip:s@192.168.0.104:5060
        Contact URI Host Part: 192.168.0.104
        Contact URI Port: 5060
      Content-Length: 0
```

Figura 24: Análisis del paquete de la solicitud *REGISTER*

```

> Internet Protocol Version 4, Src: 192.168.0.110, Dst: 192.168.0.104
> User Datagram Protocol, Src Port: 5060, Dst Port: 5060
< Session Initiation Protocol (401)
  < Status-Line: SIP/2.0 401 Unauthorized
    Status-Code: 401
    [Resent Packet: False]
    [Request Frame: 12251]
    [Response Time (ms): 26]
  < Message Header
    < Via: SIP/2.0/UDP 192.168.0.104:5060;branch=z9hG4bK044f40e3;received=192.168.0.104
      Transport: UDP
      Sent-by Address: 192.168.0.104
      Sent-by port: 5060
      Branch: z9hG4bK044f40e3
      Received: 192.168.0.104
    < From: <sip:diego@192.168.0.110>;tag=as29739002
      < SIP from address: sip:diego@192.168.0.110
        SIP from address User Part: diego
        SIP from address Host Part: 192.168.0.110
        SIP from tag: as29739002
      < To: <sip:diego@192.168.0.110>;tag=as46afab2c
        < SIP to address: sip:diego@192.168.0.110
          SIP to address User Part: diego
          SIP to address Host Part: 192.168.0.110
          SIP to tag: as46afab2c
        Call-ID: 505cc5e03ab13ba64c68bd115cd07abe@127.0.0.1
        [Generated Call-ID: 505cc5e03ab13ba64c68bd115cd07abe@127.0.0.1]
      < CSeq: 102 REGISTER
        Sequence Number: 102
        Method: REGISTER
        Server: Asterisk PBX 18.10.0~dfsg+~cs6.10.40431411-2
        Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE
        Supported: replaces, timer
      < WWW-Authenticate: Digest algorithm=MD5, realm="asterisk", nonce="2d84e456"
        Authentication Scheme: Digest
        Algorithm: MD5
        Realm: "asterisk"
        Nonce Value: "2d84e456"
      Content-Length: 0

```

Respuesta de error en la solicitud

Figura 25: Análisis del paquete de la respuesta 401

```

> Frame 12254: 510 bytes on wire (4080 bits), 510 bytes captured (4080 bits) on interface \Device\NPF...
> Ethernet II, Src: 38:d5:7a:5e:fb:49 (38:d5:7a:5e:fb:49), Dst: Chongqin_8f:6b:a3 (4c:eb:bd:8f:6b:a3)
> Internet Protocol Version 4, Src: 192.168.0.110, Dst: 192.168.0.104
> User Datagram Protocol, Src Port: 5060, Dst Port: 5060
> Session Initiation Protocol (403)
  Status-Line: SIP/2.0 403 Forbidden
    Status-Code: 403
    [Resent Packet: False]
    [Request Frame: 12253]
    [Response Time (ms): 72]
  Message Header
    Via: SIP/2.0/UDP 192.168.0.104;branch=z9hG4bK6fb25f25;received=192.168.0.104
      Transport: UDP
      Sent-by Address: 192.168.0.104
      Sent-by port: 5060
      Branch: z9hG4bK6fb25f25
      Received: 192.168.0.104
    From: <sip:diego@192.168.0.110>;tag=as29739002
      SIP from address: sip:diego@192.168.0.110
        SIP from address User Part: diego
        SIP from address Host Part: 192.168.0.110
        SIP from tag: as29739002
    To: <sip:diego@192.168.0.110>;tag=as46afab2c
      SIP to address: sip:diego@192.168.0.110
        SIP to address User Part: diego
        SIP to address Host Part: 192.168.0.110
        SIP to tag: as46afab2c
    Call-ID: 505cc5e03ab13ba64c68bd115cd07abe@127.0.0.1
    [Generated Call-ID: 505cc5e03ab13ba64c68bd115cd07abe@127.0.0.1]
    CSeq: 103 REGISTER
      Sequence Number: 103
      Method: REGISTER
    Server: Asterisk PBX 18.10.0~dfsg+~cs6.10.40431411-2
    Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE
    Supported: replaces, timer
    Content-Length: 0

```

Figura 26: Análisis del paquete de la respuesta 403

Otra de las peticiones del protocolo SIP es la de *CANCEL*, en la figura 27 se observa la captura de este mensaje SIP con su respectiva respuesta, este mensaje se utiliza para cancelar una solicitud, por ejemplo un *INVITE*, que se encuentra en progreso como es el caso de estos mensajes.

Esta solicitud es respondida con el código 487 *Request Terminated*, el cual confirma que una petición fue terminada.

En la figura 28 se muestra a detalle la composición de esta solicitud.

1.. 2141.9189..	192.168.0.110	192.168.0.104	SIP/SDP	929 Request: INVITE sip:2001@192.168.0.104
1.. 2142.9729..	192.168.0.110	192.168.0.104	SIP/SDP	929 Request: INVITE sip:2001@192.168.0.104
1.. 2142.9930..	192.168.0.104	192.168.0.110	SIP	551 Status: 100 Trying
1.. 2143.1287..	192.168.0.104	192.168.0.110	SIP	567 Status: 180 Ringing
1.. 2143.2321..	192.168.0.104	192.168.0.110	SIP	567 Status: 180 Ringing
1.. 2148.6624..	192.168.0.110	192.168.0.104	SIP	397 Request: CANCEL sip:2001@192.168.0.104
1.. 2148.6635..	192.168.0.104	192.168.0.110	SIP	501 Status: 487 Request Terminated
1.. 2148.6639..	192.168.0.104	192.168.0.110	SIP	485 Status: 200 OK
1.. 2148.7021..	192.168.0.110	192.168.0.104	SIP	453 Request: ACK sip:2001@192.168.0.104:5060

Figura 27: Paquetes capturados con Wireshark, solicitud *CANCEL* y respuestas 487

```

> Frame 15999: 397 bytes on wire (3176 bits), 397 bytes captured (3176 bits) on interface \Device\NPF-
> Ethernet II, Src: 38:d5:7a:5e:fb:49 (38:d5:7a:5e:fb:49), Dst: Chongqin_8f:6b:a3 (4c:eb:bd:8f:6b:a3)
> Internet Protocol Version 4, Src: 192.168.0.110, Dst: 192.168.0.104
> User Datagram Protocol, Src Port: 5060, Dst Port: 5060
√ Session Initiation Protocol (CANCEL)
  √ Request-Line: CANCEL sip:2001@192.168.0.104 SIP/2.0
    Method: CANCEL
    Request-URI: sip:2001@192.168.0.104
      Request-URI User Part: 2001
      Request-URI Host Part: 192.168.0.104
    [Resent Packet: False]
  √ Message Header
    √ Via: SIP/2.0/UDP 192.168.0.110:5060;branch=z9hG4bK0f380fcf
      Transport: UDP
      Sent-by Address: 192.168.0.110
      Sent-by port: 5060
      Branch: z9hG4bK0f380fcf
      Max-Forwards: 70
    √ From: <sip:ext098@192.168.0.110>;tag=as71d56ef6
      √ SIP from address: sip:ext098@192.168.0.110
        SIP from address User Part: ext098
        SIP from address Host Part: 192.168.0.110
      SIP from tag: as71d56ef6
    √ To: <sip:2001@192.168.0.104>
      √ SIP to address: sip:2001@192.168.0.104
        SIP to address User Part: 2001
        SIP to address Host Part: 192.168.0.104
      Call-ID: 43895676660c22a30e8300234d59721c@192.168.0.110:5060
      [Generated Call-ID: 43895676660c22a30e8300234d59721c@192.168.0.110:5060]
    √ CSeq: 102 CANCEL
      Sequence Number: 102
      Method: CANCEL
      User-Agent: Asterisk PBX 18.10.0-dfsg+~cs6.10.40431411-2
      Content-Length: 0

```

Solicitud CANCEL

Figura 28: Análisis del paquete de la solicitud *CANCEL*

La última petición analizada es la de *ACK*, la cual es enviada por el usuario origen que envió la petición *INVITE*, en este caso el servidor Asterisk, para hacer saber al usuario del servidor Issabel que su respuesta *200 OK* ha sido recibida correctamente. Después de la recepción de este mensaje ambos pueden empezar a enviar el tráfico de audio.

En la figura 29 se muestra la captura de paquetes de una llamada completa, desde el mensaje *INVITE*, hasta la finalización de la misma mediante el mensaje *BYE*. En la figura se resalta el mensaje de *ACK*, que inicia el envío de tráfico de audio entre los dos dispositivos.

El paquete de este mensaje de *ACK* se muestra a detalle en la figura 30, se observa que este paquete contiene los mismos campos analizados anteriormente.

Finalmente, en la figura 31 se observa un gráfico que muestra el intercambio de mensajes SIP entre el servidor Asterisk y el servidor Issabel, en este gráfico también se puede apreciar que después de recibir la petición *ACK* inicia el intercambio de datos mediante el protocolo RTP entre los dos dispositivos.

2...	2492.3536...	192.168.0.110	192.168.0.104	SIP/SDP	927 Request: INVITE sip:2001@192.168.0.104
2...	2492.3557...	192.168.0.104	192.168.0.110	SIP	551 Status: 100 Trying
2...	2492.4888...	192.168.0.104	192.168.0.110	SIP	567 Status: 180 Ringing
2...	2492.5991...	192.168.0.104	192.168.0.110	SIP	567 Status: 180 Ringing
2...	2496.1239...	192.168.0.104	192.168.0.110	SIP/SDP	886 Status: 200 OK
2...	2496.1294...	192.168.0.110	192.168.0.104	SIP	453 Request: ACK sip:2001@192.168.0.104:5060
2...	2527.1338...	192.168.0.110	192.168.0.104	SIP	603 Request: OPTIONS sip:192.168.0.104
2...	2527.1345...	192.168.0.104	192.168.0.110	SIP	554 Status: 200 OK
3...	2574.7232...	192.168.0.110	192.168.0.104	SIP	484 Request: BYE sip:2001@192.168.0.104:5060
3...	2574.7242...	192.168.0.104	192.168.0.110	SIP	482 Status: 200 OK

Figura 29: Paquetes capturados de una conversación completa

Session Initiation Protocol (ACK)	
Request-Line: ACK sip:2001@192.168.0.104:5060 SIP/2.0 Method: ACK	Solicitud ACK
Request-URI: sip:2001@192.168.0.104:5060 Request-URI User Part: 2001 Request-URI Host Part: 192.168.0.104 Request-URI Host Port: 5060 [Resent Packet: False] [Request Frame: 23446] [Response Time (ms): 3776]	
Message Header	
Via: SIP/2.0/UDP 192.168.0.110:5060;branch=z9hG4bK1406747f Transport: UDP Sent-by Address: 192.168.0.110 Sent-by port: 5060 Branch: z9hG4bK1406747f Max-Forwards: 70	
From: <sip:ext098@192.168.0.110>;tag=as1c4ec5b9 SIP from address: sip:ext098@192.168.0.110 SIP from address User Part: ext098 SIP from address Host Part: 192.168.0.110 SIP from tag: as1c4ec5b9	Información del remitente
To: <sip:2001@192.168.0.104>;tag=as4dec2bac SIP to address: sip:2001@192.168.0.104 SIP to address User Part: 2001 SIP to address Host Part: 192.168.0.104 SIP to tag: as4dec2bac	Información del destinatario
Contact: <sip:ext098@192.168.0.110:5060> Contact URI: sip:ext098@192.168.0.110:5060 Contact URI User Part: ext098 Contact URI Host Part: 192.168.0.110 Contact URI Host Port: 5060 Call-ID: 7818faf34f97f2404fe774e670e6a472@192.168.0.110:5060 [Generated Call-ID: 7818faf34f97f2404fe774e670e6a472@192.168.0.110:5060]	
CSeq: 102 ACK Sequence Number: 102 Method: ACK User-Agent: Asterisk PBX 18.10.0~dfsg+~cs6.10.40431411-2 Content-Length: 0	

Figura 30: Análisis del paquete de la petición ACK

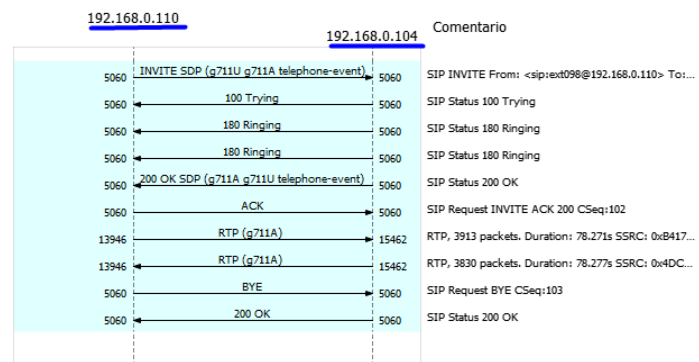


Figura 31: Gráfico del flujo de peticiones y respuestas en una llamada VoIP

Análisis del protocolo RTP: RTP es un protocolo estándar utilizado para la transmisión de audio y video en tiempo real a través de redes IP. En el caso de Issabel, cuando un usuario realiza una llamada de VoIP, la plataforma utiliza RTP para enviar los paquetes de audio en tiempo real entre los dispositivos que participan en la llamada, como los teléfonos IP o *softphones*.

Para analizar estos paquetes, Wireshark implementa una funcionalidad capaz de seguir y analizar los flujo RTP, esta funcionalidad se encuentra en la pestaña de Telefonía, en la figura 32 se muestra como acceder a dicha funcionalidad.

Esta funcionalidad permite analizar el flujo RTP tal como se muestra en la imagen 33, si se compara esta imagen con el gráfico mostrado en la figura 31 se puede comprobar que este flujo RTP corresponde con el de la llamada analizada anteriormente, ya que las direcciones de origen y destino corresponden así como el número de paquetes RTP esperados.

En la figura 34 se muestra las opciones que ofrece Wireshark para el análisis de flujos RTP, en la pestaña de *Gráfica* se puede observar la forma de las señales de las voces transmitidas, ya que esta información no se encuentra cifrada y debido a esta falta de cifrado, es posible reproducir los paquetes capturados, en la figura 35 se muestra cómo se ve la herramienta de *RTP Player*, que permite reproducir los paquetes capturados, al realizarlo se comprobó que efectivamente la grabación correspondía con el diálogo realizado durante la prueba.

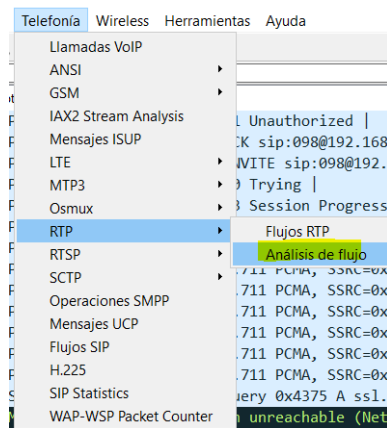


Figura 32: Análisis del flujo RTP con Wireshark

Forward		Envío	Retorno	Gráfica						
192.168.0.110:13946 → 192.168.0.104:15462		Paquete	Sequence	Delta (ms)	Jitter (ms)	Skew	Ancho de banda	Marker	Estado	
SSRC 0x0b4179b1		23494	29496	0.00	0.00	0.00	1.60		✓	
Max Delta 75.55 ms @ 26758		23495	29497	38.45	1.15	-18.45	3.20		✓	
Max Jitter 10.12 ms		23496	29498	19.18	1.13	-17.63	4.80		✓	
Mean Jitter 3.34 ms		23497	29499	20.63	1.10	-18.26	6.40		✓	
Max Skew -85.47 ms		23498	29500	19.24	1.08	-17.50	8.00		✓	
RTP Packets 3913		23500	29501	25.17	1.34	-22.67	9.60		✓	
Expected 3913		23501	29502	20.47	1.28	-23.14	11.20		✓	
Lost 0 (0.00 %)		23502	29503	16.59	1.41	-19.73	12.80		✓	
Seq Errs 0		23503	29504	17.51	1.48	-17.23	14.40		✓	
Start at 2496.245588 s @ 23494		23505	29505	20.41	1.41	-17.64	16.00		✓	
Duration 78.27 s		23507	29506	30.27	1.97	-27.91	17.60		✓	
Clock Drift -1 ms		23511	29507	19.96	1.85	-27.88	19.20		✓	
Freq Drift 8000 Hz (-0.00 %)		23513	29508	19.84	1.74	-27.72	20.80		✓	
Reverse		23517	29509	19.83	1.64	-27.55	22.40		✓	
192.168.0.104:15462 → 192.168.0.110:13946		23522	29510	38.61	2.70	-46.16	24.00		✓	
SSRC 0x4dccc2b4f		23523	29511	1.38	3.70	-27.54	25.60		✓	
Max Delta 0.00 ms @ 0		23525	29512	22.01	3.59	-29.55	27.20		✓	
Max Jitter 0.00 ms		23527	29513	19.57	3.40	-29.12	28.80		✓	
Mean Jitter 0.00 ms		23529	29514	19.44	3.22	-28.56	30.40		✓	
Max Skew 0.00 ms		23531	29515	18.91	3.09	-27.47	32.00		✓	
RTP Packets 0		23533	29516	19.95	2.90	-27.42	33.60		✓	
Expected 1		23535	29517	20.55	2.75	-27.97	35.20		✓	
Lost 1 (100.00 %)		23537	29518	19.46	2.61	-27.43	36.80		✓	
Seq Errs 0		23539	29519	20.73	2.49	-28.16	38.40		✓	
Start at 0.000000 s @ 0		23541	29520	20.44	2.36	-28.60	40.00		✓	
Duration 0.00 s		23543	29521	19.07	2.28	-27.67	41.60		✓	
Clock Drift 0 ms		23545	29522	20.05	2.14	-27.71	43.20		✓	
Freq Drift 1 Hz (0.00 %)										

Figura 33: Análisis del flujo RTP con Wireshark

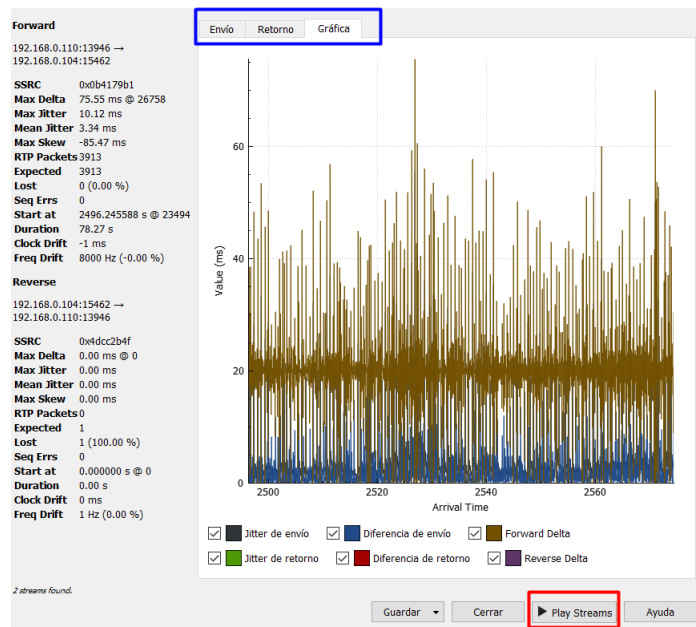


Figura 34: Análisis del flujo RTP con Wireshark



Figura 35: Reproducción de los paquetes RTP

5. Conclusiones

Tras realizar la práctica de implementación de un servicio de VoIP, se pueden destacar las siguientes conclusiones:

- La realización de esta práctica permitió una comprensión práctica del funcionamiento de la telefonía IP y la implementación a través del protocolo SIP. El análisis de los paquetes con Wireshark permitió observar de manera detallada el intercambio de peticiones y respuestas SIP, necesario para establecer la conexión entre el cliente y el servidor, y para enviar los datos de media mediante el protocolo RTP. Además, se pudo apreciar la ventaja de contar con una interfaz gráfica de usuario, como la que ofrece Issabel, en comparación con Asterisk, ya que facilita la administración y configuración del servicio de telefonía.
- Al tratarse de una implementación sencilla no empresarial y con software libre, un tema importante que se debe tomar en cuenta es el de la seguridad informática, ya que los datos enviados por RTP no cuentan con ningún tipo de cifrado, lo que los hace vulnerables a ser capturados y reproducidos por un software de análisis de red como Wireshark, lo que podría comprometer la privacidad y confidencialidad de las comunicaciones.

- Se demostró la utilidad de utilizar un protocolo estandarizado como SIP para la iniciación, modificación y finalización de sesiones interactivas de usuario en el intercambio de voz. Al implementar este protocolo en la conexión entre Issabel y Asterisk, se pudo establecer una comunicación eficiente y efectiva entre los servidores. Además, se observó que SIP permite la interoperabilidad entre diferentes sistemas de telefonía IP, lo que aumenta su versatilidad y utilidad en distintos contextos.

Referencias

1. J. López, *VoIP y Asterisk: redescubriendo la telefonía*. [Online]. Available: <https://books.google.com.ec/books?id=UI-fDwAAQBAJ>
2. R. G. Gil, “Seguridad en voip: Ataques, amenazas y riesgos,” *Universitat de València*, 2012.
3. TechnologyRadar, “Tipos de sistemas de telefonía ip y cuál elegir para tu empresa,” *TechnologyRadar*, 2021. [Online]. Available: <https://www.avanzada7.com/es/blog/sistemas-de-telefonía-ip-y-cual-elegir-para-tu-empresa>
4. J. A. Yépez Jiménez, “Hardening y alta disponibilidad en sistemas telefónicos basados en issabel pbx,” Ph.D. dissertation, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas . . . , 2018.
5. K. A. Defaz Parra and D. S. Salazar Barrionuevo, ““implementación de una central telefónica voz ip utilizando software libre issabel pbx y comunicaciones unificadas basado en asterisk en la constructora ma construcciones”.” B.S. thesis, Ecuador: Latacunga: Universidad Técnica de Cotopaxi (UTC)., 2020.