ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

PROTOTIPO DE UNA CENTRAL TELEFÓNICA IP BASADO EN HARDWARE Y SOFTWARE LIBRE

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

PELÁEZ FEIJOÓ DAVID HERNÁN

david.pelaez@epn.edu.ec

DIRECTOR: MSc. Christian Tipantuña Tenelema

christian.tipantuna@epn.edu.ec

CODIRECTOR: MSc. José Antonio Estrada Jiménez

jose.estrada@epn.edu.ec

Quito, agosto 2016

DECLARACIÓN

Yo, David Hernán Peláez Feijoó, declaro bajo juramento que el trabajo 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.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____

David Hernán Peláez Feijoó

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por David Hernán Peláez Feijoó, bajo nuestra supervisión.

Ing. Christian Tipantuña, MSc.

DIRECTOR DEL PROYECTO

Ing. José Antonio Estrada, MSc.

CODIRECTOR DEL PROYECTO

AGRADECIMIENTO

Agradezco a Dios por las bendiciones recibidas a lo largo de toda mi vida.

A mi padre, Hernán, quien fue mi primer maestro. Con sus enseñanzas y ejemplo ha sido mi modelo a seguir como persona, como profesional y como ser humano. Gracias por toda tu paciencia, tu amor y tus consejos que me han servido a lo largo de todo este camino.

A mi madre, Jenny, mujer fuerte y llena de virtudes quien con su amor, paciencia y entrega por sus hijos ha sabido guiarnos por la vida. Gracias madre por todo el sacrificio que has hecho para que yo siempre este bien.

Y a ambos por apoyarme en todas y cada una de las decisiones que he tomado, aun cuando no hayan estado de acuerdo conmigo siempre me permitieron elegir mi camino y es el que me ha llevado hasta donde estoy ahora.

A mi hermana Eva, por estar conmigo siempre que te he necesitado y darme tu apoyo y consejos en momentos difíciles.

A Melissa, mi hermana, mi amiga y mi cómplice, gracias por toda tu paciencia, tu cariño y las aventuras juntos.

A todos los amigos que han estado conmigo, algunos desde el primer día y que aún continúan, a los que conocí durante el camino y que siguen aquí, y a aquellos con los que compartimos ideales.

A Christian y José, mis directores en este proyecto de titulación, gracias por su apoyo en este proyecto, por incentivarme y motivarme a realizar investigación.

DEDICATORIA

A mis adorados padres, Hernán y Jenny, a mis hermanas Eva y Melissa, al pequeño Jack, a mis abuelos que ya no están aquí y a mi abuelito Neptalí que aún me acompaña.

David Hernán.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

En el presente proyecto se contempla la implementación del prototipo de una central telefónica IP basada en hardware y software libre diseñada para ser utilizada en ambientes de pequeñas oficinas o sitios con baja densidad poblacional. En la implementación se contempla el análisis del hardware a utilizar el cual debe cumplir las características de hardware libre y ser compatible con el software de IP-PBX; para el presente proyecto se utiliza una Raspberry Pi. En cuanto al software se utiliza la distribución de código abierto *uElastix* diseñada especialmente para trabajar en arquitecturas ARM como la que utiliza la Raspberry Pi y que brinda todas las funcionalidades de una PBX.

En la central telefónica IP se configuran las funcionalidades básicas de una PBX como establecimiento de llamadas, asignación de extensiones para usuarios, transferencia y captura de llamadas, intercomunicación entre extensiones, entre otras. Se utiliza una troncal SIP para comunicar a la central IP-PBX con un gateway de voz que permite realizar llamadas hacia la PSTN. El sistema también utiliza una interfaz inalámbrica WLAN para permitir la conexión de los usuarios al servidor de telefonía IP. Se implementan también algunas políticas de seguridad basadas en el propio Firewall IP-TABLES embebido en uElastix y la herramienta Fail2ban para evitar ataques de fuerza bruta hacia el sistema.

Adicionalmente la investigación contempla también la realización de un procedimiento de pruebas para verificar la funcionalidad del sistema IP-PBX, además de un test MOS (Mean Opinion Score) que verifica la calidad de voz del sistema de telefonía. Los resultados de todas estas pruebas se documentan y presentan en este trabajo.

PRESENTACIÓN

En diversos escenarios, como: zonas rurales, pequeñas localidades o en oficinas de pequeñas y medianas empresas, es muy común que para realizar llamadas locales se empleen centrales de tipo privado con el propósito de ahorrar costos derivado de la utilización de la PSTN, de manera general estas centrales privadas pueden tener una configuración más o menos amigable y su instalación y mantenimiento puede resultar más o menos complejo dependiendo el modelo, marca y la capacidad de la central telefónica. Adicionalmente la inclusión de uno u otro servicio puede estar limitado al hardware, a la compra de licencias o instalación de actualizaciones. Por ello, en el presente proyecto se plantea un prototipo de central de telefonía IP totalmente funcional, versátil, y sencillo de utilizar; un prototipo desarrollado utilizando enteramente plataformas de hardware y software libre.

En cuanto a las plataformas de hardware y software libre se puede mencionar que la filosofía de “libre” no es un tema novedoso tratado por un reducido grupo de personas. Más bien hoy en día constituye toda una filosofía y tendencia que ha ido en aumento. Esta filosofía actualmente se encuentra siendo adoptada a nivel nacional e internacional, tanto en el campo profesional como en el académico e investigativo. Uno de los ejemplos más claros de la evolución y de la adopción del hardware libre es la masificación de drones e impresoras 3D, los mismos que hace unos pocos años eran tecnologías no tan conocidas y de muy difícil acceso por su escasez y precio. La tendencia de hardware y software libre promueve el uso de tecnologías y soluciones innovadoras las cuales son muy flexibles, escalables, de bajo costo y que cada día son mejoradas. En la actualidad hay varios ejemplos del uso de plataformas de hardware libre además de varios proyectos y dispositivos que funcionan bajo esta idea.

Por otra parte, en el proyecto se ha considerado la utilización de una red Wi-Fi y de teléfonos inteligentes, esto debido a que al momento existe un gran auge de estas tecnologías y el uso de las mismas se encuentra bastante difundido. El hecho de tener en cuenta la utilización de un red inalámbrica como infraestructura de telecomunicaciones, viene dado por las características de este tipo de red, es decir, la facilidad de implementación y configuración, además de que este tipo de redes

brindan cierto grado de movilidad facilitando de esta manera el desplazamiento de los usuarios dentro del sistema de comunicaciones.

Combinando la tecnología inalámbrica, la telefonía IP y plataformas de hardware y software libre se tiene un sistema muy flexible que puede ser escalable y que puede estar abierto a modificaciones y mejoras.

El aspecto de la seguridad también ha sido considerado, debido a que habitualmente los servidores de comunicaciones sufren diversos tipos de ataque, el tipo de ataque a ser tratado en el presente proyecto es el de fuerza bruta, ya que este tipo de ataques consumen un gran ancho de banda, y ocupa parte del procesamiento del CPU, lo que podría ser perjudicial en el rendimiento del sistema. Proteger al sistema de intrusos y garantizar que el sistema funcione correctamente es indispensable en un sistema de comunicaciones.

Al ser el prototipo un sistema de bajo costo, versátil y escalable representa una solución bastante adecuada y accesible para entornos con baja densidad poblacional o empresas pequeñas. Así también el prototipo tiene un gran potencial en el ámbito de la educación ya que incluso a futuro puede ser implementado, utilizado y analizado en laboratorios de distinta índole como: telefonía IP, sistemas operativos, comunicaciones inalámbricas y demás disciplinas afines. En definitiva el prototipo planteado representa un gran aporte en el campo del hardware libre, ya que a partir de este prototipo, de la información recopilada y de los resultados obtenidos, es posible extender el análisis e implementación a sistemas cada vez más complejos y que se vean enmarcados en ámbito de investigación multidisciplinaria.

CAPÍTULO 1

1 FUNDAMENTOS DE TELEFONÍA IP

1.1 INTRODUCCIÓN A LA TELEFONÍA [1] [2]

La PSTN (*Public Switched Telephone Network*)[1] es una red de conmutación de circuitos que hace posible la comunicación con cualquier persona. Es una red en la que los terminales telefónicos se comunican con una central de conmutación mediante un canal compartido por el que viaja la voz en ambos sentidos.

Actualmente la PSTN sirve a los abonados utilizando circuitos digitales. Los sistemas de voz digital utilizan una tecnología conocida como PDH[2] la cual permite enviar varios canales telefónicos sobre un mismo medio usando técnicas de multiplexación por división de tiempo TDM[3] y equipos digitales de transmisión. Los sistemas PDH vienen en jerarquías y se tienen 3 estándares: el japonés, el americano y el europeo que también se utiliza en Sudamérica. El estándar E1 agrupa 30 canales digitales para llamadas de voz y dos canales para señalización. En lugares como EEUU, Canadá y Japón los proveedores utilizan circuitos T1 y J1 respectivamente, los cuales permiten 24 canales de voz [3].

Para los circuitos digitales se crearon dos protocolos de señalización:

a. **Señalización por canal asociado (CAS – Channel Associated Signaling):** La información de señalización se transmite en el mismo canal que los datos, utilizando algunos bits de los datos [4]. Se toma el octavo bit de cada canal de comunicación cada seis tramas reemplazándolo por información de señalización.

b. **Señalización por canal común (CCS – Channel Common Signaling):** La información de señalización de varios canales de datos se transporta por un canal de señalización compartido aparte del canal de datos [4].

---

[1]Red Telefónica Pública Conmutada
[2]Plesiochronous Digital Hierarchy o Jerarquía Digital Plesiócrona,
[3]Multiplexación por división de tiempo. Es una técnica para transmitir varias señales simultáneamente sobre un mismo enlace, dividiendo el dominio del tiempo en ranuras, y se asigna cada ranura de tiempo a cada señal.

La IUT-T ha definido un sistema de señalización por canal común conocido como SS7, cuyo principal objetivo es el de proporcionar una señalización normalizada internacionalmente y que sea de uso general.

1.1.1 ELEMENTOS DE LA PSTN

La PSTN actualmente es una red constituida por varios elementos que interactúan de forma conjunta. Estos elementos se muestran en la Figura 1.1.

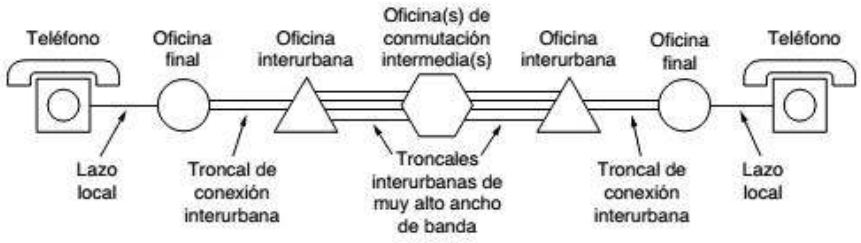


Figura 1.1 Elementos de la PSTN *[4]*

Algunos de estos elementos son: [4]

- **Terminales Telefónicos:** Convierten el audio en señales eléctricas. Permiten la conexión a la PSTN.
- **Lazo Local:** Es el enlace entre los abonados y el proveedor del servicio telefónico.
- **Oficinas de conmutación:** Es en donde las llamadas pasan de una línea troncal a otra.
    - **Oficina Central Local:** Es la oficina a la que se conectan los terminales telefónicos de los usuarios.
    - **Oficinas Tándem:** Son oficinas de conmutación que se conectan con la oficina central mediante troncales de conexión urbanas.
    - **Oficinas de Conmutación Intermedias:** Interconectan las centrales Tándem.
- **Línea Troncal:** Se encarga de la conexión entre las oficinas de conmutación.

1.1.2 PBX – PRIVATE BRANCH EXCHANGE

Es una central telefónica más pequeña que se utilizan de forma privada por empresas, instituciones educativas y en otros escenarios. Este sistema permite a los usuarios internos realizar llamadas dentro del entorno empresarial sin utilizar los recursos de la PSTN.

Si se desea realizar llamadas hacia la PSTN se utilizarían las líneas troncales que conectan a la empresa con dicha red.

1.2 TELEFONÍA IP [1] [2]

La telefonía IP se refiere al servicio telefónico realizado con la tecnología de VoIP[4]. Consiste en convertir la voz en paquetes y transmitirlos utilizando el protocolo IP por una red de datos. La transmisión de voz mediante el protocolo IP es susceptible a los retardos, lo cual es una desventaja. A pesar de esto, los protocolos de VoIP se encargan de que los paquetes de voz digitalizada lleguen a su destino a tiempo, negocian que codificador/decodificador de voz se va a utilizar y brindan seguridad en la transmisión de voz encriptando los datos.

1.2.1 VENTAJAS DE LA VOZ SOBRE IP [1] [2]

- **Reduce el costo de las comunicaciones:** La VoIP aprovecha la infraestructura de telecomunicaciones previamente instalada como LAN o WAN para establecer llamadas entre los usuarios aun cuando se encuentren en oficinas remotas ahorrando los costos de implementar nueva infraestructura para el servicio de telefonía.
- **Reduce el costo del cableado:** La implementación de VoIP reduce los costos del cableado ya que la misma infraestructura instalada para Internet puede ser utilizada para los sistemas de VoIP, en lugar de tener una conexión dedicada para cada servicio.
- **Simplifica las redes de Voz:** Debido a que las redes de datos conectan oficinas y los dispositivos móviles de los trabajadores, los sistemas de VoIP enrutan el tráfico de voz a través de la red de la empresa en lugar de salir a

[4] Voice over IP protocol

la PSTN. Esto también brinda un control centralizado de todos los dispositivos de voz ligados a la red de la empresa.

- **Permite la movilidad de los equipos de telefonía:** Los equipos para telefonía IP en su mayoría funcionan instantáneamente al conectarlos dentro de las oficinas y acceder a los recursos de red, lo cual permite traslados sin necesidad de hacer una re configuración de la red de voz. Adicionalmente los usuarios pueden acceder al servicio de telefonía IP desde cualquier lugar donde dispongan de una conexión a Internet con el hardware y software necesario.
- **IP Softphones:** La posibilidad de utilizar las redes de datos para la telefonía sobre IP permite utilizar aplicaciones de software que hacen la función de teléfonos IP. También existen aplicaciones para teléfonos inteligentes que permiten actuar a estos como softphones. El uso de las redes inalámbricas, por otro lado, permite a los usuarios llevar sus extensiones de telefonía con ellos a todo lugar dentro de la infraestructura de la empresa.
- **Comunicaciones Unificadas:** Todos los servicios de E-mail, mensajería instantánea, fax y correo de voz pueden ser integrados como un solo sistema y configurado dentro de una solo aplicación, de manera que facilite su control y manejo.
- **Aumento de la productividad:** Las extensiones de telefonía IP pueden funcionar en más de un dispositivo por usuario lo cual evita perder una llamada que podría ser importante.
- **Convergencia de los servicios:** Debido a que los servicios de voz, datos y video se combinan en la misma red, los usuarios pueden iniciar una llamada telefónica e interactuar con otras aplicaciones de voz, datos o video para mejorar la experiencia de comunicación.
- **Estándares abiertos compatibles:** Se puede conectar varios dispositivos de diferentes proveedores, lo que permite elegir el equipo que mejor se ajuste a las necesidades de la red independientemente de la marca.

1.2.2 COMPONENTES DE UN SISTEMA DE TELEFONÍA IP [5]

1.2.2.1 Terminales telefónicos

Son los dispositivos mediante los cuales un usuario se conecta al sistema de telefonía para poder realizar llamadas. Se pueden implementar en hardware (Teléfono IP) o software (Softphone). Un teléfono IP se conecta a la red de datos a través de una interfaz Ethernet para realizar llamadas, mientras que un softphone implementa muchas de las funciones de un teléfono IP y se instala en un computador o dispositivo móvil.

1.2.2.2 VoIP Gateway [6] [7]

Un Gateway para VoIP es un servidor y se sitúa en el extremo de la red y permite la conexión de del sistema de telefonía IP con la PSTN [8]. El *gateway* convierte el tráfico de telefonía en paquetes IP para luego ser transmitido por una red de datos.

Un *Gateway VoIP* se conecta a una central PBX mediante puertos analógicos o digitales denominados troncales. El *gateway* se conecta al sistema de telefonía tradicional en situaciones donde no se dispone de una troncal SIP hacia un proveedor VoIP y la aplicación requiere la conexión hacia la PSTN o para construir sistemas redundantes.

Un *gateway* de VoIP se puede utilizar de varias formas. Una de ellas es para convertir llamadas telefónicas de la PSTN entrantes en VoIP/SIP, de esta manera el *gateway* permite recibir y realizar llamadas en la red telefónica pública. Otra de las formas como se utiliza un *gateway de VoIP* es para conectar una central PBX tradicional con la red IP, de esta forma el *gateway* permite realizar llamadas de VoIP, que luego se pueden terminar a través de un proveedor de servicios VoIP.

1.2.2.3 Tarjetas telefónicas

En una central telefónica generalmente se tienen tarjetas telefónicas analógicas y digitales:

- ***Tarjetas analógicas:*** Se utilizan en los servidores de telefonía IP para permitir la conexión a la PSTN mediante interfaces FXO o para conectar terminales analógicos utilizando interfaces FXS.

- *Interfaz FXO (Foreing eXchange Office):* Permite a una central telefónica conectarse a la PSTN. Es el puerto que recibe la línea telefónica analógica [9].
- *Interfaz FXS (Foreing eXchange Suscriber):* Es el puerto que envía la línea analógica al abonado. Envía tono de marcado hacia un equipo terminal [9].

- ***Tarjetas digitales:*** Permiten la conexión entre un sistema de telefonía IP y la red digital RDSI. Son de dos tipos: BRI (Basic Rate Interface) y PRI (Primary Rate Interface) [8].

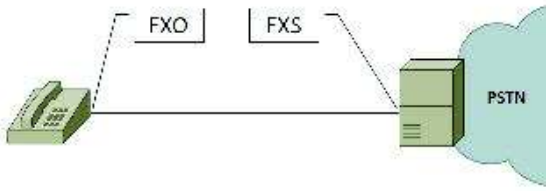


Figura 1.2 Conexión de puertos FXO y FXS *[61]*

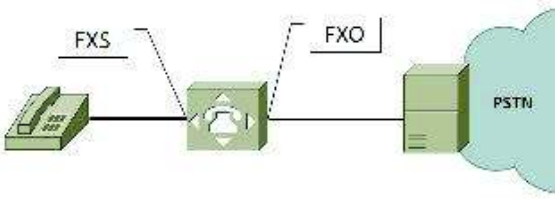


Figura 1.3 Conexión de puertos FXO y FXS con una PBX *[61]*

Cuando se trabaja con una PBX se deben conectar las líneas que suministra la oficina central a través de los puertos FXO de la central, en tanto que los teléfonos se conectan a los puertos FXS. Tal como se observa en la Figura 1.2 cuando la conexión entre el abonado telefónico y la PSTN es directa el puerto FXS del proveedor del servicio telefónico se conecta directamente con el puerto FXO del equipo telefónico del usuario. En la Figura 1.3 la conexión entre el usuario y el proveedor del servicio telefónico se realiza a través de una PBX. En este caso la PBX cuenta con puertos FXO que reciben la conexión que proviene de la PSTN y luego mediante los puertos FXS se conectan al puerto FXO del equipo telefónico del usuario final.

1.2.2.4 IP PBX

Una central telefónica IP es un sistema de comunicaciones que permite cursar tráfico telefónico transportado en paquetes, utilizando el protocolo IP para esta tarea. Una IP PBX se utiliza generalmente dentro de una empresa para brindar servicio de telefonía entre los usuarios, sin utilizar los recursos de la PSTN. También permite la comunicación de las extensiones telefónicas internas con la PSTN y enrutar llamadas provenientes de esta red hacia las extensiones internas.

1.3 PROTOCOLOS DE VoIP

Los protocolos de VoIP se clasifican es dos categorías, estas son: protocolos de señalización y protocolos de transporte.

1.3.1 PROTOCOLOS DE SEÑALIZACIÓN [1]

Los protocolos de señalización se encargan del registro de los terminales en el servidor de telefonía IP, localizar los terminales de usuario, iniciar y finalizar las llamadas, negociar algunos parámetros para el establecimiento de las llamadas.

Entre los protocolos de señalización más comúnmente usados están H.323, SIP, e IAX2.

1.3.1.1 Protocolo H.323 [10]

H.323 es una recomendación de la ITU que define la forma de proveer comunicaciones multimedia a través de redes basadas en conmutación de paquetes.

La recomendación describe los componentes de un sistema H.323, que incluye los terminales, gateways H.323, gatekeepers, unidades de control multipunto. La recomendación también define los mensajes de control y procedimientos para que estos componentes puedan comunicarse.

a. **Terminales H.323.-** Proporcionan a los usuarios la capacidad de tener mantener conferencias de audio o video en conferencias punto – punto o multipunto.
b. **Gateway H.323.-** Permiten la comunicación de una red H.323 con otro tipo de redes, proveyendo servicios de traducción de protocolos y control de llamadas entre los diferentes tipos de redes.

c. **Gatekeepers H.323.-** Proveen control de admisiones, y servicios de traducción de direcciones.
d. **Unidades de control Multipunto (MCU).-** Proveen el soporte para conferencias multipunto.

1.3.1.1.1 Protocolos que componen H.323

La pila de protocolos que componen la recomendación H.323 se muestra en la Tabla 1.1

Tabla 1.1 Protocolos que componen H.323

PROTOCOLO	CARACTERÍSTICA
H.225	Señalización necesaria para el establecimiento y control de la llamada entre dos puntos terminales H.323
RAS	Protocolo utilizado entre puntos finales y Gatekeepers que permite el registro, admisión, cambios de ancho de banda y funciones.
H.245	Señalización de control para negociar las capacidades y uso del canal.
H.235	Protocolo de seguridad. Provee un framework para seguridad en sistemas H.323
H.450	Protocolo funcional genérico para el soporte de servicios suplementarios en sistemas H.323
H.246	Interoperabilidad con redes de circuitos conmutados.
H.26x	Series ITU de códecs de video como H.261, H.263
G.7xx	Series ITU de códecs de audio como G.711, G.723. G.729

1.3.1.1.2 Etapas de una llamada H.323

El procedimiento para la realización de una llamada H.323 sigue las siguientes 5 etapas:

a. Descubrimiento y registro
b. Configuración de llamadas
c. Flujo de señalización de llamadas
d. Flujo de datos y flujo de control de datos
e. Terminación de llamada

1.3.1.2 Protocolo SIP [1]

El protocolo SIP es un protocolo de señalización no propietario estandarizado por la IETF (Internet Engineering Task Force) en el RFC 3261. SIP es un protocolo de la capa de aplicación del modelo TCP/IP, y tiene el propósito de iniciar sesiones multimedia como voz, video, mensajería instantánea.

SIP se concentra en el establecimiento, modificación y terminación de las sesiones, y para ello hace uso de algunas funcionas aportadas por otros protocolos. Utiliza RTP[5] para llevar los paquetes de voz o video que intercambian los participantes de una sesión SIP desde el origen a su destino. También se complementa con el protocolo SDP[6] para negociar los parámetros de la conexión multimedia como por ejemplo las direcciones IP a utilizar, los puertos y los códecs a utilizar durante la comunicación.

SIP es un protocolo transaccional que sigue un modelo cliente – servidor, es decir, el cliente realiza peticiones que el servidor atiende y se genera una respuesta que depende de la naturaleza y método de petición. El servidor responde a estas peticiones ya sea aceptando o rechazando las mismas en una serie de mensajes de respuesta, los cuales llevan un código de estado que indican si las peticiones fueron resueltas con éxito o si se produjeron errores durante la transacción.

[5] RTP: Real-time Transport Protocol
[6] SDP: Session Description Protocol

1.3.1.2.1 Componentes del protocolo SIP

SIP implementa algunas de sus funcionalidades para el establecimiento y finalización de sesiones multimedia utilizando varios componentes.

Existen dos elementos fundamentales que se describen a continuación:

a. ***User Agent (UA):*** Puede ser un cliente o un servidor y es la entidad lógica que inicia o responde a una transacción SIP.
   - *User Agent Client (UAC):* Inicia un pedido y acepta las respuestas. Típicamente en telefonía el UAC inicia una llamada.
   - *User Agent Server (UAS):* Es una entidad lógica que genera las respuestas a las peticiones SIP.

b. ***Servidores SIP:*** Existen de tres tipos:
   - *Proxy Server:* Es un componente intermediario entre el UAC y el UAS u otro Proxy. Retransmite solicitudes y decide a qué otro servidor debe remitir las solicitudes. El UAC envía peticiones al Proxy, el mismo que las envía al UAS. Es decir, los clientes nunca interactúan directamente con el servidor sino que lo hacen a través del Proxy. Existen dos clases de Proxy Servers: Statefull Proxy y Stateless Proxy.
     - Statefull Proxy: Es una entidad que mantiene las transacciones entre el cliente y el servidor durante el procesamiento de las peticiones.
     - Stateless Proxy: Es una entidad que no mantiene el estado de las transacciones entre el cliente y servidor durante el procesamiento de las peticiones. Un Stateless Proxy reenvía todas las peticiones y todas las respuestas que recibe.
   - *Redirect Server:* Envía respuestas de redirección a las peticiones que recibe. Reencamina las peticiones hacia el próximo servidor.
   - *Register Server:* Servidor (UAS) que acepta mensajes de registro de los usuarios y suministra un servicio de localización y traducción de direcciones en el dominio que controla.

1.3.1.2.2 Direccionamiento SIP

Una de las funciones de los servidores SIP es la localización de los usuarios y resolución de nombres. El direccionamiento estándar del protocolo SIP es similar a una dirección de correo electrónico que en este caso se denomina SIP URI1 y puede tener una de las siguientes formas:

- sip:usuario@dominio, donde dominio es un nombre de dominio completo o FQDN2
- sip:usuario@equipo, donde equipo es el nombre de la máquina.
- sip:número_telefónico@gateway, donde el gateway permite acceder al número de teléfono a través de la PSTN

sips:user@fqdn.com, la palabra sips indica una dirección SIP URI segura la cual utiliza el puerto 5061, que es el puerto asignado por defecto para el protocolo de transporte seguro TLS.

1.3.1.2.3 Señalización SIP

SIP utiliza un conjunto de mensajes que tienen una operación específica. Los UAC realizan las peticiones y los UAS regresan las respuestas a las peticiones de los clientes. Para esto SIP define dos tipos de mensajes, las solicitudes o métodos y las respuestas o códigos de estado. La forma general de un mensaje SIP consiste en una línea de inicio, cabecera del mensaje, luego una línea vacía que indica el final de las cabeceras, y por último el mensaje que es opcional.

a. **Cabeceras SIP:** Las cabeceras SIP se utilizan para transportar información necesaria a las entidades SIP. Una petición SIP debe contener estos campos de la cabecera: Via, From, To, CallID, Cseq, y Max-Forwards, los cuales contienen la información necesaria para el enrutamiento, identificación, y ordenamiento. A continuación se detallan estos campos:
    - **Vía:** Indica el transporte usado para el envío e identifica la ruta del request. Cada proxy añade una línea a este campo.
    - **From:** Indica la dirección IP del origen del mensaje de petición.
    - **To:** Indica la dirección IP del destinatario al que se entrega la petición.

En un mensaje SIP, la línea de inicio tiene el nombre de “request line”. Esta línea de inicio contiene la dirección sip (Request URI), y la versión del protocolo. En la Figura 1.4 se observa la request-line de un método INVITE de SIP. En el paquete seleccionado se puede identificar el tipo de petición (request) que se realiza. Dentro de la descripción del paquete se puede identificar el método (INVITE), la dirección URI – SIP que está envía el método de solicitud (sip:1012@192.168.0.10) y el protocolo de transporte utilizado (UDP1).

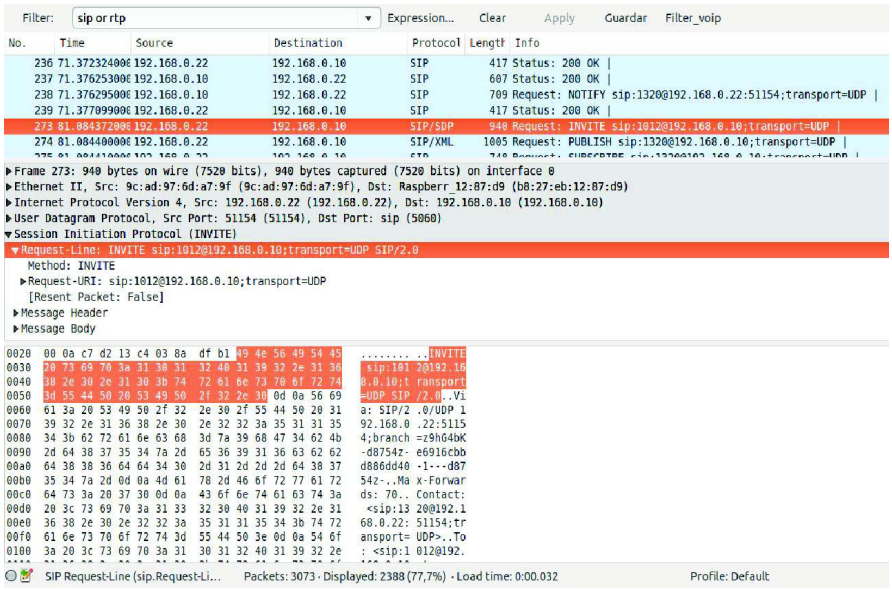


Figura 1.4 Contenido del paquete de una petición (Request Line)

Existen ciertos campos del paquete SIP que son obligatorios. Estos campos tienen un comportamiento especial dependiendo de las circunstancias. A continuación en la Tabla 1.1 se presenta una descripción de cada campo.

Tabla 1.2 Campos de cabecera de un paquete IP

CAMPOS DE CABECERAS	
Campo	**Descripción**
Via	Indica a los nodos a donde enviar el paquete SIP. Este parámetro posee el requerimiento de empezar con SIP/2.0 y el stack de comunicación.

CAMPOS DE CABECERAS	
Campo	**Descripción**
From	Esta es la identificación de quien inicia el proceso de petición.
To	Especifica el receptor de la petición. No necesariamente debe ser el nombre o URI del último receptor. También se utiliza para mostrar el nombre del usuario.
CSeq	Este campo provee un valor que ayuda a identificar las transacciones y ordenarlas.
Max-Forwards	Este valor limita el número de saltos que un paquete puede dar en su camino al destinatario. La recomendación dice que debe ser configurado en 70.
Contact	Este campo debe estar presente en la petición y contiene una dirección URI que debe coincidir con la usada anteriormente en la cabecera.
Allow	Ésta es una lista de los métodos que son soportados por el *User Agent* generando el mensaje.

b. **Métodos o solicitudes:** Un método es una función, y estas funciones están envueltas dentro de los mensajes. Los nodos SIP tienen reglas para cada método, las cuales pueden ser comunes para algunos de ellos. Las peticiones o solicitudes SIP se caracterizan por contener la línea inicial del mensaje llamada Request-Line, que es la que contiene el nombre del método empleado en ese momento, el identificador del destinatario de la petición o Request-URI y la versión del protocolo SIP que se esté utilizando.

   Los métodos existentes en SIP se describen en la Tabla 1.2.

Tabla 1.3 Listado de métodos dentro de SIP

Método	**Descripción**
Invite	Para iniciar una sesión. Invita a un usuario a iniciar una llamada.
Ack	Facilita el intercambio de mensajes confiables. Confirma una solicitud *Invite.*
Bye	Termina una conexión entre usuarios o termina una llamada.
Cancel	Cancela un requerimiento o búsqueda por un usuario.
Options	Solicita información acerca de las capacidades del servidor SIP.
Register	Registra una ubicación de usuario.
Info	Informa al servidor acerca del estado de señalización de la sesión.

En la Figura 1.5 se puede observar un ejemplo real de un mensaje usando el método **ACK**. En este paquete se puede identificar el tipo de solicitud (request) enviado hacia el host con la URI (sip:1012@192.168.0.10:5060). Dentro del paquete se puede identificar también las cabeceras **Vía, Contact, From, To, Cseq.**

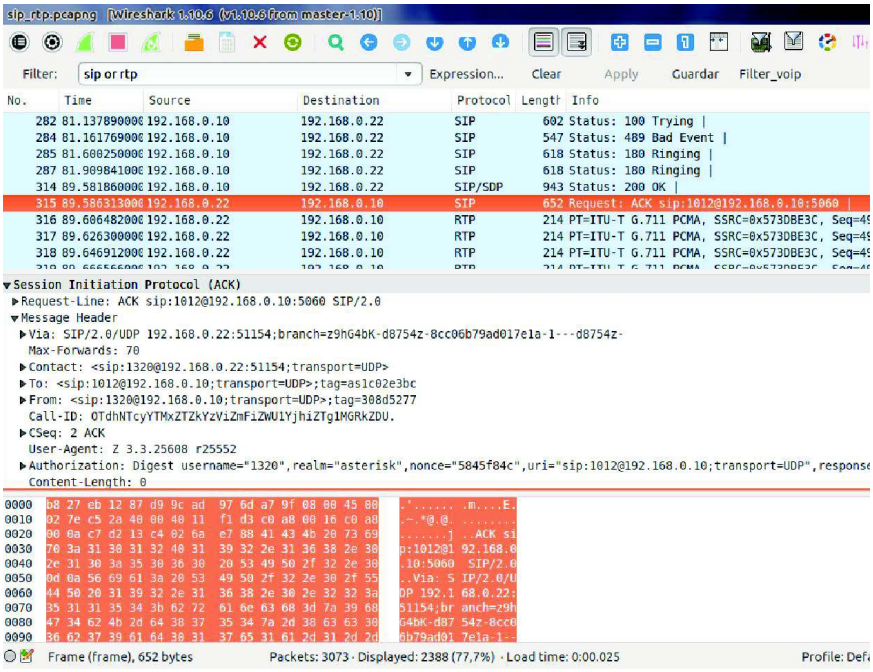


Figura 1.5 Contenido del método ACK

c. **Respuestas:** Las respuestas o códigos de estado de SIP son enviadas por el receptor luego de la recepción e interpretación de un mensaje de solicitud. Estos mensajes tienen una línea de estado del inicio llamada Status-Line la cual contiene la versión de protocolo SIP que se está utilizando, el código del estado para la transacción o Status-Code y una descripción o Reason-Phrase.

   Existe una gran cantidad de códigos que se pueden devolver como repuestas, los cuales indican el escenario en el que se encuentra la transacción entre el cliente y el servidor o entre clientes. El primer dígito define la clase de la respuesta enviada.

   La descripción de los diferentes códigos de respuesta se presenta de manera resumida en la Tabla 1.4.

Tabla 1.4 Códigos de respuesta en una transacción SIP

CÓDIGO	TIPO DE RESPUESTA	DESCRIPCIÓN	EJEMPLO
1xx	*Provisional*	Petición recibida, puede continuar con el proceso de petición.	100 Trying, 180 Ringin

CÓDIGO	TIPO DE RESPUESTA	DESCRIPCIÓN	EJEMPLO
2xx	*Success*	La acción fue recibida satisfactoriamente.	200 OK, 202 Accepted
3xx	*Redirection*	Nuevas medidas deben tomarse con el fin de completar la solicitud.	302 Moved Temporarily
4xx	*Client Error*	La petición tiene mala sintaxis o no pudo ser completada en el servidor.	404 Not Found
5xx	*Server Error*	El servidor no pudo completar una petición válida.	501 Not Implemented
6xx	*Global Failure*	La petición no pudo ser completada en ningún servidor.	603 Decline

En la Figura 1.6 se observa un ejemplo de un mensaje de respuesta 200 OK.

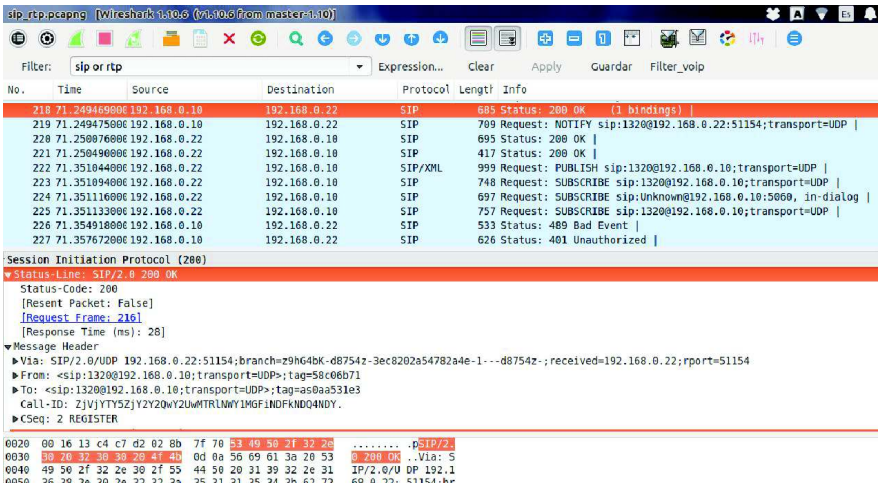


Figura 1.6 Mensaje de respuesta 200 OK

1.3.1.2.4 Proceso de comunicación con SIP

A continuación se detalla el proceso de una llamada utilizando el protocolo SIP.

En la Figura 1.7 se observa un diagrama de los mensajes SIP que se intercambian durante las transacciones entre el cliente y servidor. Estos mensajes constan de varias peticiones y respuestas.

Cuando se realizan llamadas entre dispositivos SIP, se sigue el siguiente proceso:

1. Como primer paso los usuarios se registran en el servidor para luego poder realizar llamadas hacia otros usuarios. Para iniciar el proceso de registro los dispositivos SIP envían un mensaje de petición *REGISTER* al servidor *IP-PBX ,* donde los campos *From* y *To* corresponden al usuario que va a registrarse, y el servidor consulta si el usuario puede ser autenticado y responde con un mensaje *OK* en caso de que el usuario tenga la credenciales correctas.
2. El siguiente paso corresponde al establecimiento de la sesión. El dispositivo SIP que genera la llamada, envía una solicitud al servidor *IP-PBX* mediante un mensaje *INVITE* para que establezca una conexión con el dispositivo SIP de destino, para esto interviene el protocolo *SDP* haciendo uso del puerto UDP 5060.
3. El servidor *IP-PBX* inmediatamente responde al dispositivo que inicia la sesión con un mensaje *TRAYING 100* para detener las retransmisiones y reenvia la petición al dispositivo SIP B (usuario B, ver Figura 1.7). El usuario B acepta la llamada, su dispositivo empieza a timbrar y envía un mensaje *RINGING 180* que el servidor reenvía al usuario A.

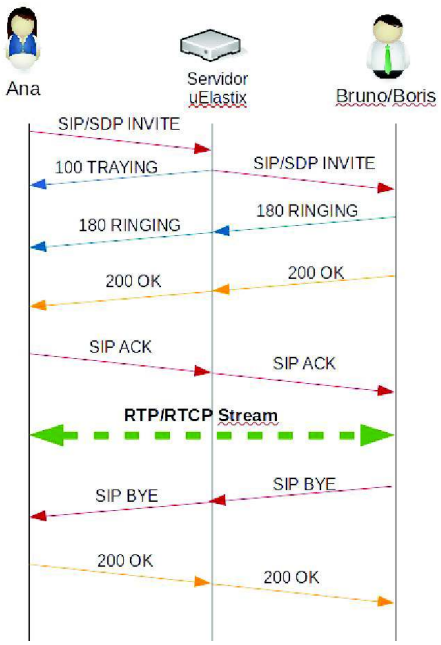


Figura 1.7 Flujo de una llamada SIP

4. El servidor *IP-PBX* establece la comunicación entre ambos usuarios y deja de intervenir en la comunicación. Todo este proceso se realiza por medio de paquetes SIP-SDP que llevan los parámetros establecidos en la negociación como puertos, direcciones, códecs, todo a través del puerto UDP 5060.

5. Los dos usuarios inician la transferencia de voz en ambos sentidos, mediante una conexión punto – punto, usando el protocolo RTP y abriendo un puerto UDP aleatorio que está entre el 10000 al 20000. Debe existir un puerto UDP abierto a ambos lados del canal de comunicación para que la transferencia de la voz se realice de manera bidireccional, de lo contrario la voz viajaría en un solo sentido provocando que solo se escuche en un lado del canal o que no se escuche nada.

6. Cuando la conversación termina se produce la finalización de la sesión. Para ello se realiza una única petición con un mensaje *BYE* que se envía al servidor y que posteriormente se reenvía al usuario B. Este contesta con un mensaje *OK* 200 para confirmar que ha recibido el mensaje correctamente.

   Ya que la comunicación es bidireccional se debe permitir el tráfico por los puertos UDP 10000 a 20000 para el tráfico entrante y saliente, así como el

puerto UDP/TCP 5060, por tal rezón resulta necesario habilitar estos puertos en el firewall para las redes donde exista telefonía IP.

1.3.1.2.5 Protocolo SDP

El protocolo SDP está pensado para la descripción de sesiones multimedia con el propósito de anunciar sesiones, invitara a las sesiones multimedia. Este protocolo está definido en el RFC 2327 y se utiliza en conjunto con el protocolo SIP siguiendo un modelo de oferta/respuesta para la negociación de las sesiones.

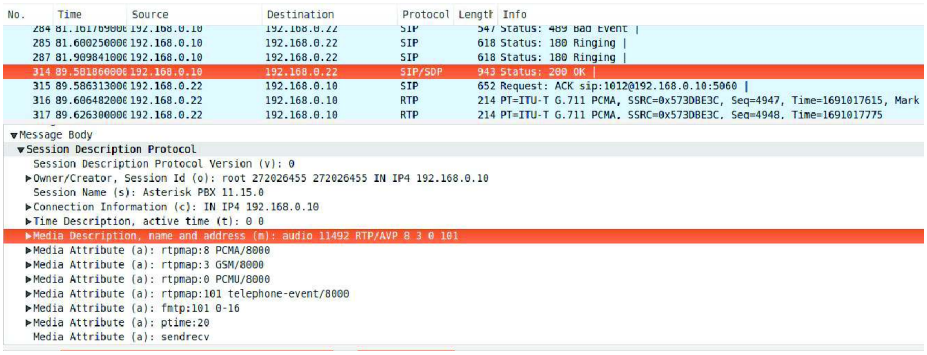


Figura 1.8 Contenido de un paquete SDP

SDP ofrece información como los códecs soportados, tipo de contenido multimedia, formato y direcciones de trasporte.

En la Figura 1.8 se ha seleccionado un paquete SDP dentro de cuyo contenido se observa los parámetros que se van a utilizar en una sesión de audio. Dentro del recuadro azul se encuentran los atributos de multimedia que se han negociado para esta sesión. Aquí se encuentra información como la versión del protocolo, el nombre de la sesión, el tipo de media (audio) y códec a utilizar (G.711 ley a).

1.3.1.3 Protocolo IAX [1]

Es un protocolo propietario desarrollado por Mark Spencer, creador de Asterisk1, para manejar conexiones VoIP entre servidores de telefonía IP, y dispositivos que manejen este mismo protocolo. Al momento se dispone de la segunda versión del protocolo denominado IAX2.

IAX utiliza el puerto UDP 4569, como único puerto para la comunicación entre puntos finales, tanto de voz como de la señalización. Además el protocolo permite también el Trunking (troncalización) de varios canales de audio en el mismo flujo de datos, lo que permite que un datagrama IP entregue información para más

llamadas sin crear retardo adicional, lo que resulta en un ahorro del ancho de banda ya que las cabeceras de los paquetes IP son las que más ocupan este recurso.

El protocolo IAX permite que las sesiones internas utilicen cualquier códec que transmita voz o video, brindando control y permitiendo la transmisión de flujos de datos multimedia sobre redes IP.

El principal objetivo de IAX es minimizar el ancho de banda utilizado para la transmisión de voz y video. Básicamente IAX se fundamenta en la multiplexación de la señalización y de los datos sobre un mismo puerto UDP entre dos sistemas.

A diferencia de SIP, IAX es un protocolo binario y no basado en texto. Esto permite reducir aún más el ancho de banda consumido por las transmisiones que se realicen con este protocolo ya que los mensajes generados tienen cabeceras reducidas.

Entre las ventajas que presenta este protocolo sobre otros se tiene:

- Menor consumo de ancho de banda.
- Soluciona mejor los problemas de NAT. Tanto la información de señalización como los datos viajan conjuntamente y por tanto lo hace menos proclive a problemas de NAT y le permite pasar los routers y firewalls de manera más sencilla.
- Los mensajes (datos) pasan más fácilmente a través de los firewalls, por la misma razón explicada en el punto anterior.

1.3.1.3.1 Fases de una llamada IAX

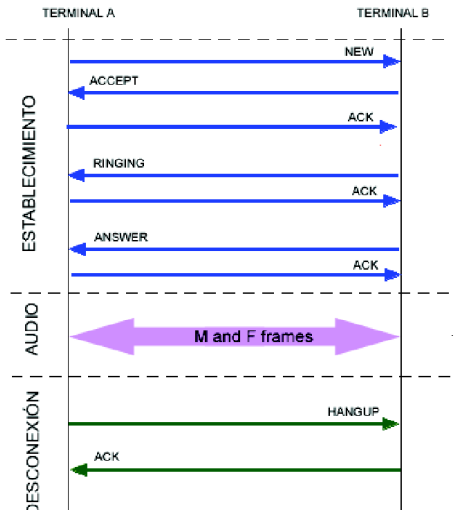


Figura 1.9 Fases de una llamada IAX *[11]*

Durante una llamada utilizando IAX se tienen tres fases: el establecimiento de las llamadas, el flujo de datos (audio) y la desconexión de la llamada. En la Figura 1.9 se presenta un esquema del proceso para establecer una llamada con el protocolo IAX. Estas fases se describen a continuación [11].

1. **Establecimiento de la llamada:** El dispositivo A inicia la conexión enviando un mensaje “NEW” al dispositivo B, el cual responde con un mensaje de “ACCEPT”, como se observa en la Figura 1.8. Luego el dispositivo A envía un “ACK” confirmando la recepción del mensaje. Enseguida el terminal que recibe la llamada envía las señales de “RINGING” y el dispositivo que inicia la llamada contesta con un mensaje “ACK” que confirma la recepción del mensaje. Finalmente el dispositivo llamado acepta la llamada con un mensaje “ANSWER” y el llamante confirma ese mensaje.
2. **Flujo de datos o llamada en curso:** En esta fase se envían dos tipos de tramas, las tramas M y las tramas F, como se observa en la Figura 1.8 las cuales son de color morado. Las tramas M son mini-tramas que contienen una cabecera de 4 bytes. En este caso no es necesario que se responda a estas tramas y cuando alguna se pierde simplemente se descartan.

Una trama M consta de los siguientes campos: *bit F*, *Source Call number*, *Timestamp.* En la figura 1.10 se puede apreciar el formato de una trama M.

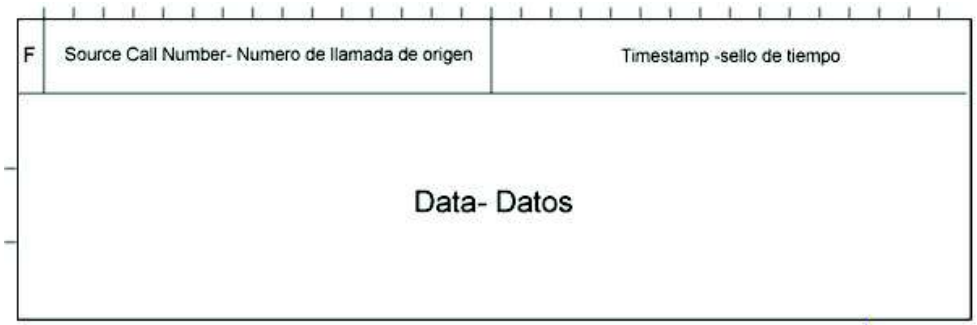


Figura 1.10 Trama M de IAX *[11]*

Aquí el campo del bit F se establece en 0, mientras el *timestamp* se compone de 16 bits para aligerar la cabecera. Los clientes pueden tener un *timestamp* de 32 bits si lo desean y pueden enviar una trama F para la sincronización.

Las tramas F también conocidas como *Full Frames* a diferencia de las tramas M deben ser respondidas obligadamente. Por lo tanto cuando un usuario envía a otro una trama F el receptor debe contestar al transmisor que ha recibido el mensaje. Los campos de una trama F los podemos apreciar en la Figura 1.10.

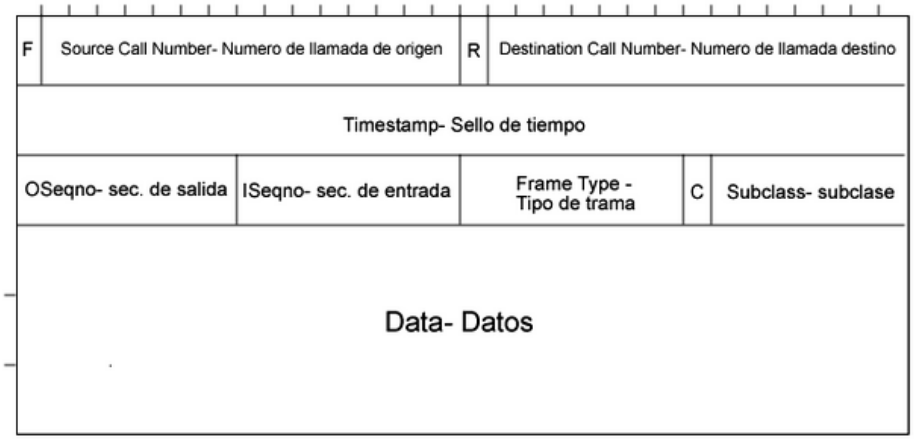


Figura 1.11 Trama F de IAX [11]

La descripción de cada campo es la siguiente:

F: Este bit sirve como indicador del tipo de trama que se ha enviado. Si es una trama F este bit se establece en 1.

Source Call Number: Este campo de 15 bits identifica cual es el origen de la conversación ya que pueden existir varias conversaciones multiplexadas en la misma línea.

R: Este es el bit de retransmisión. Se establece a 1 cuando la trama es retransmitida.

Destination call number: Similar al número de origen, identifica el destino al que se está llamando.

Timestamp: Campo utilizado para marcar el tiempo de cada paquete.

Oseqno: 8 bits que identifican el número de secuencia de salida de los paquetes. Empieza por cero y se va incrementando con cada mensaje.

Iseqno: 8 bits que identifican el número de secuencia de entrada de los paquetes.

Frame Type: Indica el tipo de trama que se está transmitiendo. Pueden ser de tipo F o tipo M

C: Establecido en cero, indica que el campo subclase debe tomarse como 7 bits. Si se establece en uno indica que el campo subclase se debe tomar como una palabra de 14 bits.

Subclass: Subclase del mensaje.

Data: Datos que son enviados en formato binario.

3. Finalización de la llamada: La finalización de llamadas consiste en el envío de un mensaje de "*hangup*", luego se envía la confirmación correspondiente de dicho mensaje y se termina la llamada.

1.3.2 PROTOCOLOS DE TRASPORTE

Los protocolos de transporte son los responsables de llevar los paquetes de voz generados por los códecs[7]. Una vez que la llamada ha sido establecida, los paquetes de voz son enviados de extremo a extremo usando el protocolo RTP (Real Time Transport Protocol)[8].

[7] Códec: Abstracción de codificador/decodificador, en VoIP es un dispositivo o software que utiliza algún algoritmo para comprimir o descomprimir las señales de voz digitalizadas.
[8] RTP: Real Time Transport Protocol.

1.3.2.1 Protocolo RTP [2] [12]

El protocolo RTP (Real-time transport protocol) provee funciones de trasporte de red extremo a extremo pensado para aplicaciones que trasmiten datos en tiempo real como audio o video. RTP no se preocupa de la reserva de recursos y no garantiza calidad de servicio durante la transmisión de datos en tiempo real. Se utiliza un protocolo de control (RTCP) que permite el monitoreo de los datos enviados y funciones de identificación.

RTP opera en la capa de transporte del modelo TCP/IP sobre UDP. El protocolo UDP provee los números de puerto y cabeceras de checksum, mientras que RTP añade estampas de tiempo y números de secuencia a las cabeceras. Esto permite al dispositivo remoto organizar los paquetes que recibe.

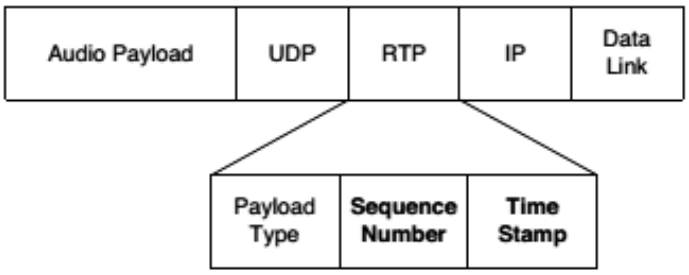


Figura 1.12 Información de cabecera de RTP *[1]*

La Figura 1.12 muestra la información de la cabecera contenida en un paquete RTP. El campo "*Payload Type*" se utiliza para identificar el tipo de datos que se está enviando, estos pueden ser audio o video.

Cuando dos dispositivos intentan establecer una sesión para transmisión de audio, RTP inicia su trabajo y elige un puerto UDP para cada transmisión RTP de una sola vía. Este puerto se elige aleatoriamente entre 10000 y 20000. Cuando la conversación es de dos vías, los dispositivos establecen transmisiones duales de RTP para cada dirección. El flujo de audio permanece en el puerto elegido inicialmente mientras dure la sesión de audio; todo este procedimiento se puede observar en la Figura 1.13.

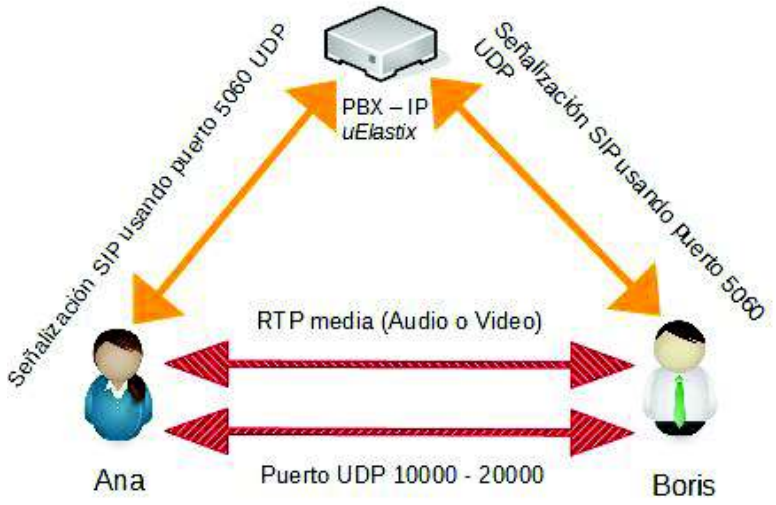


Figura 1.13 Funcionamiento de RTP en una sesión SIP

1.4 CÓDECS DE VOZ

En un sistema de VoIP la voz debe ser convertida en paquetes que puedan ser transportados a través de la red, para esto las señales de voz deben ser digitalizadas. Este proceso involucra 3 fases, el muestreo, cuantificación y codificación. Para este proceso se utilizan los denominados códecs de voz o vocoder. Los códecs de voz permiten realizar la conversión de una señal de voz de analógica a digital, pero además de esto también se encargan de la compresión de los paquetes de voz digitalizada con algún algoritmo para disminuir el la tasa de transmisión de bits.

Algunos de los códecs de voz más utilizados en la VoIP son: G.711, G.729, G.722, iLBC, y GSM.

1.4.1 G.711 [13] [14]

Es uno de los códecs más utilizados en la telefonía IP. El estándar utiliza dos métodos de compansión (compresión/expansión), estos son: *ley u* que se utiliza en Norteamérica y *ley A* que se utiliza en Europa y el resto de América.

Este códec está descrito en la recomendación ITU-T G.711, donde se describen algunas características para la codificación de señales de voz. La velocidad de

muestreo se establece en 8000 muestras por segundo y la utilización de 8 bits por muestra, por lo cual este codificador proporciona un flujo de datos de 64 Kbps [15].

Las dos leyes de compansión utilizan una curva basada en perfiles logarítmicos para el proceso de compresión mediante segmentos. En la Figura 1.14 se observa una representación gráfica del proceso de codificación de una señal utilizando la ley A. Esta curva logarítmica consta de 13 segmentos que corresponden a cada intervalo de cuantización de la señal.

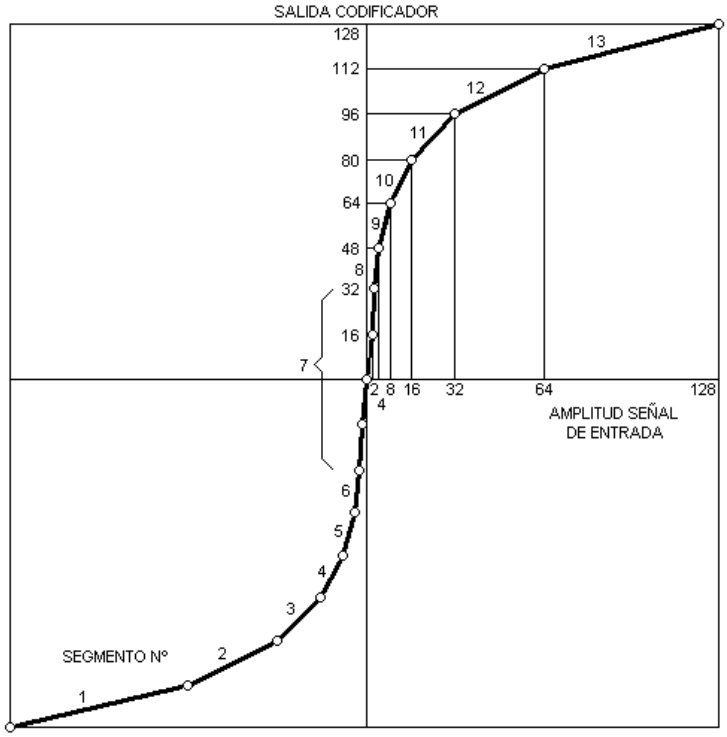


Figura 1.14 Curva característica de compresión de 13 segmentos

1.4.2 G.729

La recomendación ITU-T G.729 describe la codificación de señales de voz a 8Kbps utilizando un algoritmo denominado (CS-ACELP) *Conjugate-Structure Algebraic-Code-Excited Linear-Prediction* que es muy utilizado por algunos códecs [8].

El códec G.729 utiliza una tasa de bits de 8 Kbps pero existen extensiones que suministran también tasas de 6.4 kbps y 11.8 kbps para peor o mejor calidad en la conversación respectivamente. Este códec posee alta compresión y baja tasa de

transmisión de bits, pese a esto no deteriora la calidad de la voz lo que lo hace muy atractivo para implementarse en sistemas VoIP.

Una descripción sencilla del procedimiento que utiliza G-729 para la compresión del audio es la siguiente: Consiste en enviar una muestra de audio una vez y simplemente decirle al dispositivo remoto que reproduzca este sonido repetidamente durante un cierto periodo de tiempo. A este proceso se denomina construir un "*codebook*" o libro de códigos de los sonidos que se están transmitiendo entre los dispositivos finales. Realizando este procedimiento, G.729 es capaz de reducir la tasa de bits a 8 Kbps.

1.4.3 GSM [16] [17]

Este códec viene definido por el estándar de comunicaciones móviles del mismo nombre (Global System for Mobile Comunications). El códec GSM (full rate) opera a una velocidad de 13 Kbps y utiliza el algoritmo *Regular Pulse Excited*.

El códec GSM proporciona buena calidad de voz, pero se considera de menor grado a la que ofrece G.729.

La desventaja de este códec radica en que si se trabaja con conexiones de mala calidad en cuanto al tiempo de respuesta o jitter, la voz puede llegar a distorsionarse tanto hasta ser incomprensible.

1.4.4 iLBC [18]

iLBC es el algoritmo para codificar señales de voz con una velocidad de muestreo de 8 KHz. El códec iLBC utiliza un algoritmo de codificación de bloques independientes y predicción lineal y tiene soporte para dos longitudes de tramas: de 20 ms a 15.2 Kbps produciendo 304 bits por bloque y 30 ms a 13.33 Kbps produciendo 400 bits por bloque. Los bloques codificados deben ser encapsulados dentro de algún protocolo de transporte, que por lo general es RTP.

El algoritmo descrito anteriormente da como resultado un sistema de codificación de voz con una respuesta controlada para la pérdida de paquetes similar a lo que se conoce como Modulación de pulsos codificados (PCM – Pulse Code Modulation) con *Encubrimiento de pérdida de paquetes (PLC – Packet Loss Covered)*, tal como el obtenido en el estándar G.711. iLBC maneja la pérdida de paquetes con una

degradación muy pequeña de la calidad de la voz. La pérdida de paquetes generalmente ocurre en conexiones con retardo.

1.4.5 G.722 [19]

Esta recomendación describe las características de un códec de banda ancha (50 a 7000 Hz) el cual es muy utilizado para una gran variedad de aplicaciones de alta calidad de voz. El codificador utiliza una técnica de modulación llamada Modulación de pulsos codificados diferencial de sub-banda adaptativa (SB-ADPCM) con una velocidad de transmisión de 64 Kbps. En esta técnica de modulación el ancho de banda se divide en dos sub-bandas (una alta y una baja) y las señales en cada banda son codificadas usando el algoritmo ADPCM. El sistema tiene tres velocidades (modos) básicas de operación para un mismo codificador de 7 KHz. Estas velocidades son 64 Kbps, 56 Kbps y 48 Kbps.

Este códec es muy útil para aplicaciones de VoIP principalmente en redes de área local con alta tasa de transmisión de bits. Adicionalmente el códec ofrece gran calidad de voz sobre otros códecs de banda estrecha como G.711, aunque consume mayores recursos del procesador.

En la Tabla 1.5 se muestra una comparativa de los códecs descritos en las secciones anteriores.

Tabla 1.5 Cuadro comparativo de códecs de voz

CODEC	ALGORITMO	VELOCIDAD DE TRANSMISIÓN
G.711	Pulse Code Modulation (PCM)	64 Kbps
G.729	CS-ACELP	8 Kbps
GSM	RPE-LTP	13 Kbps
iLBC	PLC – Packet Lose Covered	15.2 Kbps / 13.3 Kbps
G.722	SB-ADPCM	64 Kbps

1.5 HARDWARE LIBRE

Cuando se habla de “hardware libre” se hace referencia a una plataforma de hardware que obedece a los requisitos del “Open Hardware Specification Program”, estos requisitos especifican que la documentación acerca del dispositivo debe estar disponible, y de libre acceso de manera que cualquier usuario pueda utilizar esta

información. Un dispositivo es considerado de hardware libre si sus especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita, recordando que libre no es sinónimo de gratis. Al ser el diseño del hardware de tipo público, cualquier persona lo puede estudiar, modificar, distribuir, materializar y vender, tanto el original como otras versiones basadas en el diseño original [20].

1.5.1 PLATAFORMAS DE HARDWARE LIBRE

Entre las plataformas de hardware libre se van a analizar 3, Asiri, Raspberry Pi y PICOSAM-9G45. Estas plataformas se eligen ya que son compatibles con el software de IP-PBX que se analiza más adelante [21]. Existen otras plataformas similares a estas como la Beagle Bone o Intel Galileo que cumplen con la misma filosofía que las anteriores en cuanto ser hardware libre, pero con la diferencia que dentro de la información previa que se ha revisado no se encuentra referencias que indiquen que sean compatibles con proyectos de VoIP.

1.5.1.1 ASIRI [22]

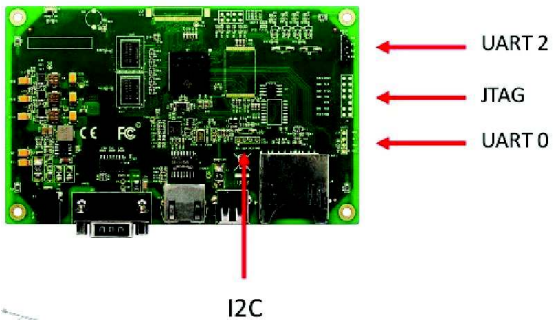


Figura 1.15 Placa ASIRI y periféricos *[23]*

En la Figura 1.15 se muestra la placa ASIRI y la posición de algunos de sus periféricos.

La placa ASIRI está basada en tecnología ARM[9] y es desarrollada por la compañía ecuatoriana PaloSanto Solutions. La placa es fabricada en alianza con la empresa china Openvox [22].

[9] Advance RISC Machine: Es una arquitectura RISC (Reduced Instruction Set Computer – Ordenador con Conjunto Reducido de Instrucciones) desarrollada por ARM Holdings.

Las características técnicas de esta placa se verán en la Tabla 1.6 [23].

1.5.1.2 RASPBERRY PI [24] [25]

La Raspberry Pi es un micro-computador desarrollado por la fundación Raspberry Pi, diseñado para ser de bajo costo, de tamaño reducido, contenido en una sola placa y con el objetivo de enseñar a estudiantes de escuelas, colegios y universidades a programar. Utiliza un procesador Broadcom SoC (System on a Chip) de arquitectura ARM.

Las características técnicas de esta placa se verán en la Tabla 1.6. En la Figura 1.16 se muestra un diagrama de la Raspberry Pi y la ubicación de sus periféricos.

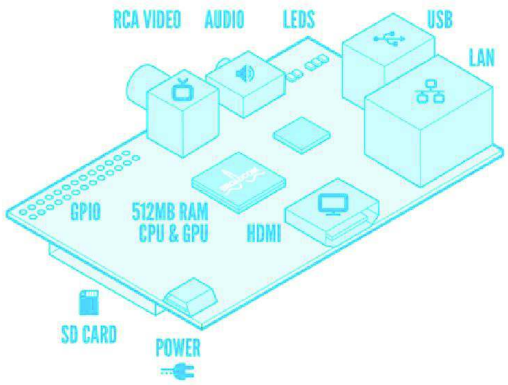


Figura 1.16 Raspberry Pi y periféricos *[25]*

1.5.1.3 PICOSAM-9G45 [26]

La pico-sam9g45 es una tarjeta de bajo costo, bajo consumo de energía, que permite añadir muchas interfaces de comunicación lo que la hace bastante versátil. Esta tarjeta es diseñada y fabricada por la empresa mini-box [27].

En la Figura 1.17 se observa la estructura física de esta tarjeta. Las características técnicas de esta placa se verán en la Tabla 1.6.

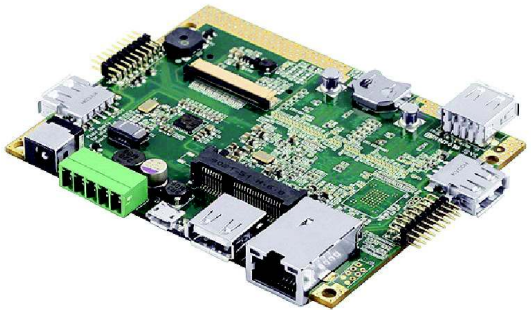


Figura 1.17 Pico-SAM9G45

1.5.1.4 Análisis de plataformas de hardware para IP PBX

Luego de describir las características y especificaciones de estas tres tarjetas de hardware libre se puede realizar un resumen comparativo de ellas y seleccionar la que se va a utilizar en el proyecto.

Tabla 1.6 Cuadro comparativo de las plataformas de hardware libre

ESPECIFICACIONES	ASIRI	RASPBERRY PI	PICOSAM-9G45
CPU	ARM926EJ-S @ 456 MHz	ARM1176JZF-S @ 700 MHz	ARM926EJ-S @ 400 MHz
Memoria SDRAM	256 MB	512 MB	256 MB
Puertos USB	1 (2.0)	2 (2.0)	4 (2.0)
Entradas de Video	X	MIPI CSI para módulo de cámara	X
Salidas de video	X	Conector RCA (PAL y NTSC), HDMI, DSI para panel LCD	Soporte nativo para LCD de 480 x 272
Salidas de audio	X	Conector 3.5 mm, HDMI	Buzzer
Almacenamiento	Variable via tarjeta SD	Variable via tarjeta SD	Variable via tarjeta SD
Conectividad de red	10/100 Ethernet (RJ45)	10/100 Ethernet (RJ45)	10/100 Ethernet (RJ45)

ESPECIFICACIONES	ASIRI	RASPBERRY PI	PICOSAM-9G45
Perifericos de bajo nivel	Pórtico Serial	32 pines GPIO con soporte para SPI, I2C, UART	18 pines GPIO con soporte para SPI, Serial
Fuente de alimentación	12 V	5 V vía conector micro USB	5 V vía micro USB, 6 – 60 V vía conector DC
Sistema Operativo	Linux	Linux: Raspbian, Fedora, Arch Linux, RISC OS	Linux / Android
Slot para tarjeta SIM	X	X	SI

De acuerdo a la información presentada en la Tabla 1.6 se observa que todas son opciones viables para el proyecto. El criterio determinante sobre cual se va a elegir dependerá de la capacidad de procesamiento, memoria RAM y almacenamiento disponible. En cuanto al procesador se observa que la placa ASIRI y la PICOSAM utilizan el mismo procesador, la diferencia está en la velocidad de trabajo, siendo de 456 MHz en la ASIRI y 400 MHz en la PICOSAM-9G45. Aparentemente poseen igual memoria RAM y la capacidad de almacenamiento viene dada por la tarjeta SD externa que se vaya a utilizar. Por otro lado la Raspberry Pi posee un mejor procesador y mayor memoria RAM que las placas anteriores, lo cual se traduce en mayor capacidad de procesamiento para las llamadas y el sistema, por lo que esta plataforma presente mejores características para la implementación del presente proyecto.

1.6 PLATAFORMAS DE SOFTWARE PARA IP – PBX

1.6.1 ELASTIX [1]

Elastix es una herramienta empresarial de código abierto para comunicaciones unificadas. Incluye un conjunto de paquetes de software libre que se distribuyen juntos como uno solo e incluye el instalador, el sistema operativo y la licencia GlPv2. Todas las distribuciones de Elastix son completas y sin limitaciones en sus características. Los usuarios tienen la libertad de hacer uso comercial, personal, y

están sujetos a las condiciones descritas en la licencia. La estructura de Elastix está basada en cinco programas que son Asterisk, Hylafax, Postfix, Openfire y Vtriger los cuales brindan las funcionalidades de PBX, Fax, E-mail, Mensajería instantánea, y Call Center respectivamente. El sistema operativo está basado en la distribución de Linux para servidores CentOS.

Entre las características más importantes de Elastix se tiene:

- Correo de voz.
- Traslado de Fax a E-mail.
- Soporte para Softphones.
- Interfaz de configuración web.
- Sala de conferencias virtuales.
- Grabación de llamadas.
- Least Cost Routing.
- Interconexión entre PBXs.
- Identificación de llamadas.
- CRM
- Reporte avanzado de llamadas

En la figura 1.18 se muestran los diferentes módulos y capas que componen Elastix y la relación entre ellos.

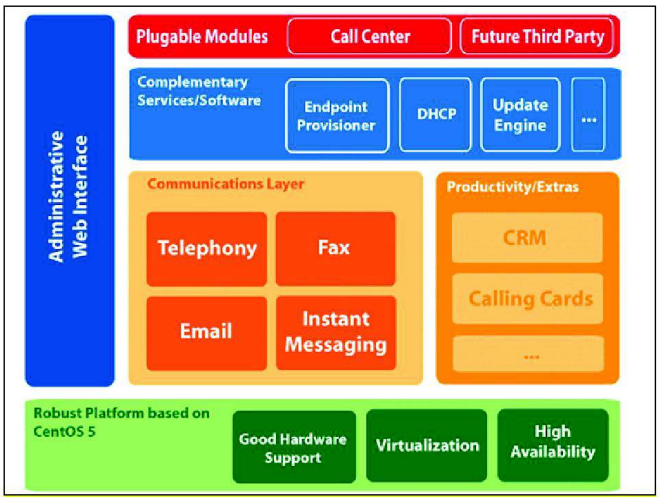


Figura 1.18 Componentes de Elastix *[1]*

1.6.1.1 Micro-Elastix [21] [28]

Micro – Elastix o uElastix es una distribución de Elastix optimizada para plataformas ARM que fue lanzada en el evento VoIP2Day + ElastixWorld en 2012. Debido a que se ha buscado la optimización de recursos y espacio, esta distribución no cuenta con los módulos de mensajería instantánea (Openfire) y call center (Vtiger) [28].

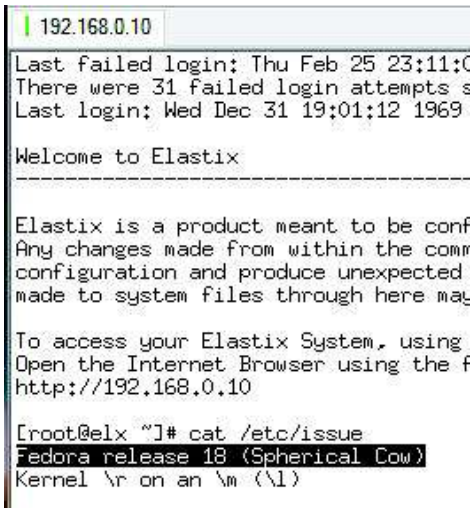


Figura 1.19 Versión del sistema Operativo de uElastix

Esta distribución está basada en el sistema operativo Fedora 18 (Spherical Cow) [29], como se observa en la captura de pantalla que se muestra en la Figura 1.19 del comando "`cat /etc/issue`" desde la consola de *uElastix.* El sistema operativo requiere como mínimo un procesador a 400 MHz y 1 GB de memoria RAM [29].

1.6.2 TRIXBOX [30] [31]

TrixBox es una distribución del software Asterisk PBX, diseñado para usuarios novatos en Asterisk, de modo que no se necesitan conocimientos profundos acerca de Linux o telefonía IP para utilizarlo. Este software está dimensionado para su uso en empresas pequeñas o familiares, con no más de una docena de usuarios. No está pensado como una plataforma para uso corporativo o empresarial.

Esta distribución se puede descargar de la página oficial del producto, y es distribuida en formato ISO[10] donde se incluye el sistema operativo y los paquetes

[10] El formato ISO, es un archivo informático que almacena una imagen exacta de un sistema de archivos o ficheros de un disco óptico, CD (Disco Compacto) o DVD (Disco Versatil Digital). Su uso más común es para la distribución de sistemas operativos como por ejemplo Linux, BSD y como Live CD.

de software y herramientas necesarias. Algunos de los componentes de TrixBox se describen a continuación:

- **Linux CentOS:** Es la distribución de Linux sobre la que está basado TrixBox.
- **Asterisk:** Es el núcleo para el soporte de telefonía IP.
- **FreePBX:** Es la interfaz gráfica vía Web para la configuración de Asterisk.
- **Flash Operator Panel (FOP):** Es una aplicación que presenta una interfaz donde el operador de la central PBX puede revisar el estado del sistema de telefonía.
- **Web Meet Me Control:** Administrador de salas de conferencias.
- **A2Biling:** Es una plataforma para llamadas prepagadas compatible con Asterisk.
- **SugarCRM:** Un CRM vía Web.

Una de las limitaciones de TrixBox es el hecho de que ha sido diseñado para trabajar en una sola máquina, es decir que no es escalable, por lo que si el sistema de telefonía empieza a crecer va a ser difícil migrar el sistema de una máquina a otra o realizar un trabajo en conjunto.

Entre las ventajas de este sistema se tiene que es muy fácil de instalar, solamente se necesita grabar la imagen ISO del sistema en un CD-ROM, y luego instalar el sistema en la maquina huésped.

Entre los requerimientos mínimos de hardware que este sistema requiere para hacer una prueba o prototipo se tiene: 384 MB de RAM y un disco de 10 GB para el almacenamiento. Si por el contrario se piensa en un sistema empresarial, el fabricante del software recomienda utilizar una máquina con un procesador de al menos 2 GHz con una memoria RAM de 1 GB y un disco de 100 GB para el almacenamiento, ya que de no cumplir con estos requerimientos se corre el riesgo de tener un bajo desempeño, baja calidad de audio en las llamadas, y una experiencia pobre por parte de los usuarios.

1.6.3 KOLAB [32] [33]

Kolab Groupware Solution es un paquete de software libre que funciona como servidor de colaboración con capacidades para e-mail, mensajería instantánea, VoIP y conferencia, calendario, notas, y tareas compartidas.

El diseño de *Kolab* trae consigo las siguientes funcionalidades:

- Base de datos centralizada para autenticación y autorización mediante el uso de LDAP.
- Mail Exchange para el intercambio de correo electrónico de tipo IMAP y POP3; y cifrado de correo usando PGP[11]
- Posee una capa de almacenamiento de datos y un protocolo de acceso primario.
- Filtrado de correo mediante reglas ACLs[12]
- *Anti-SPAM* y *anti-Virus* integrado.
- Posibilidad de uso off-line de correo electrónico si se usa KDE Kontact, Thunderbird SyncKolab o Microsoft outlook.
- Múltiples herramientas de colaboración como: Mensajería instantánea, VoIP y conferencia, Videoconferencia.

Para proveer la funcionalidad de todos estos servicios Kolab tiene el soporte de algunos componentes de software que se integran de una forma modular fácil de manejar y de cambiar. Estos componentes son:

- **OpenLDAP:** Utilizado como la base de datos donde se guardan las credenciales de los usuarios para la auntenticación en el servidor.
- **Cyrus:** Provee el acceso al correo electrónico de tipo IMAP y POP3.
- **SASL:** Es la capa de seguridad y auntenticación que hace uso de OpenLDAP como el repositorio de la información de auntenticación.
- **RoundCube:** Es una solución para correo electrónico con una interfaz web amigable fácil de instalar y configurar.
- **Postfix:** Es el MTA (mail transfer agent) para el servidor de Kolab.

---

[11] Pretty Good Privacy: Método de encriptación de datos para correo electrónico.
[12] Acces Control List: Lista de control de acceso, mecanismo que permite controlar el flujo de tráfico en una red IP.

- **APACHE:** Brinda el entorno de administración web del sistema.
- **OpenSSL:** Brinda la encriptación de datos.
- **DOVECOT:** Funciona como servidor de correo electrónico para sistemas Linux que provee de seguridad como su principal objetivo.

1.6.4 FREESWITCH [34]

Freeswitch es una solución de software de código abierto diseñada para enrutar e interconectar sistemas de comunicaciones con distintos tipos de datos multimedia como voz, video, o mensajes de texto. Freeswitch también brinda una plataforma muy estable de telefonía IP, dentro de la cual se pueden desarrollar muchas aplicaciones con la ayuda de algunas herramientas libres que el equipo de Freeswitch pone a disposición de los usuarios.

El desarrollo del proyecto de Freeswitch se enfoca en algunos objetivos de diseño que incluyen modularidad, soporte para diferentes sistemas operativos, escalabilidad y estabilidad.

Freeswitch soporta varias tecnologías de comunicación lo cual facilita la integración con otros sistemas de código abierto para PBX – IP, posee un módulo de conferencias, también tiene soporte para códecs de voz de banda estrecha y banda ancha.

Entre los paquetes de software que utiliza *Freeswitch* dentro de su plataforma están:

- Apache Portable Runtime: Es un repositorio para las librerías de software que utiliza Freeswitch.
- SQLite: Para el manejo de bases de datos.
- Sofia-SIP: Es una librería para desarrollo de software cliente para el protocolo SIP.
- SpanDSP: Librería para procesamiento digital de señales.
- Libsrtp: Librería de código abierto para manejo del protocolo sRTP[13]

[13] Secure Real-Time Transport Protocol.

1.6.4.1 Freeswitch para Raspberry Pi

Existe una versión de Freeswitch que puede ser instalada en la Raspberry Pi [35] que ha sido modificada de la versión original para que sea compatible con la arquitectura ARM del procesador de la Raspberry Pi, esta versión no se distribuye con todos los módulos que utiliza Freeswitch, solamente trae el módulo SBC[14] que permite el registro de teléfonos IP y la realización de llamadas [36].

Esta versión de Freeswitch es compatible con el sistema operativo Raspbian[15] (Wheezy) que utiliza la Raspberry Pi y toma varias horas en ser compilado [36].

Los requisitos mínimos de hardware el funcionamiento de Freeswich en este sistema operativo son: procesador de 1 GHz y memora RAM de 1GB [35] [37].

1.6.5 ANÁLISIS DE LAS PLATAFORMAS DE SOFTWARE LIBRE PARA IP PBX

En las secciones anteriores se han descrito las características de varios paquetes de software para la implementación de centrales PBX de VoIP.

Kolab es una plataforma con muy buenas características para el manejo de herramientas colaborativas como correo electrónico, calendario, tareas, pero necesita de módulos adicionales para poder tener la funcionalidad de telefonía IP [33], algo que no es deseable en el presente proyecto debido a que instalar módulos adicionales puede consumir demasiados recursos de la plataforma de hardware que se considera utilizar.

Trixbox y Elastix poseen características muy similares en cuanto a los servicios que brindan, mismo sistema operativo sobre el que trabajan y tienen similares módulos incluidos para dar la funcionalidad de PBX. Visto de esta manera se analizan las diferencias existentes entre ambas opciones y un punto de inflexión aparece en cuanto al costo de adquisición del software.

Trixbox está disponible en dos versiones, una gratuita y otra de pago. La versión gratuita se encuentra descontinuada y no tiene todas las funcionalidades disponibles por lo que esto puede ser un limitante a la hora de realizar este

---

[14] Session Border Controller: Dispositivo utilizado en sistemas VoIP para controlar la señalización y las transmisiones de datos multimedia durante llamadas telefónicas.

[15] Sistema operativo basado en Debian diseñado para trabajar en la Raspberry Pi.

proyecto. La versión de pago tiene un costo mínimo de 15 USD al año por cada usuario [38] pero no posee todas las funcionalidades y particularmente no permite la conexión con softphones, algo primordial en este proyecto. Por su parte Elastix es completamente gratuito y todas sus funcionalidades estas disponibles. Puede ser usado, copiado, modificado, y redistribuido libremente según los lineamientos de la licencia GLP v2 bajo la cual se rige, lo cual es muy bueno para la realización del presente proyecto. Además Elastix provee de una distribución más ligera llamada uElastix la cual está optimizada para trabajar con la arquitectura del hardware que se va a utilizar.

Freeswitch también es software de código abierto con muchas prestaciones para plataformas IP – PBX y herramientas para el desarrollo de aplicaciones, adicionalmente ofrece una versión de su sistema para trabajar en la Raspberry Pi, lo cual permitiría trabajar con este software en el presente proyecto, pero su limitante viene a la hora de la instalación ya que se debe realizar sobre otro sistema operativo (Raspbian) lo cual consumiría muchos recursos del CPU y su compilación toma bastante tiempo. La instalación básica toma un tiempo de compilación de 10 horas y cada módulo adicional para funcionalidades adicionales toma al menos 2 horas en compilar e instalar. Finalmente la versión de Freeswitch para Raspberry Pi solo permite registrar teléfonos y realizar llamadas, no brinda funcionalidades adicionales.

En este análisis no se pretende descalificar a las otras plataformas de software, más bien se quiere mostrar en cuanto a algunas características que la mejor opción para realizar este trabajo es Elastix – uElastix ya que es fácil de adquirir, posee alta compatibilidad con la plataforma de hardware, y es de distribución libre.

En la Tabla 1.7 se muestra un cuadro comparativo de las plataformas de software descritas anteriormente.

Tabla 1.7 Cuadro comparativo de plataformas de Software.

Software	Sistema Operativo	Licencia	Protocolos Compatibles	Requisitos de Hardware	Observaciones
uElastix	Fedora 18	GPL	SIP, IAX2	Procesador 400 MHz, 1 GB RAM	No posee módulos de mensajería instantánea y CRM
Trixbox	CentOS	Propietario	SIP, IAX2	Procesador 2 GB, 1 GB RAM	Versión libre descontinuada

Software	Sistema Operativo	Licencia	Protocolos Compatibles	Requisitos de Hardware	Observaciones
Kolab	Compatible con Linux y Windows	GPL	No compatible con protocolos VOIP	Documentación no muestra información sobre requisitos de hardware	No posee funcionalidad de PBX, se necesitan módulos adicionales para VoIP.
FreeSwitch	Compatible con Linux, Windows, BSD	Mozilla Public License	SIP, IAX2, H.323	Procesador Pentium 1 GHz, 1 GB RAM	Versión para Raspberry Pi solo posee funcionalidad de SBC

CAPÍTULO 2

2 INSTALACIÓN Y CONFIGURACIÓN DE LA IP PBX

El presente proyecto de titulación tiene como finalidad la configuración e implementación de una central telefónica IP privada basada en hardware y software libre. La implementación se hace sobre una Raspberry Pi como plataforma de hardware y el software de código abierto uElastix; y para la interconexión del sistema con la PSTN se utiliza un *Gateway* de voz.

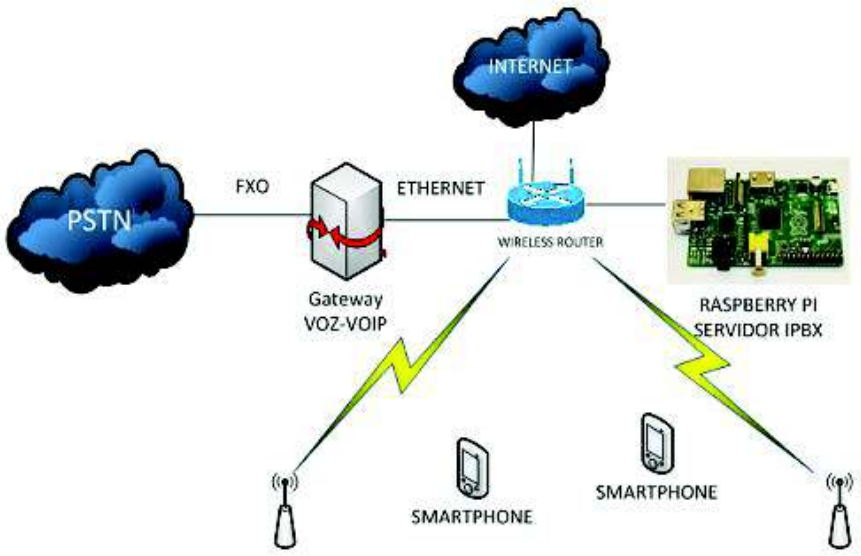


Figura 2.1 Infraestructura de telecomunicaciones para la IP-PBX

En la Figura 2.1 se puede ver que el sistema utiliza para la interfaz inalámbrica una WLAN, a través de la cual los clientes de VoIP se comunican con el servidor *uElastix*. Adicionalmente se va a mostrar la forma de conectar y registrar teléfonos IP fijos. Se utiliza un router inalámbrico Huawei hg532 que soporta el protocolo 802.11 b/g/n. A continuación en este capítulo se presenta la implementación y configuraciones del sistema.

2.1 INSTALACIÓN DEL SOFTWARE PARA IP – PBX uELASTIX

Ya que en la Raspberry Pi la función de disco duro la hace la tarjeta SD, uno de los puntos más importantes al momento de la instalación es el tipo de memoria SD a utilizar. La SD Association define el estándar Speed Class que indica con un

símbolo la velocidad de escritura de las tarjetas SD para ayudar a los consumidores a decidir cuál es la tarjeta que mejor se adapte a sus necesidades [39]. El estándar define la *velocidad mínima garantizada* de desempeño para la escritura de información en las tarjetas SD a una velocidad constante [40], principalmente en aplicaciones de transmisión de datos. Las clases definidas por la SD Association son clase 2, 4, 6, y 10. Las tarjetas que trabajan a velocidades del Speed Class 2, 4 y 6 operan en el modo Normal Speed, que es el modo convencional del bus de datos, y las tarjetas que trabajan con la especificación Speed Class 10, operan en el modo High Speed [41]. En la Tabla 2.1 se exponen las velocidades de escritura definidas en el estándar para cada clase.

Tabla 2.1 Velocidades de lectura/escritura de las tarjetas SD *[39]*

<table>
<tr><th>Estándar</th><th>Clase</th><th>Velocidad de lectura y escritura de datos</th><th>Modo de Bus SD</th><th>Aplicación típica</th></tr>
<tr><td rowspan="4">Speed Class</td><td>Clase 10</td><td>10 MB/s</td><td>High Speed</td><td>Reproducción de Video en Full HD.
Imágenes HD.
Grabación Continua</td></tr>
<tr><td>Clase 6</td><td>6 MB/s</td><td rowspan="3">Normal Speed</td><td rowspan="2">Grabación de video en HD y Full HD</td></tr>
<tr><td>Clase 4</td><td>4 MB/s</td></tr>
<tr><td>Clase 2</td><td>2 MB/s</td><td>Grabación de video en definición estándar</td></tr>
</table>

Una vez que se conoce las especificaciones de las velocidades de las tarjetas de memoria SD se eligió utilizar para este proyecto una tarjeta SD de 16 GB de almacenamiento y clase 10, ya que es la que trabaja a mayores velocidades de lectura y escritura lo cual mejora el desempeño del sistema.

Una vez que se ha elegido la tarjeta SD se siguen los siguientes pasos para instalar uElastix en la Raspberry Pi.

Una vez que se conoce las especificaciones de las velocidades de las tarjetas de memoria SD se eligió utilizar para este proyecto una tarjeta SD de 16 GB de almacenamiento y clase 10, ya que es la que trabaja a mayores velocidades de lectura y escritura lo cual mejora el desempeño del sistema.

Una vez que se ha elegido la tarjeta SD se siguen los siguientes pasos para instalar *uElastix* en la Raspberry Pi.

1. **Descargar uElastix:** Se debe descargar la imagen ISO para Raspberry PI de la página oficial: http://uelastix.com/, posteriormente se descomprime esta imagen en el equipo que se va a utilizar para la instalación y se obtiene una carpeta "elastix-arm-2014-01-30" con tres archivos necesarios para la instalación. Estos archivos son: BOOT.tar.gz, root.tar.gz, usr.tar.gz.
2. **Realizar el particionamiento de la tarjeta SD:** En el presente proyecto se ha utilizado una laptop con el sistema operativo Ubuntu 14.04 y la herramienta fdisk para crear las particiones en la tarjeta SD. Las particiones que se realizaron en la tarjeta SD son 3. La primera es de tipo FAT[16] y de un tamaño mínimo de 256 MB. Esta primera partición contendrá al directorio /boot el cual contiene todos los archivos necesarios para el proceso de arranque del sistema. Este almacena los datos que se utilizan en el arranque antes de que el kernel comience a ejecutar programas del usuario [28].

   La segunda y tercera partición serán de tipo EXT3[17] y de al menos 1.6 GB y 1 GB respectivamente y contendrán a los directorios root y usr de la misma manera.

   Una vez que se han realizado las particiones necesarias se crean los sistemas de archivos para cada partición.

   - #mkfs.vfat -n' /BOOT' /dev/mmcblk0p1
   - #mkfs.ext3 -L '/usr' /dev/mmcblk0p2

[16] File Allocation Table, sistema de archivos desarrollado por MS-DOS y admitido por la mayoría de sistemas operativos y que permite el intercambio de archivos entre sistemas operativos que coexisten en la misma computadora.

[17] Third Extended Filesystem, es un sistema de archivos principalmente utilizado en distribuciones Linux con registro por diario en el que se almacena la información necesaria para restablecer los datos del sistema afectados en caso de falla.

- #mkfs.ext3 -L '/' /dev/mmcblk0p3

La opción "*-n"* establece el nombre del volumen dentro del sistema de archivos FAT, y la opción *"-L"* realiza la misma acción que la opción *"-n"* pero para sistemas de archivos "*ext3*" [42].

3. **Montar los archivos en las particiones:** Se copian los archivos en cada partición creada con el comando que se muestra a continuación.

- `#mount /dev/mmcblk0p1 /mnt/`
- `#tar -C /mnt/ -xzf BOOT.tar.gz`
- `#umount /dev/mmcblk0p1`
- `#mount /dev/mmcblk0p2 /mnt/`
- `#tar -C /mnt/ -xzf usr.tar.gz`
- `#umount /dev/mmcblk0p2`
- `#mount /dev/mmcblk0p3 /mnt/`
- `#tar -C /mnt/ -xzf root.tar.gz`
- `#umount /dev/mmcblk0p3`

Aquí se montan las particiones creadas en la carpeta correspondiente /mnt/ y se copian los archivos descomprimidos para cada partición montada.

4. **Arrancar el sistema:** Una vez que se ha instalado el software en la tarjeta SD se ingresa la tarjeta en la Raspberry Pi y se procede a encender el dispositivo, si todo ha sido instalado correctamente se debe observar los leds de estado en la Raspberry Pi titilando en color amarillo.

2.1.1 INTERFAZ WEB DE ADMINISTRACIÓN DEL SISTEMA [1]

Para acceder a esta GUI, en un buscador web se escribe la dirección IP 192.168.1.251 que es la dirección por defecto que tiene el sistema. El usuario para acceder a esta interfaz es "admin" y la contraseña es "palosanto". También se puede acceder mediante una conexión ssh a la consola de línea de comandos, en este caso el usuario es "root" y la contraseña nuevamente es "palosanto".

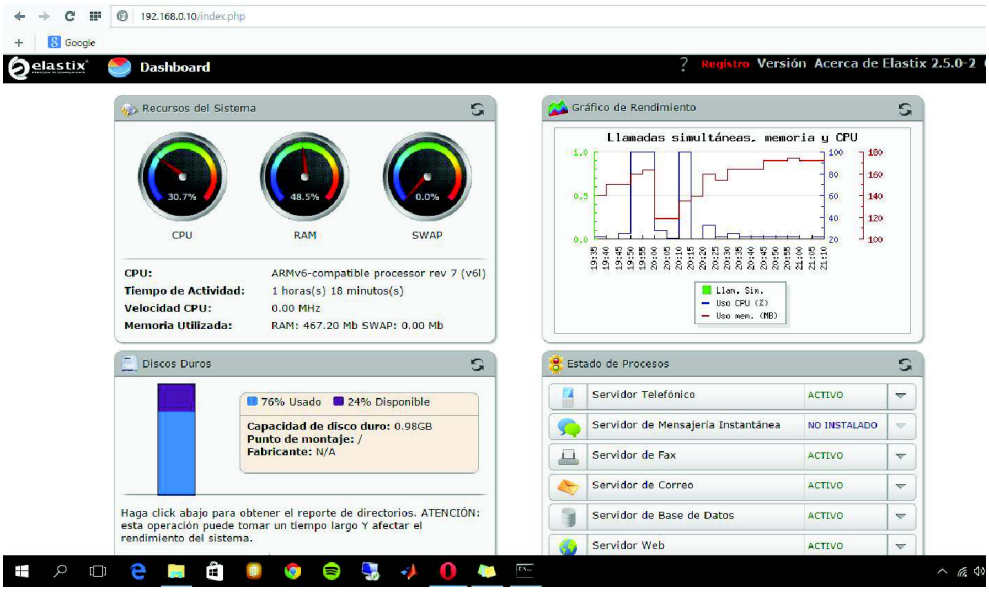


Figura 2.2 Dashboard de administración de uElastix

Como se observa en la Figura 2.2, al ingresar a la interfaz web de *uElastix* la página de inicio es el *Dashboard,* que es una pantalla donde se presentan algunos aplicativos que muestran un resumen del uso de los recursos del sistema, el estado de algunos procesos y las llamadas realizadas hasta el momento. Esta pantalla es totalmente configurable ya que se puede elegir la información que se desea mostrar al ingresar a la interfaz web. En este caso se han mantenido las aplicaciones que vienen configuradas por defecto, éstas son: Recursos del sistema, Estado de Procesos, Discos Duros y Gráfico de Rendimiento. De los gráficos mostrados, el gráfico de rendimiento es el que más información aporta al momento de realizar el monitoreo del servidor.

2.2 CONFIGURACIÓN DE PARÁMETROS DE RED

Una de las primeras acciones a realizar es la configuración de los parámetros de red, para que el servidor de IP-PBX esté dentro de la infraestructura de la oficina o el ambiente en el que se vaya a utilizar y también para que tenga acceso a Internet. Como se mencionó anteriormente esta distribución viene con una dirección IP por defecto que es la 192.168.1.251 con máscara 24. Es recomendable que se configure una dirección IP estática para que los usuarios puedan comunicarse siempre con el servidor. La configuración de estos parámetros se realiza desde el menú “Sistema >> Red“, tal como se indica en la Figura 2.3.

Figura 2.3 Menú para configuración de parámetros de red

Para configurar los parámetros de red se da clic en “Editar parámetros de red”, y luego se despliega un menú como el que se muestra en la Figura 2.4, donde es necesario cambiar los parámetros preestablecidos con los valores indicados en la Tabla 2.2.

Tabla 2.2 Parámetros de red del servidor IP PBX

host	Elx.localdomain.com
Puerta de enlace	192.168.0.1
DNS Primario	192.168.0.1

Figura 2.4 Parámetros básicos de red

Para editar los parámetros de la tarjeta de red del servidor, damos click en la interfaz correspondiente, que en este caso es “*etherneto*”, y se muestra la pantalla que se ve en la Figura 2.5.

Figura 2.5 Edición de los parámetros de red de uElastix

En la ventana de la Figura 2.5 se cambia el tipo de asignación a estático y se configura la IP del servidor y la máscara según los valores de la Tabla 2.3. Estos valores están de acuerdo con la red de prueba con la que se está trabajando, basada en el esquema que se muestra en la Figura 2.1 al inicio de este capítulo.

Tabla 2.3 Parámetros de la tarjeta de red

Dirección IP	192.168.0.10
Máscara de Red	255.255.255.0

2.3 CONFIGURACIÓN DE EXTENSIONES [1]

El menú PBX es uno de las más importantes en *uElastix*, aquí se tienen las configuraciones generales de los servicios provistos por el servidor.

La configuración de la PBX se inicia por la creación de las extensiones para los usuarios del sistema telefónico IP. Para ello se ingresa al menú *PBX* > *Configuración de PBX* > *Extensiones*, donde se accede a la pantalla que se muestra en la Figura 2.6. En esta pantalla se pide determinar el tipo de dispositivo que se va registrar en la central telefónica.

Una IP-PBX con *uElastix* soporta 5 tipos de dispositivos: *SIP, IAX2, DAHDI, Virtual y Custom*.

Las características de estos dispositivos son las siguientes:

- **Dispositivos SIP:** Son los dispositivos que soportan el protocolo SIP. Este es el protocolo mayormente soportado por teléfonos IP, dispositivos ATA[18] y softphones.
- **Dispositivos IAX2:** Son dispositivos que soportan el protocolo IAX2 propietatio de Digium, la empresa que desarrolla el sistema Asterisk.
- **Dispositivos ZAP:** Estos dispositivos trabajan con canales DAHDI. Éste es un conjunto de controladores para hardware telefónico como tarjetas PCI que permiten configurar canales para la conexión a la PSTN.
- **Virtual:** Añade la capacidad de crear una extensión virtual.
- **Custom:** Puede ser utilizado para asociar una extensión a un número externo.

Figura 2.6 Menú de configuración de extensiones

Para el objeto de este proyecto se ha seleccionado trabajar con extensiones de tipo SIP, como ya fue discutido en el capítulo 1. Los otros parámetros que se necesitan configurar son:

- Número de la extensión.
- Nombre asignado a esa extensión.
- Clave para el registro del dispositivo terminal en el servidor.

[18] ATA: Analog Telephone Adapter: Adaptador para teléfonos analógicos, permite utilizar un teléfono convencional como teléfono IP en una IP-PBX.

Ya para la configuración, primero se selecciona el tipo de extensión a utilizar (SIP), para ello se hace *click* en el menú *"Device"* y se elige "*Generic SIP Device*", se guardan los cambios dando click en "*Submit*". Luego de esto se despliega una pantalla con una serie de opciones donde se ingresan los demás parámetros para la configuración de la extensión. Como ejemplo se presentan los parámetros para la extensión 101. Estos parámetros son los siguientes:

- Tipo de extensión: SIP
- Número de extensión: 101
- Nombre de extensión: Usuario1-101
- Clave de extensión: raspi101*

Figura 2.7 Configuración de extensión SIP

En la Figura 2.7 se observa el menú en donde se deben ingresar estos valores en los parámetros correspondientes.

Se ingresan los valores descritos anteriormente y se procede a dar *click* en "*Submit*" nuevamente. Un paso importante es dar *click* en la opción resaltada en rosado que aparece en la parte superior de la pantalla para que los cambios sean aplicados de forma permanente.

Como se puede ver en la figura 2.7, la configuración de una extensión consiste en asignar estos cuatro parámetros, los demás campos se dejan con la configuración por defecto. El formato de la clave asignada puede ser cualquiera. En el presente proyecto se va a utilizar de la siguiente forma: raspi[número de extensión]*.

En total se han configurado 14 extensiones de las cuales, 10 pertenecen a usuarios de la IP-PBX y las cuatro extensiones restantes se utilizan para realizar pruebas.

Adicionalmente se ha habilitado el correo de voz en las extensiones de manera que cuando no se conteste una llamada esta sea enviada a al sistema de correo de voz y se pueda dejar un mensaje al usuario que se ha estado llamando:

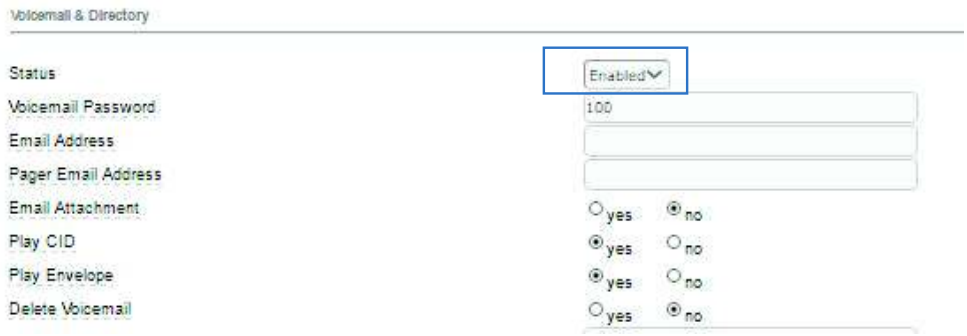


Figura 2.8 Configuración de correo de voz

Para habilitar el correo de voz en una extensión hay que dirigirse a la sección "Voicemail & Directory" en la configuración de extensiones y en la opción "Status" seleccionar "Enable" como se observa en el cuadro azul de la Figura 2.8. Luego se debe configurar una contraseña para que el usuario pueda ingresar a su correo de voz. Finalmente se selecciona "Yes" en las opciones "Play CID" y "Play Envelope" para que el sistema de correo de voz locute el número de extensión y la fecha cuando se escuche el mensaje de voz.

2.4 CONDIGURACIÓN DE CLIENTE SIP EN SMARTPHONE

Como cliente SIP se utiliza la aplicación Zoiper que se puede instalar en casi cualquier smartphone y es compatible con las plataformas Android y iOS [43]. Esta aplicación se distribuye a través de Google Play o iTunes de manera gratuita, permite hacer llamadas de VoIP a través de redes 3G o WiFi siempre y cuando se cuente con un proveedor de VoIP. Soporta los protocolos SIP e IAX, tiene un marcador integrado, permite las funciones de mute y altavoz, soporta transporte sobre UDP o TCP, y trabaja con los códecs G.711 (ulaw, alaw), iLBC y GSM [43].

La configuración de la extensión en este cliente es bastante simple e intuitiva, solamente se necesita tres datos: la dirección IP del servidor IP-PBX, el número de la extensión y la contraseña.

Para realizar la configuración se ingresa a "*Config* >> *Cuentas* >> *SIP"*. En la Figura 2.10 se muestra la pantalla a la que se accede para ingresar los datos necesarios para la configuración. Estos datos son:

- La dirección IP del servidor (Host: 192.168.0.10).
- El número de la extensión (nombre de usuario: 1320).
- Clave o contraseña que es la misma que se configura en la central IP-PBX para la extensión.

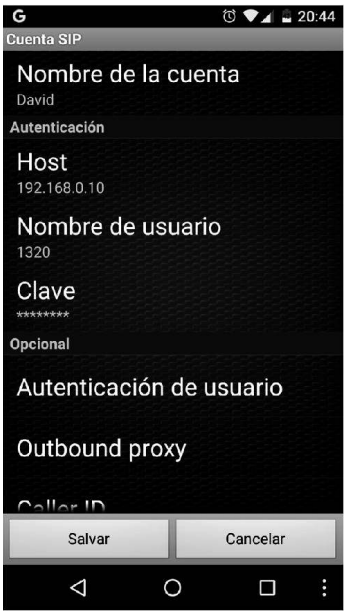


Figura 2.9 *Interfaz para configuración de cuenta SIP*

Una vez ingresados los datos antes mencionados en los campos correspondientes, se guarda la configuración y se espera a que el cliente se registre en el servidor. Como se observa en la Figura 2.10, cuando el cliente se registra en el servidor aparece un aviso en la parte superior, como la que se observa en el recuadro, que indica que está listo para hacer llamadas.

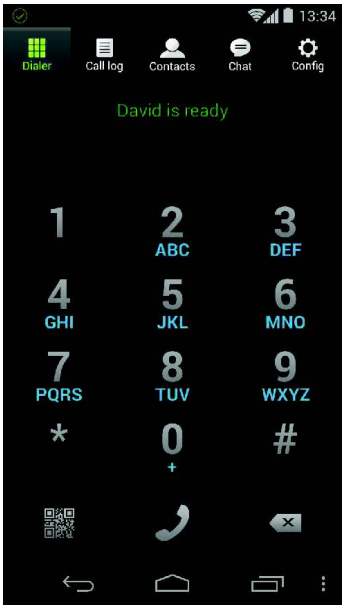


Figura 2.10 Interfaz de Zoiper para Android

2.5 CONFIGURACIÓN DE TELÉFONO IP

A continuación se muestra la configuración de un teléfono IP el cual se ha añadido al presente proyecto para verificar su funcionamiento con la central IP PBX. El dispositivo es un teléfono IP marca *Grandstream* modelo *Budgetone 200*.

Para acceder a la interfaz web de configuración del teléfono IP primero se verifica la dirección IP que tiene el dispositivo, la cual ha tomado vía DHCP desde el router. Para verificar esta dirección se puede navegar a través del menú del teléfono IP utilizando las teclas enceradas en el cuadro de color rojo, como se observa en la Figura 2.11.

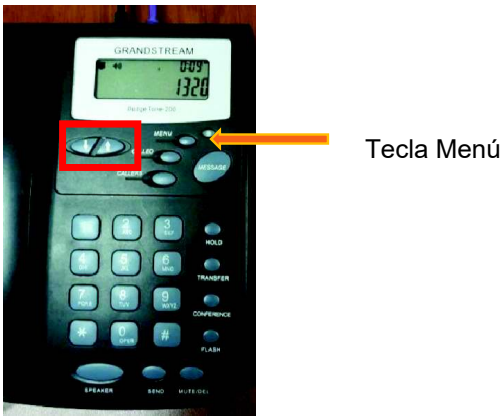


Figura 2.11 Teléfono IP Grandstream 200

Una vez que se conoce la dirección IP del dispositivo se procede a ingresar a la interfaz web mediante un navegador. Dentro de la interfaz web se da clic en “BASIC SETTINGS” para ir a las opciones básicas de configuración del dispositivo.

Aquí se configuran las siguientes opciones como sigue:

- IP address: dynamically assigned vía DHCP
- Display account name instead of Date: Yes

Las otras opciones se dejan como están por defecto. Luego se da clic en “Update” para guardar las configuraciones.

El siguiente paso es dar clic en “ADVANCE SETTINGS”. Aquí se configuran las frecuencias de los diferentes tonos que va a utilizar el teléfono IP, como tono de marcado, tono de ocupado, etc.

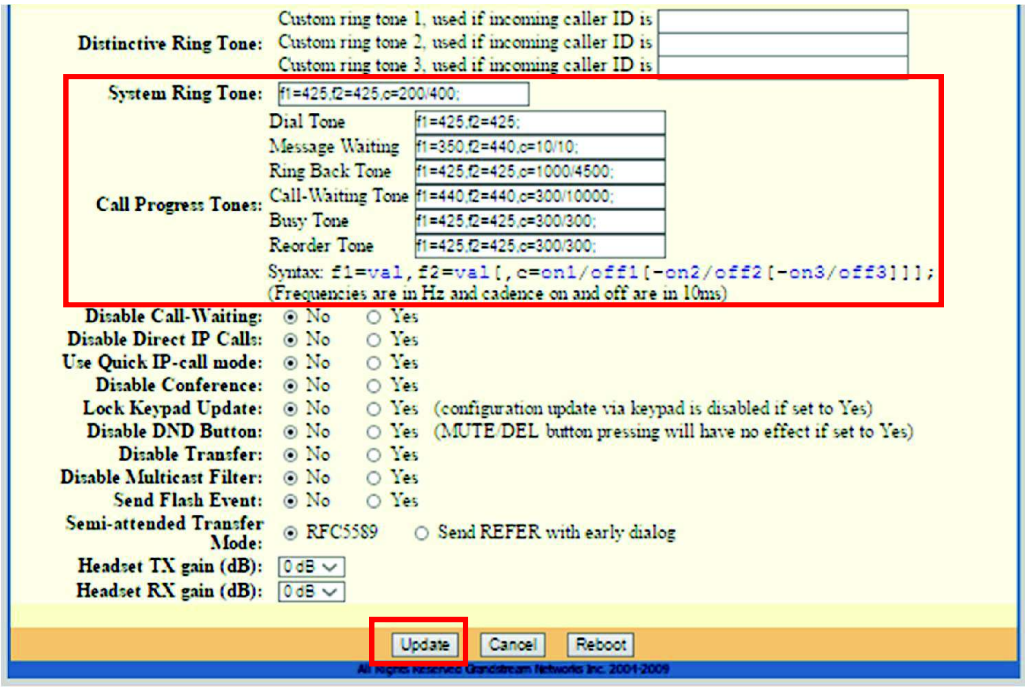


Figura 2.12 Configuraciones avanzadas para teléfono IP

En la Figura 2.12 se observa las frecuencias que se han configurado para cada tono del teléfono IP. Estas frecuencias corresponden a la recomendación E.180 [44]

de la ITU-T[19] correspondientes a los tonos usados en redes nacionales de telefonía para cada país.

Se da clic en el botón “Update” (El botón dentro de recuadro de color rojo en la Figura 2.12) para guardar los cambios.

Finalmente se ingresa a la opción “ACOUNT” para configurar los parámetros de la cuenta SIP para el teléfono IP como el número de extensión, la dirección IP de la central IP PBX, la contraseña para que el equipo se registre en la central IP PBX. Estos parámetros son muy similares a los que se debía configurar para el cliente SIP instalado en un Smartphone.

Los valores que se deben ingresar en estos parámtros son los que se detallan en la Tabla 2.4.

Tabla 2.4 Parámetros de cuenta SIP para registro en la central IP PBX

Parámetro	Dato	Descripción
Account Name	102	Nombre de la cuenta SIP
SIP Server	192.168.0.10	Dirección IP de la central IP PBX
SIP user ID	102	Número de la extensión asignada al teléfono IP
Authenticate Password	rpi102*	Contraseña para el registro del teléfono IP en la central IP PBX
Name	102	Un nombre para que se muestre en la pantalla del teléfono IP

En la Figura 2.13 se muestra dentro de cuadro de color rojo el ingreso de los parámetros descritos en la Tabla 2.4 a la interfaz web de configuración.

[19] Unión Internacional de Telecomunicaciones

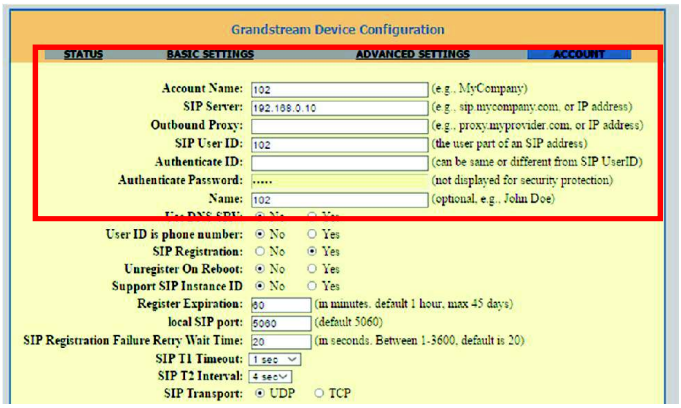


Figura 2.13 Parámetros de configuración cuenta SIP

Si todo ha sido configurado correctamente en la pestaña “STATUS” se va a observar el mensaje “Registered” lo cual indica que el teléfono IP se ha registrado en la central IP PBX exitosamente, como se observa en el cuadro de color verde de la Figura 2.14.

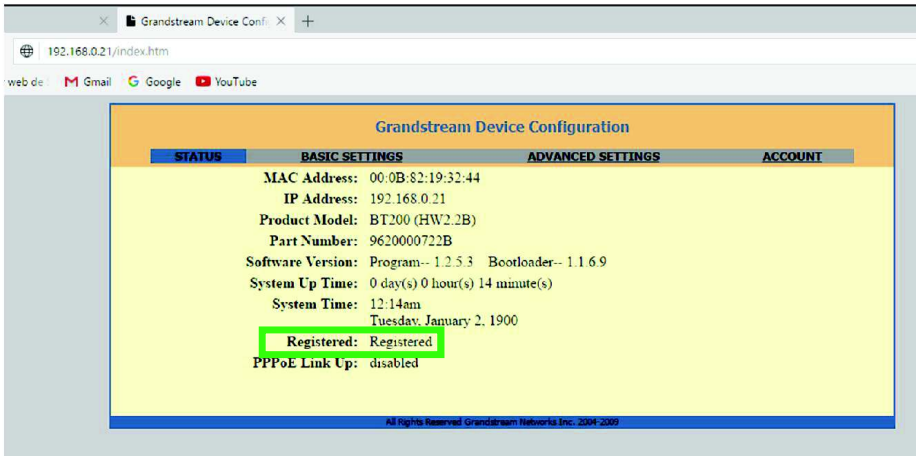


Figura 2.14 Estado del teléfono IP

2.6 CONFIGURACIÓN DE TRONCAL SIP [1] [45]

La salida de llamadas telefónicas se puede realizar por diferentes medios como líneas E1, troncales SIP y líneas analógicas. Para el caso particular de este proyecto, se va a configurar una troncal de tipo SIP que comunique al servidor IP-PBX con el *gateway* de voz para que las llamadas puedan ser terminadas en la PSTN.

Una troncal es la que permite que una llamada sea terminada en otro sistema de comunicación de voz, por ejemplo desde la central IP-PBX hacia la Red de telefonía pública conmutada o *PSTN*. En el presente proyecto se va a configurar un enlace troncal SIP desde la central IP-PBX hacia el Gateway de voz de manera que las llamadas puedan ser enrutadas hacia la PSTN. Finalmente el Gateway conectará las llamadas mediante la línea telefónica del usuario.

Una troncal SIP se configura en un servidor (En este caso el servidor de telefonía IP) y es utilizada por otros servidores, es decir, el dispositivo que está del otro lado de la conexión TCP o UDP de la troncal SIP sería otro servidor (En este caso el *gateway VoIP*) y no un dispositivo final de usuario [46].

Para la configuración de una Troncal SIP, primero se accede al menú PBX>Troncales, luego de esto se despliega una pantalla y se pide elegir el tipo de troncal a crear, que para el presente caso será de tipo SIP.

Luego de elegir el tipo de troncal, se despliega una nueva pantalla con algunas opciones para la configuración de la troncal como se observa en la Figura 2.15.

Figura 2.15 Parámetros de configuración de Troncal SIP

Algunos de los parámetros a configurar son el *Trunk Name, Outbound Caller ID, Dialed Number Manipulations Rules,* y los *Peer Details.* A continuación se explican cada uno de estos parámetros:

- ***Trunk Name:*** En este campo se asigna un nombre distintivo a la troncal para poder identificarla.

- ***Outbound Caller ID:*** En esta opción se especifica el número de origen que utilizará una extensión cuando se realice una llamada hacia la PSTN.
- ***Dialed Number Manipulations Rules:*** Las reglas de marcado indican cómo se debe marcar dentro de la IP-PBX para establecer una llamada hacia la troncal SIP. Además existe cierta sintaxis que se debe respetar cuando se escriben las reglas de marcado dependiendo de cómo se quiere que se ejecuten las mismas. Estas se reglas se detallan a continuación en la Tabla 2.5.

Tabla 2.5 Reglas de marcado

Patrón	Descripción
X	Cualquier dígito de 0 a 9
Z	Cualquier dígito de 1 a 9
N	Cualquier dígito de 2 a 9
[1234-9]	Cualquier dígito que se encuentre entre los corchetes.
.	Uno o más caracteres
|	Separa el número ubicado a la izquierda del número marcado.
+	Adiciona un prefijo al número marcado.

- ***Peer Details:*** Para que una troncal funcione correctamente se requieren configurar ciertos parámetros denominados *"peers"* que se refieren a la forma cómo la troncal establece una conexión con la central. En la figura 2.15 se observa los *peer details* necesarios para el funcionamiento de la troncal SIP. A continuación se describe cada uno de estos [47]:
  - *dtmfmode = rfc2833:* El rfc2833 es el método de señalización mediante tonos más comúnmente utilizado. *Dtmfmode* se refiere al modo en que se realiza la señalización en la troncal.
  - *canreinvite = no:* Cuando se establece el valor de canreinvite en "no" significa que la PBX no puede interconectar dos teléfonos

directamente sino que enruta las llamadas entre ellos a través de sí misma.

- *context = from-internal:* Aquí se establece cómo se van a manejar las llamadas provenientes de la troncal hacia la PBX. *"from-internal"* quiere decir que cualquier llamada proveniente de esta troncal será tratada como si se tratara de una llamada hecha por cualquier teléfono de la central telefónica, es decir que puede ser desviada a alguna extensión, a un IVR[20] o alguna otra ruta entrante. Generalmente se establese el valor de *"from-internal"* para troncales que empatan dos sistemas de comunicación confiables.
- *host = dynamic:* Este campo indica el sistema remoto que se va a conectar a la IP – PBX, en este caso la troncal SIP. Si el sistema remoto se va a registrar dentro del servidor utilizando sus credenciales, se debe configurar este campo como *"dynamic"*.
- *type = friend:* Significa que existe la posibilidad de hacer y recibir llamadas desde este servidor y que los detalles del peer se usarán para las llamadas entrantes y salientes en la troncal.
- *port = 5060*: Es el puerto por defecto que utiliza la PBX para enviar las llamadas y por donde se espera que ingresen las llamadas hacia la misma.
- *qualify = yes*: Quiere decir que la PBX enviará peticiones periódicas a la troncal para que ésta se identifique. Si no existe respuesta luego de 2 segundos, la PBX asumirá que la troncal no está en funcionamiento y dejará de enviar llamadas hacia ésta. También es posible establecer el valor de este parámetro en un número que indica el tiempo en milisegundos que el sistema esperará por una respuesta de la troncal.

[20] IVR: Interactive Voice Responce, Respuesta de voz interactiva.

- *disallow=all*: Cuando se establece el valor de este parámetro en *"all",* la PBX no utiliza ningún códec de audio a menos que éste sea expresamente permitido en el parámetro correspondiente.
- *allow=ulaw*: Se establece que el codec *g.711* (*ulaw)* es el codec que está permitido para ser utilizado en el sistema. Este comando solo tiene efecto si se han deshabilitado todos los códecs en el parámetro anterior.

Se llenan las casillas correspondientes en la página de configuración que se observan en la Figura 2.16 con los valores asignados a cada uno. El resumen de parámetros los encontramos en la Tabla 2.6.

Tabla 2.6 Detalle de Peers configurados en la troncal SIP

PARÁMETRO	**VALOR**	**DESCRIPCIÓN**
Trunk Name	4999	Nombre de la troncal SIP
Outbaund Caller ID	6001769	Número de la línea analógica
Dialed Manipulations Rules	(Prepend) + 9 | X	Regla de marcado para llamada de salida

Peer Details	`dtmfmode=rfc2833`
	`canreinvite=no`
	`context=from-internal`
	`host=dynamic`
	`type=friend`
	`port=5060`
	`qualify=yes`
	`disallow=all`
	`allow=ulaw`	

Cuando ya se ingresen todos los parámetros, se hace clic en *"submit"* para guardar los cambios y aparecerá la nueva troncal en la derecha de la pantalla como se observa en la Figura 2.17. Finalmente para asegurar que se guarde la configuración se da clic en *"Apply Configuration changes here"*. Esta opción sirve para dar una confirmación final de los cambios que se han realizado.

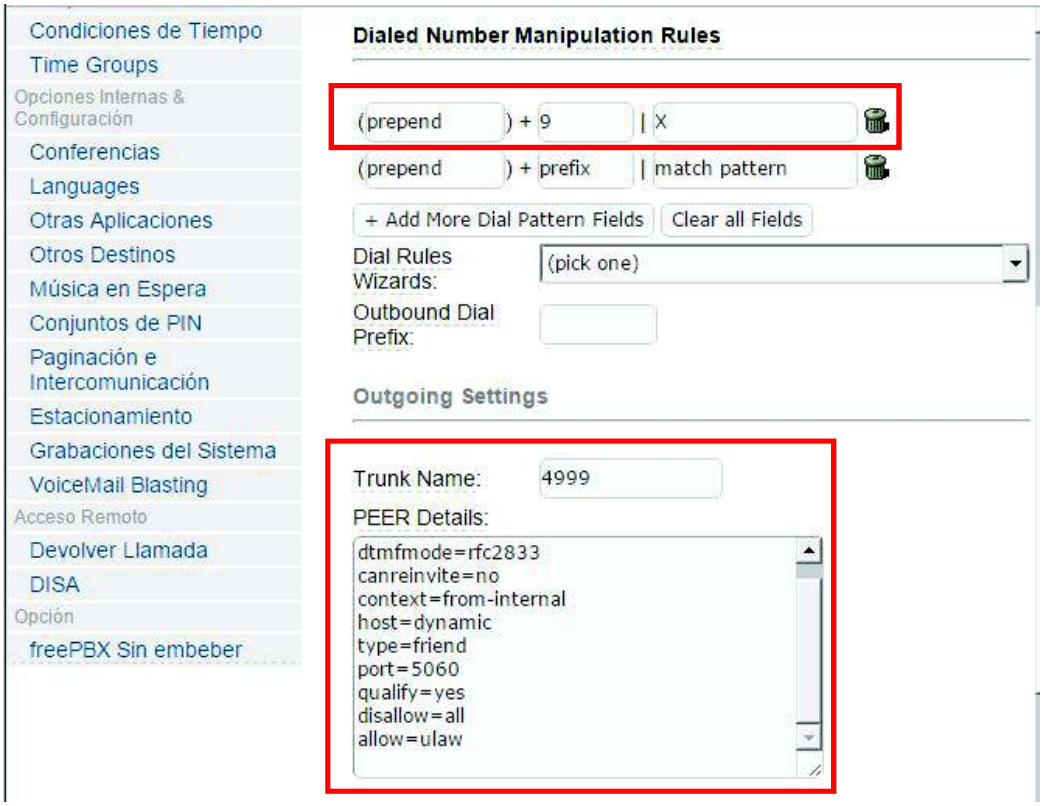


Figura 2.16 Reglas de marcado y peer details de la troncal SIP

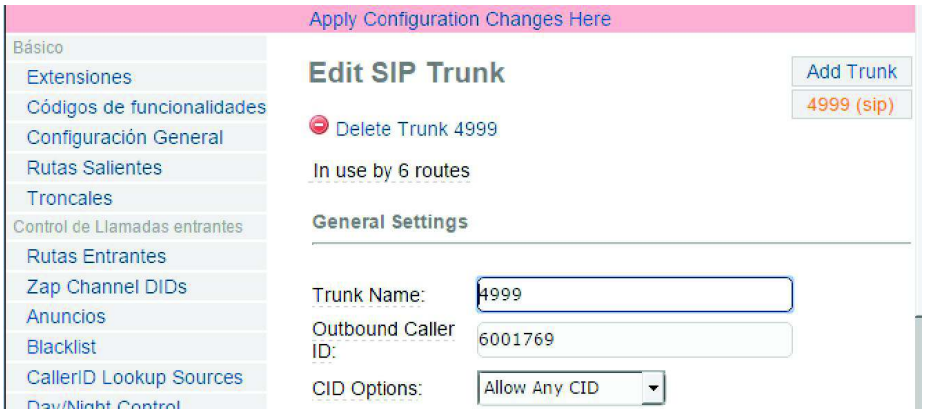


Figura 2.17 Troncal SIP creada

2.7 CONFIGURACIÓN DE RUTAS ENTRANTES Y SALIENTES

Las rutas son las reglas que le indican a la PBX por cuál troncal debe enviar una llamada siguiendo las reglas del *plan de marcado*[21] [45].

De la misma forma también se pueden configurar algunas rutas para recibir las llamadas, de esta manera cuando se recibe una llamada por una troncal específica, ésta puede ser desviada hacia una extensión o a un IVR[22] [1]. Para ello se implementa un plan de marcado para las llamadas entrantes.

Estas rutas pueden ser de dos tipos: rutas de salida y rutas de entrada.

2.7.1 RUTAS DE SALIDA [1]

Una ruta de salida sigue las reglas de marcado que indican al servidor *uElastix* por cuál de las troncales configuradas debe establecer una llamada, es decir la ruta de salida indica el camino que debe seguir una llamada para conectarse con su destino. Cuando se marca un número la PBX busca coincidencias entre las rutas creadas y procede a establecer la llamada por la troncal correspondiente.

La PBX permite tener más de una troncal para una misma ruta de salida, las troncales se utilizan en base a prioridades definidas por el administrador de la PBX. Así se pueden crear todas las rutas de salida que se desee, aplicando las reglas de marcado correspondientes y tomando en cuenta que un mismo plan de marcado no puede repetirse en diferentes rutas de salida ya que se producirían conflictos de enrutamiento de las llamadas.

2.7.1.1 Configuración de la Ruta de Salida

Para realizar la configuración de una ruta saliente, en el menú de PBX se ingresa a Rutas Salientes en el menú PBX >> Configuración PBX >> Rutas Salientes. Una vez en esta página se despliega una pantalla como la que se observa en la Figura 2.17 donde se va a ingresar algunos parámetros de configuración.

---

[21] Plan de marcado: Conjunto de instrucciones que ejecutará el servidor para el manejo de las llamadas entrantes.

[22] Interactive Voice Responce: Mecanismo que contesta automáticamente las llamadas y ofrece un menú interactivo al usuario con algunas opciones a elegir.

Figura 2.18 Parámetros de configuración para rutas entrantes

Como se observa en la Figura 2.18, lo primero es establecer un nombre para la ruta de salida. Este nombre debe colocarse de acuerdo a la función que va a cumplir esta ruta. Para este ejemplo el nombre de la ruta es "*to_pstn*" porque las llamadas se van a dirigir hacia la PSTN.

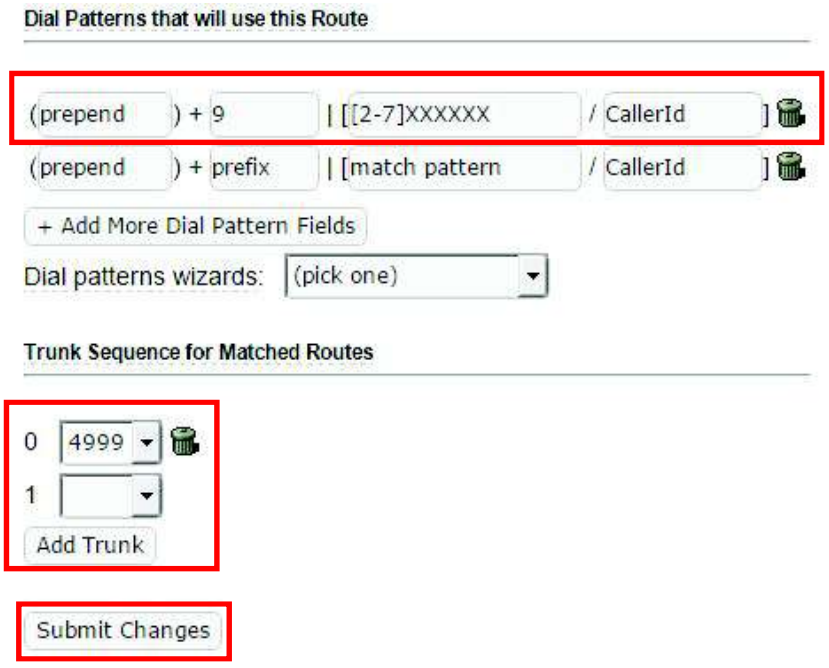


Figura 2.19 Configuración de los patrones de marcado y troncal de salida

Las siguientes opciones que se configuran son los patrones de marcado, como se observa en la Figura 2.19. Como se mencionó anteriormente en la sección 2.6 y

en la tabla 2.5, estos patrones responden a ciertas reglas de marcado que utiliza la PBX para determinar la ruta por donde se va a enviar la llamada. Para la ruta de salida el patrón de marcado dice lo siguiente:

Prefijo de marcado 9 + cualquier dígito entre 2 y 7 + 6 números cualquiera.

Si el número que marca el usuario coincide con el patrón de marcado, se establece la llamada, caso contrario se busca otra ruta que coincida.

Luego, en el menú Trunk Sequence for Matched Routes o Secuencia de troncales para las rutas coincidentes, se escoge el orden en que se debe intentar hacer uso de las troncales para enviar las llamadas. Finalmente se aceptan los cambios (Submit Changes) y la ruta queda guardada. En total se han configurado 6 rutas salientes que son: to_pstn, Emergencia, Servicios, 1800, Nacional y Móvil. Los patrones de marcado para estas rutas se muestran en el Anexo A.1.

2.7.1.2 Ruta de salida para número de emergencia 911

Para la configuración de la ruta para las llamadas al número de emergencia 911 se ingresa al menú “Rutas Salientes” de la misma manera que en el caso anterior. Aquí se elige el nombre para la nueva ruta “Emergencia”, pero se cambia el patrón de marcado de manera que se pueda llamar directamente al número 911.

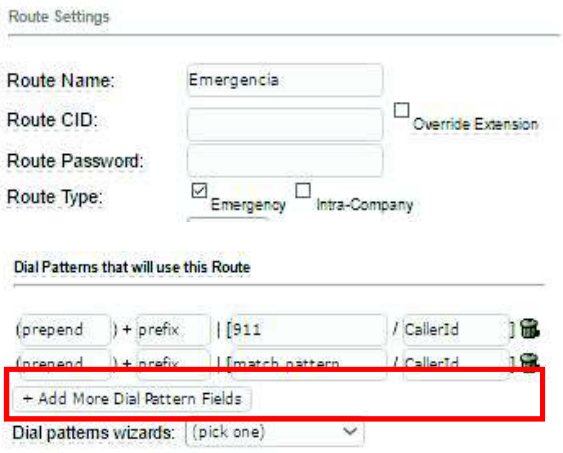


Figura 2.20 *Configuración de Ruta de salida "Emergencia"*

De esta manera el patrón de marcado queda como se observa en el cuadro rojo de la Figura 2.20. De esta manera se marca directamente al número 911 sin anteponer ningún prefijo facilitando así las llamadas a este servicio de emergencia. La troncal por donde va a salir esta llamada es la misma que se ha configurado antes (4999).

2.7.2 RUTAS DE ENTRADA

Una ruta de entrada permite a la PBX saber hacia dónde debe enviarse una llamada recibida, este procedimiento está basado en ciertas condiciones, por ejemplo cuando se desea que una llamada se reciba en un IVR al estilo de una contestadora automática, se debe indicar que las llamadas provenientes de cierta troncal se transfieran al IVR. De la misma forma, si en una empresa existen varias troncales con un número telefónico cada una, se deberán crear igual número de rutas de entrada como troncales existan.

2.7.2.1 Configuración de rutas de entrada

Para realizar la configuración de la ruta de entrada en el menú de PBX se accede a Rutas Entrantes de la siguiente forma *PBX >> Configuración PBX >> Rutas Entrantes.*

Una vez en esta página se despliega la pantalla que se observa en la Figura 2.21, donde se procede a dar un nombre que describa a la ruta.

Figura 2.21 Parámetros de configuración de ruta entrante

En el presente caso la ruta de entrada servirá para que todas las llamadas entrantes a la PBX sean desviadas a un IVR, como se observa en la Figura 2.22, donde se darán varias opciones a los usuarios para que elijan una de ellas en función de su necesidad.

Figura 2.22 Establecimiento del destino para las llamadas entrantes

2.8 CONFIGURACIÓN DE IVR - INTERACTIVE VOICE RESPONSE

La respuesta interactiva de voz o IVR (Interactive Voice Response) es una funcionalidad de la IP-PBX capaz de contestar una llamada y capturar o entregar información a través del teléfono, donde se indican una serie de opciones a las que se puede acceder mediante la pulsación de un botón en el teléfono.

Los IVR son ampliamente utilizados en instituciones bancarias o en empresas donde se reciben gran afluencia de llamadas y se desea reducir la necesidad de personal y el costo relacionado a la realización de estas tareas. Se utiliza entonces la tecnología de IVR para redirigir una llamada según las necesidades del cliente hacia otro departamento o hacia una persona más calificada para resolver determinada solicitud, reduciendo así los tiempos de espera.

2.8.1 PROCEDIMIENTO PARA LA CONFIGURACIÓN DEL IVR

Antes de realizar la configuración del IVR, se debe disponer de un archivo de audio que será el que se utilice para la reproducción del mensaje. Existen dos maneras de obtener el audio, cargar un archivo pre-grabado ó grabar un mensaje desde una extensión. Para la configuración del IVR en la PBX se ha optado por utilizar un archivo pre-grabado. Para ello se recurre al servicio de TTS (Text to Speech) brindado por la página www.nuance.es[46] el cual permite convertir un texto a voz con una muy buena calidad y un tono casi natural ofreciendo voces en español femeninas y masculinas. Luego de escribir el texto que se quiere pasar a voz se procesa y se descarga al servidor. El servidor IP-PBX con uElastix permite utilizar archivos de audio en formato Mp3 monoestéreo y WAV con codificación PCM, 16 bits y con un muestreo a 8000 Hz. En el presente proyecto se va a utilizar un archivo de audio en formato Mp3.

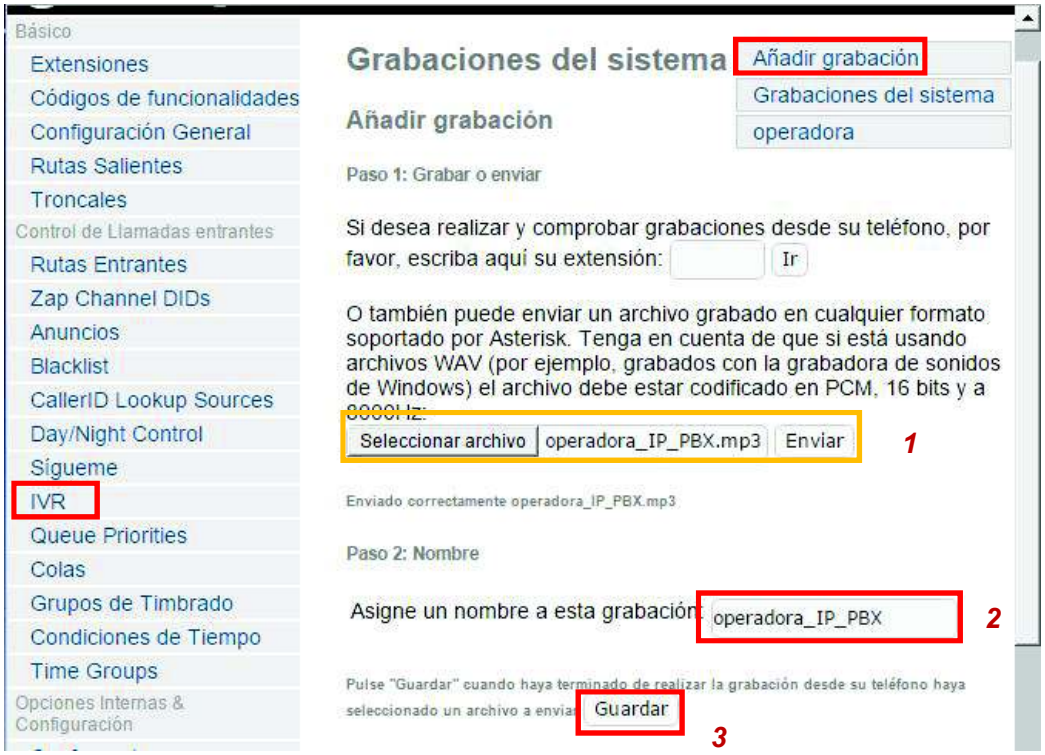


Figura 2.23 Pantalla para cargar las grabaciones del sistema

Ahora el siguiente paso es cargar el archivo de audio en el sistema. Para ello en el menú de configuración de la PBX se accede al menú *Grabaciones del sistema* en *PBX* > *Configuración de PBX* > *Grabaciones del sistema,* donde aparecerá la pantalla que se muestra en la Figura 2.23.

Para añadir la grabación se siguen los siguientes pasos:

1. Dar Click en ***Seleccionar archivo:*** Se selecciona el archivo de audio que se va a cargar, se da *click* en aceptar y posteriormente se observa el archivo en la pantalla de las grabaciones. Para que el archivo se cargue en el sistema se da click en la opción ***Enviar***.
2. Nombre de la grabación: A la grabación se le asigna el mismo nombre que tiene el archivo de audio.
3. Guardar la grabación: Finalmente se da click en ***Guardar*** y la grabación aparecerá en el listado de grabaciones del sistema en el lado superior derecho de la pantalla.

A continuación se configura el IVR como tal. Para ello se accede a la opción de IVR en el menú PBX siguiendo la ruta *PBX* >> *Configuración de PBX* >> *IVR*, luego de lo cual se ingresa a la pantalla que se muestra en la Figura 2.24.

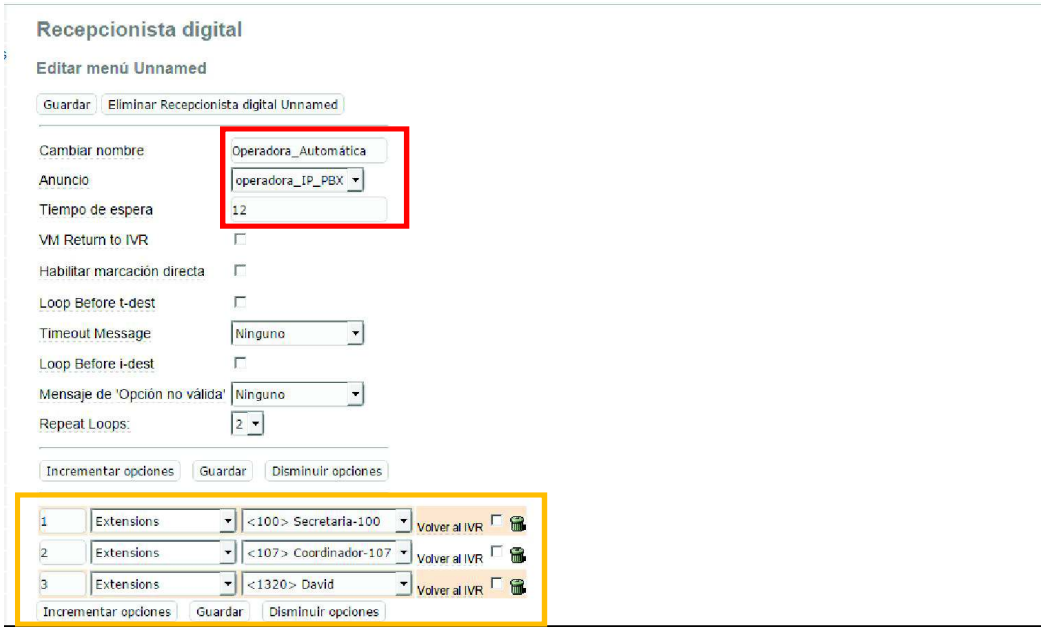


Figura 2.24 Pantalla para configuración de IVR

En la pantalla de la Figura 2.24, los parámetros que se deben configurar son el nombre de la operadora, el audio que previamente se cargó al sistema y que se va a reproducir cuando ingrese una llamada a la IP-PBX, el tiempo de espera que debe transcurrir antes que se reproduzca el audio y finalmente las opciones que van a poder elegir las personas cuando llamen.

Como se observa en el recuadro naranja de la Figura 2.24, se dan tres opciones a los usuarios, que concuerdan con el audio que se reproduce, para que sus llamadas sean transferidas de acuerdo a la necesidad que tengan.

Luego de llenar todos los parámetros descritos anteriormente se da *clic* en *Guardar* y el IVR será guardado en el servidor listo para ser agregado a una ruta entrante, como fue descrito en la sección 2.7.2.1.

2.9 CONFIGURACIÓN DE CÓDIGOS PARA USUARIOS PRIVILEGIADOS

Existen rutas de salida configuradas con restricciones, como las llamadas hacia números móviles celulares y llamadas a nivel nacional. Para las rutas de salida con restricciones, se ha asignado una configuración la cual brinde acceso a esa ruta a usuarios privilegiados. Esta opción se encuentra en la sección PBX >> PIN Set. Mediante el uso de este código o PIN[23] ciertos usuarios tendrán la posibilidad de realizar llamadas por las rutas de llamadas Celulares, Nacionales, en las cuales previamente se ha habilitado la opción para el uso del PIN. La configuración de estas rutas se puede ver en el Anexo A.1.

En la configuración se da una descripción del PIN con el que posteriormente se añadirán las rutas de salida que requieran del PIN. Luego se seleccionará la opción *"Record in CDR"* para que las llamadas que se realicen por medio del PIN también aparezcan en el registro de llamadas. Finalmente se añaden los códigos a utilizar los cuales deben ser marcados por los usuarios cuando el sistema se lo solicite para que la llamada pueda ser realizada.

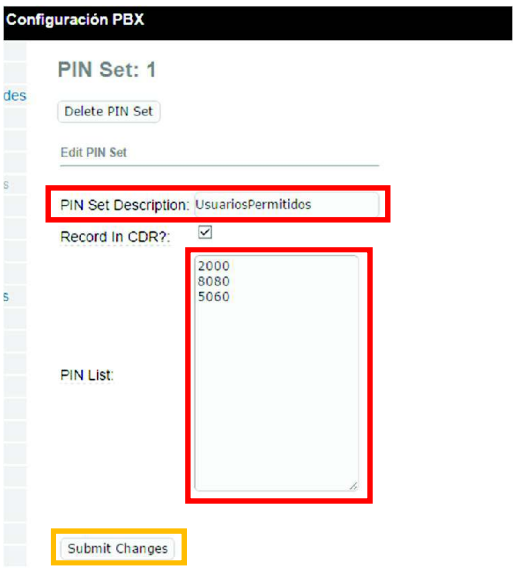


Figura 2.25 Configuración de PIN para usuarios privilegiados

[23] PIN: Código que debe marcar un usuario para poder realizar llamadas a ciertas rutas que tienen restricciones.

En la Figura 2.25 se observa la pantalla de configuración para el PIN. Los códigos a utilizar son los que se indican en el parámetro *PIN List.*

2.10 CONFIGURACIÓN DE *PAGIN* E INTERCOMUNICACIÓN

Estos perfiles se configuran con la finalidad de juntar un determinado grupo de extensiones en una comunicación semejante a una conferencia donde todos puedan participar y escuchar. Para ellos se han creado tres perfiles, los cuales son los siguientes:

- Código 7069 – Intercomunica a los usuarios de un determinado departamento.
- Código 7071 – Intercomunica a las extensiones 107 (Coordinador), 108 (Auditor), y 109 (Director).
- Código 7070 – Es un perfil para pruebas de funcionamiento que puede mantenerse o ser eliminado al finalizar la configuración del sistema.

La configuración de estos perfiles se encuentra en la Figura 2.26.

En la figura 2.26 se observa los usuarios asignados a la conferencia 7071, los cuales recibirán una llamada del usuario que marque a la extensión de intercomunicación.

Figura 2.26 Números asignados al grupo de Intercomunicación

2.11 CONFIGURACIÓN DE FOLLOW ME (SEGUIR)

Esta configuración se puede aplicar cuando se requiere que alguno de los usuarios de cierto departamento conteste y atienda algún requerimiento. La funcionalidad del perfil FOLLOW ME permite que una llamada entrante vaya saltando de una extensión a otra si algún usuario no se encuentra disponible, el tiempo de timbrado antes de que la llamada salte a otra extensión se ha establecido en 10 segundos.

Como se observa en la Figura 2.27 los parámetros que se configuran son el tiempo de timbrado y la siguiente extensión a la que salta la llamada en caso de no ser atendida en el tiempo establecido.

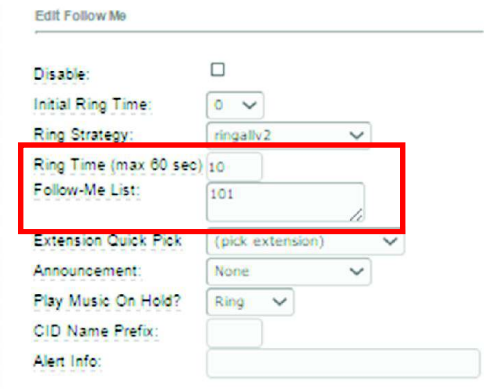


Figura 2.27 Configuración de FOLLOW ME

2.12 CONFIGURACIÓN DE CONFERENCIA

El software para IP PBX uElastix permite configurar sesiones de conferencias de manera que 3 o más personas puedan mantener una llamada entre todos ellos simultáneamente.

Para configurar una conferencia se accede al menú *PBX >> Configuración de PBX >> Conferencias* y se mostrará la pantalla que se muestra en la Figura 2.28.

Figura 2.28 Configuración de Conferencia

En ésta pantalla aparecerán las opciones para la configuración de la conferencia. Los parámetros principales que se deben configurar se muestran dentro del cuadro de color rojo en la Figura 2.28. En la Tabla 2.7 se muestran los valores que deben configurar en estos parámetros y una descripción de cada uno.

Tabla 2.7 Parámetros para configuración de Conferencia

Parámetro	Valor	Descripción
Número de conferencia	600	El número al que deben llamar los usuarios para conectarse a la sesión de la conferencia.
Nombre de la conferencia	Conferencia1	Un nombre para identificar a la conferencia

Adicionalmente se pueden configurar otras opciones para la conferencia, como las que se muestran en el cuadro de color verde de la Figura 2.28. Las opciones que se han habilitado son:

- *Talker Optimization:* Cuando se habilita esta opción los usuarios que no están hablando se los toma como si estuvieran en mute de manera que no se codifica el audio que provenga de estos usuarios.
- *Talker Detection:* Permite identificar el canal por el que se está hablando en la conferencia, es decir que se detecta al usuario que está hablando.
- *Modo silencioso:* No reproducir sonidos cuando alguien entra o sale de la conferencia.
- *Contador de usuario:* Dice el número de usuarios que se encuentran en la conferencia al entrar o salir de la misma.
- *Música en espera: Habilita un sonido de espera mientras la conferencia tiene solo un participante.*

2.13 CONFIGURACIÓN DEL GATEWAY DE VOZ

Lo primero que se realiza es la configuración básica que consiste en dar los parámetros de red, el modo como va a trabajar el servidor NAT/DHCP embebido y los parámetros para reenvío de llamadas hacia la PSTN o hacia la IP-PBX.

En la Tabla 2.8 se listan estos parámetros con sus respectivos valores:

Tabla 2.8 Parámetros para configuración básica del Gateway

Parámetros de Red	**Valores**
Dirección IP	192.168.0.11
Máscara de red	255.255.255.0
Puerta de enlace	192.168.0.1
Servidor DNS	192.168.0.1
Configuración del servidor NAT/DHCP	
Modo del Dispositivo	Puente
Habilitar servidor DHCP	No
Unconditional Call Forward to PSTN:	4999
Unconditional Call Forward to VoIP:	7777@192.168.0.10:5060

Lo siguiente es ir a la sección de parámetros avanzados donde se va a configurar los parámetros de línea telefónica para que el *gateway* pueda reconocer cuando existe una llamada entrante de la PSTN, cuando la línea está ocupada o cuando se debe colgar la llamada y dejar libre la línea como los que se muestran en la Figura 2.29

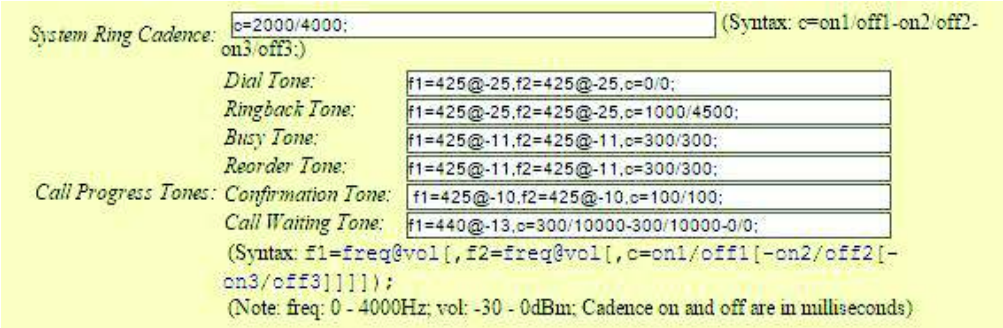


Figura 2.29 Parámetros de la línea telefónica

Estos parámetros corresponden a la recomendación E.180[24] de la ITU[25] correspondientes a los tonos usados en redes nacionales de telefonía. Cabe decir que estos valores se tomaron como referencia y se hicieron varias pruebas para determinar cuáles eran las frecuencias correctas para los diferentes tonos y los tiempos de encendido y apagado, llegando a los valores que se muestran en la Figura 2.29.

El siguiente paso es la configuración del puerto FXO. En la Figura 2.30 se observan los parámetros de autenticación para que el *gateway* se registre en el servidor uElastix.

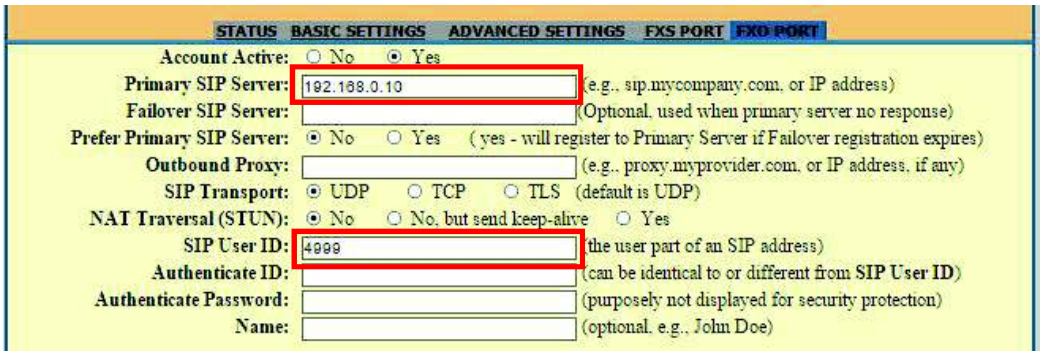


Figura 2.30 Parámetros para registro en el servidor IP-PBX

Como primer paso se ingresan los parámetros para el registro del gateway en el servidor. Estos parámetros corresponden a los señalados anteriormente cuando se configuró la troncal SIP. Aquí se ingresa la dirección IP del servidor uElastix, nombre de la troncal SIP y la contraseña para el registro. En la tabla 2.9 se puede ver en detalle estos parámetros.

Tabla 2.9 Parámetros de registro

Parámetros	**Detalle**
Servidor SIP primario	192.168.0.10

[24] *Tones used in national networks*, ITU-T recommendation E.180, marzo,1998

[25] ITU o UIT por sus siglas en español que significa Unión Internacional de Telecomunicaciones.

Parámetros	Detalle
Protocolo de Transporte	UDP
Usuario SIP	4999 (Nombre de la Troncal)
Contraseña	1234

Luego se configuran ciertos parámetros básicos para el funcionamiento del puerto FXO. Un aspecto importante aquí es los puertos 5060 UDP y el 5012 UTP que corresponden a los protocolos SIP y RTP respectivamente, éstos deben coincidir con los parámetros configurados en el servidor *uELastix* para que la comunicación pueda establecerse sin ningún conflicto por los puertos que se utilicen.

En la Figura 2.31 se puede ver los campos en donde se debe ingresar estos parámetros a más de otros que corresponden a la configuración básica del puerto FXO.

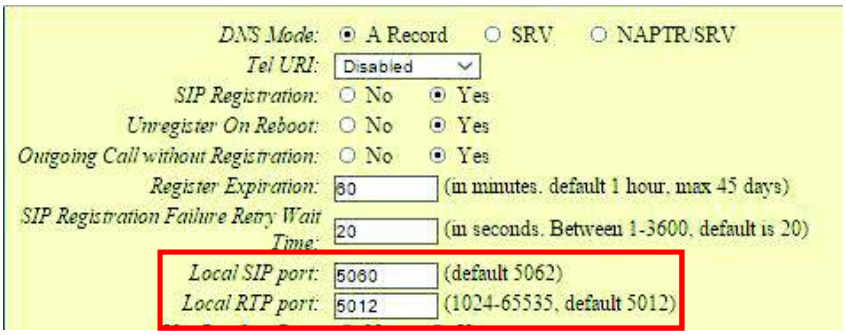


Figura 2.31 Puertos para protocolos SIP y RTP

Los siguientes parámetros corresponden al perfil DTMF que se va a utilizar en el *gateway*. Aquí se selecciona el método RFC2833 tal como está configurado en la troncal SIP, ver sección 2.6. En la Figura 2.32 se observa la configuración de estos parámetros.

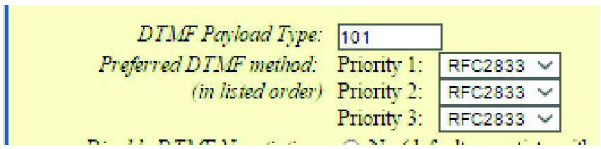


Figura 2.32 Perfil DTMF

A continuación se configuran los códecs de voz que puede manejar el *gateway.* Aquí se colocan en orden de prioridad de acuerdo a los que se configuraron en el servidor. Ya que las extensiones de los usuarios y el cliente SIP utilizan un codec G.711 ley *u,* este códec también se configura en el gateway.

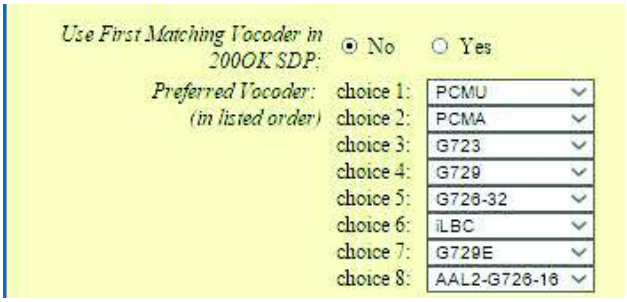


Figura 2.33 Códecs de voz a utilizar

En la Figura 2.33 se observa que la primera opción de códec es PCMU que corresponde al códec G.711 con *ley u*, la siguiente opción es PCMA, correspondiente al códec G.711 con *ley a.*

Finalmente se configuran los parámetros para la terminación de las llamadas en el puerto FXO para que se conecten con la PSTN, como se muestra en la Figura 2.34.

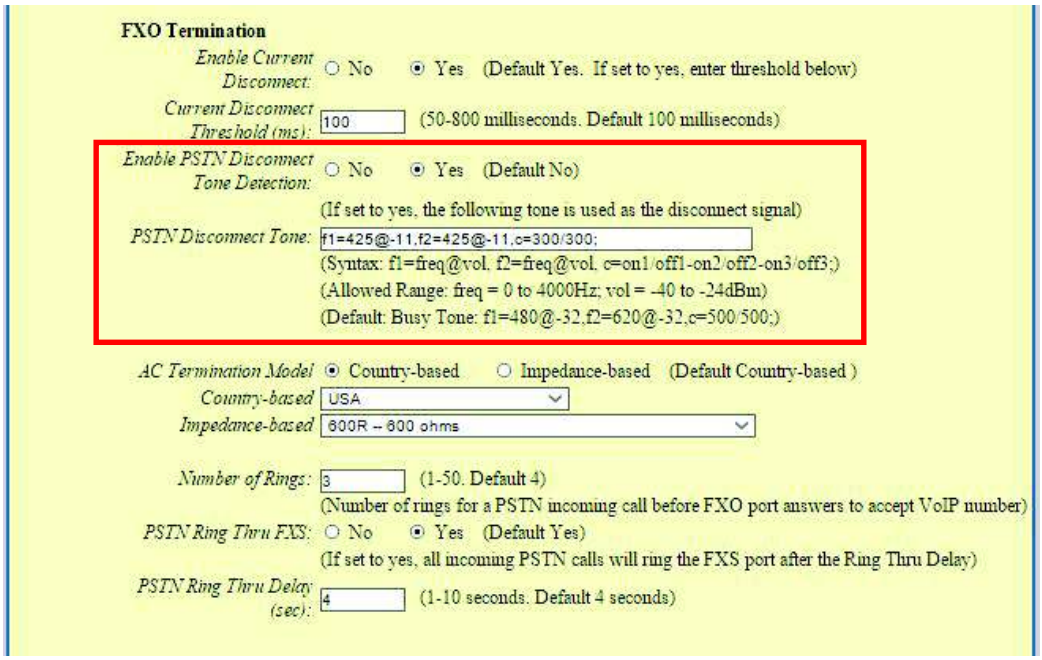


Figura 2.34 Terminación de llamadas en el puerto FXO

Lo más importante aquí es configurar la detección del tono de desconexión de la PSTN para que el *gateway* pueda dejar libre la línea una vez se haya terminado una llamada. Para ello se debe ingresar la frecuencia del tono de desconexión y habilitar esta opción. En la figura 2.34 se observa habilitada la opción de detección del tono de desconexión y la frecuencia del tono utilizando nuevamente la recomendación de la ITU descrita anteriormente.

Realizando los procedimientos descritos anteriormente finalmente queda configurado el *gateway* y se puede revisar en la pestaña de estado si ya se ha registrado en el servidor y si la línea está libre u ocupada por una llamada.

Como se observa en la Figura 2.35 el puerto FXO se encuentra registrado en el servidor y la línea está disponible para realizar llamadas. El puerto FXO no está registrado ya que no se ha configurado ningún dispositivo para que se conecte a ese puerto.

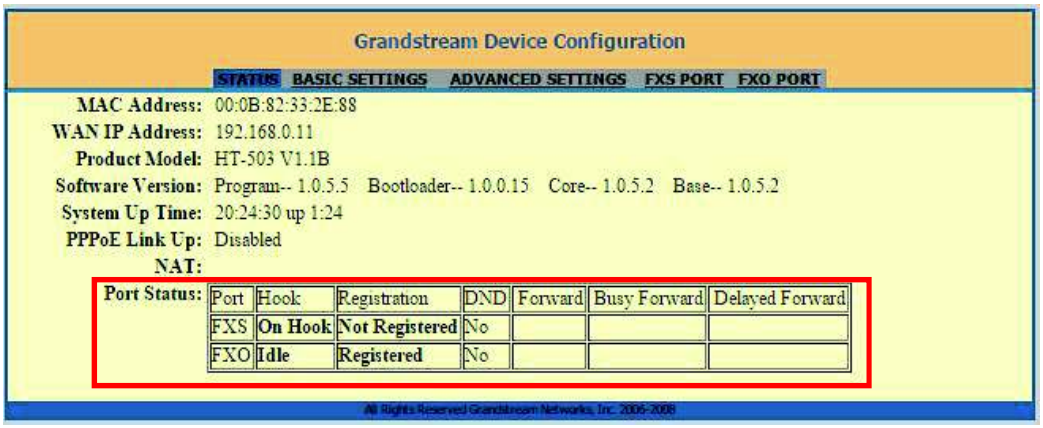


Figura 2.35 Estado del gateway de voz

2.14 CONFIGURACIÓN DE FIREWALL Y FAIL2BAN

En toda implementación de un sistema de comunicaciones sobre el protocolo IP la seguridad es una de las tareas más importantes a tener en cuenta sobre todo en el caso de una IP – PBX que esté conectada al Internet ya que la misma puede ser blanco fácil de muchos tipos de ataques que pueden dañar el sistema, causar fraudes telefónicos o robar información de los usuarios.

Algunas de las amenazas pueden incluir ataques de DoS[26] (Denial of Service), y robo de sesiones o *password cracking*[27] en SSH[28] y sistemas Web, que son parte de las plataformas de VoIP y que incrementan el interés de los atacantes que buscan no solo tener acceso a una base de datos del servidor sino realizar gran cantidad de llamadas telefónicas que podrían costarle miles de dólares a una determinada empresa. Adicionalmente al hablar de Voz sobre IP también está el protocolo SIP y sobre el cual también existen algunas amenazas potenciales como *Eavesdropping* (Escucha no autorizada), DoS en VoIP, *SIP brute force attack* (Ataques de fuerza bruta al protocolo SIP), *VoIP Spam, Caller ID Spoofing*.

Aun cuando el servidor de la central IP – PBX no tenga extensiones remotas que se conecten a través de Internet, por una VPN o vía WAN, se deben restringir las redes IP desde donde se permiten conexiones al servidor, los accesos a la interfaz Web también deben estar limitados y las sesiones SSH, ya que si algún usuario logra tener acceso por cualquiera de estos métodos al servidor, tendrá el control total sobre éste.

El servidor *uElastix* cuenta con un módulo de seguridad basado en las *IPTABLES* que son el *firewall* de los sistemas Linux, que permite realizar una configuración básica para minimizar el riesgo de un acceso indebido al servidor ya sea por SSH o la interfaz web. Adicionalmente se puede complementar la seguridad del servidor con la inclusión de un software llamada *Fail2ban,* el cual es un analizador de logs que monitorea intentos de registro que fallan un determinado número de veces y bloquea de manera definitiva o durante un tiempo determinado por el administrador del servidor, las direcciones IP de donde provienen los ataques de una forma proactiva. *Fail2Ban* es software libre y se distribuye bajo la licencia GNU[29]. Se puede copiar/modificar dentro de los términos que especifica la licencia que está publicada en la *Free Software Foundation*. El veto se realiza actualizando el firewall IPTABLES [48].

---

[26] Denegación de servicio mediante la saturación del ancho de banda de la red o sobrecarga del servidor.
[27] Proceso mediante el cual se intenta adivinar la contraseña de los usuarios.
[28] Secure Shell
[29] GNU General Public License, Declara que el software es libre y garantiza a los usuarios finales la libertad de usar, compartir, copiar y modificar el software.

2.14.1 CONFIGURACIÓN DEL FIREWALL INTERNO DE *uElastix*

La configuración del Firewall se va a realizar para restringir el acceso a la interfaz Web y las conexiones de SSH en el servidor y de manera que el acceso sea unicamente utilizando la dirección IP del administrador de la plataforma. Para ello se siguen los siguientes pasos: [49]

1. **Habilitar el Firewall:** Para activar el Firewall que viene desactivado por defecto en todas las distribuciones de Elastix, se accede al menú *Seguridad >> Cortafuegos >> Reglas de Cortafuegos.* Luego en la pantalla del Firewall que se observa en la Figura 2.36 aparece un mensaje donde se señala que está desactivado. Se da *clic* en "Activar Cortafuegos" en el botón resaltado en rojo.

Figura 2.36 Pantalla principal para configuración del Firewall

Una vez activado el firewall se puede modificar las reglas correspondientes a los puertos de HTTP, HTTPS y SSH.

2. **Modificar las reglas para HTTP, HTTPS y SSH:** Estos puertos corresponden a los servicios Web y de las conexiones por SSH. Se buscan las reglas número 11(SSH), 13(HTTP) y 16(HTTPS).

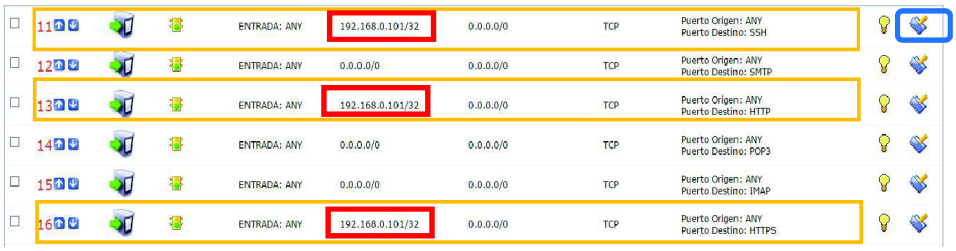


Figura 2.37 Pantalla para modificar reglas del Firewall

Se van a editar las reglas de tal manera que ahora permitan el acceso únicamente desde la **dirección IP 192.168.0.101**, correspondiente al administrador de la plataforma. La primera regla a modificar es la del servicio SSH. Para ello se da *clic* en el ícono encerrado en un círculo azul de la Figura 2.37, el cual lleva a la pantalla de edición de las reglas del firewall.

Figura 2.38 Pantalla de edición de la regla para SSH

Como se observa en la Figura 2.38 se procede a filtrar por la dirección IP de origen, permitiendo de esta manera que únicamente la dirección IP del administrador se conecte al servidor mediante SSH. Cabe recalcar que el protocolo SSH utiliza el puerto 22 TCP para sus conexiones. En la sección DETALLES DEL PROTOCOLO en la Figura 2.38 se selecciona el protocolo (TCP), el puerto de origen de las peticiones que en este caso será cualquier puerto y el puerto destino que será el puerto 22 (SSH). Luego en “Detalle de la acción” se elige aceptar para que permita esta conexión y cualquier conexión desde otra dirección IP queda denegada.

Aplicando el mismo procedimiento se configuran las políticas para HTTP y HTTPS, cambiando el puerto o servicio de destino correspondiente a estos protocolos.

3. **Guardar y aceptar los cambios:** Una vez que se han realizado todos los cambios en las reglas, se guardan los cambios dando click en la opción *Guardar cambios* que aparece en la pantalla principal del *Firewall.*

Figura 2.39 Aceptar los cambios y guardar

En la figura 2.39 se observa cómo quedan los cambios aplicados y se muestra la IP del administrador que es la única que tiene acceso ahora.

2.14.2 CONFIGURACION DE FAIL2BAN [50] [51]

Fail2ban está escrito en el lenguaje Python y para que funciones se debe asegurar que Python esté instalado en el servidor. Para ello se ejecuta el comando "*python*" en la consola de *uElastix* y si se dispone del paquete instalado se accederá a la consola del mismo, caso contrario se observará un mensaje de error.

```
[root@elx ~]# python
Python 2.7.3 (default, Aug 26 2012, 05:20:34)
[GCC 4.7.1 20120720 (Red Hat 4.7.1-5)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figura 2.40 Consola de Python en uElastix

Como se puede ver en la Figura 2.40, sí se tiene instalado el lenguaje Python en el servidor por lo que se puede continuar con la instalación de Fail2ban.

En la versión de *uElastix* que se utiliza para este proyecto el paquete Fail2Ban ya vino instalado previamente por lo cual se pasa directamente a la configuración del mismo.

Los pasos para configurar el servicio de fail2ban son los siguientes:

1. **Modificar el archivo *jail.conf*:** Se ingresa al directorio /etc/fail2ban y se busca el archivo *jail.conf*, se accede al mismo y se busca la sección [ssh-iptables]. Aquí se modifica esa sección para que quede como se muestra en la Figura 2.41.

```
[ssh-iptables]

enabled  = true
filter   = sshd
action   = iptables[name=SSH, port=ssh, protocol=tcp]
           sendmail-whois[name=SSH, dest=dvlux4@gmail.com, sender=fail2ban@servehttp.com, sendername="Fail2Ban"]
logpath  = /var/log/secure
maxretry = 3
```

Figura 2.41 Configuración para SSH

A continuación se explica cada parámetro de la configuración.

- *enable = true:* Habilita el monitoreo.
- *filter = sshd:* Filtro para proteger al servidor OpenSSH
- *action = iptables[name=SSH, port=ssh, protocol=tcp]*: Establece el nombre del servicio, el puerto de comunicación (22 para SSH) y el protocolo sobre el que trabaja.
- Adicionalmente se indica que envíe un correo electrónico a la dirección del administrador cuando haya una dirección IP vetada por varios intentos fallidos de registro.
- *logpath = /var/log/secure*: Es el directorio donde debe buscar los logs para leer y determinar si existe un ataque.
- *maxretry = 3*: Número máximo de intentos de registro antes de ser vetado.

2. De la misma forma se realiza la configuración para el protocolo SIP en asterisk. Los parámetros configurados quedan se observan en la Figura 2.42:

```
[asterisk-iptables]
enabled = true
filter  = asterisk
action  = iptables-allports[name=ASTERISK, protocol=all]
          sendmail-whois[name=ASTERISK, dest=dvlux4@gmail.com, sender=fail2ban@pbx.serverhttp.com]
logpath  = /var/log/asterisk/messages
maxretry = 2
findtime = 21600
bantime = 600
```

Figura 2.42 Configuración para protocolo SIP

La configuración es similar a la realizada para SSH como se observa en la Figura 2.42, se cambia el filtro que para este caso será “asterisk”, se realiza el escaneo de todos los puertos, el número máximo de intentos es 2, y el tiempo que permanece bloqueada la dirección IP atacante (*bantime*) es 600[30] segundos, luego de ese tiempo la dirección IP puede estar disponible nuevamente para conectarse. El directorio para leer los logs es */var/log/asterisk/messages*

[30] Ya que el presente proyecto se trata de un prototipo, se establece este tiempo con la finalidad de realizar pruebas.

3. **Modificar el archivo logger.conf:** Se busca este archivo en el directorio */etc/asterisk/logger.conf*. Se modifica la línea *messages* para que quede de la siguiente forma:

   messages => notice, debug.

4. **Script de arranque con el sistema:** Se configura el servicio fali2ban para que inicie al arrancar el sistema, esto se realiza con el siguiente comando: "*chkconfig fail2ban on",* y se inicia el servicio con "*/etc/init.d/fail2ban start".* En la Figura 2.43 se observa que el servicio de Fail2ban ha iniciado correctamente.

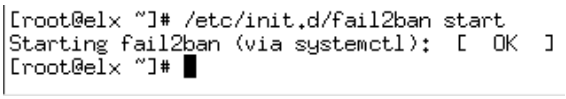


Figura 2.43 Servicio Fail2Ban iniciado correctamente

Luego de realizar las configuraciones de Fail2Ban y una vez que se inicia el servicio, se revisan las IPTABLES para verificar que las políticas aplicadas por Fail2ban se hayan agregado. En la Figura 2.44 se puede observar que ya han sido actualizadas las IPTables en *uElastix.*

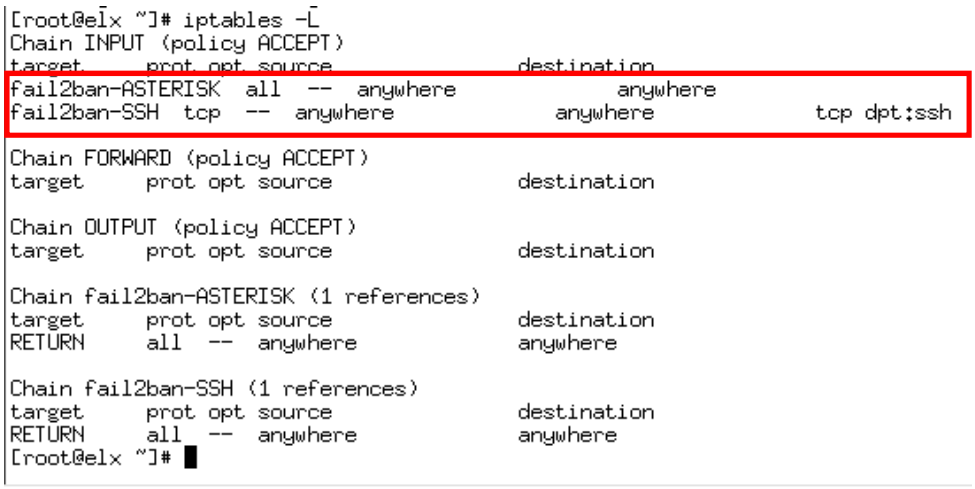


Figura 2.44 IPTables actualizadas con las reglas de Fail2Ban

2.15 CONFIGURACIÓN DEL PUNTO DE ACCESO INALAMBRICO

En esta sección se describe la configuración del router inalámbrico para la infraestructura de telecomunicaciones que se utiliza en este proyecto. Como se

mencionó inicialmente este proyecto utiliza una infraestructura como la que se muestra en la Figura 2.1 al inicio de este capítulo.

El router inalámbrico es el equipo que provee de conectividad entre el servidor IP-PBX, los teléfonos inteligentes con el cliente SIP instalado en ellos y el *gateway* de voz, además brinda acceso a Internet. El router inalámbrico utilizado es un equipo marca Huawei y modelo hg532s.

Tabla 2.10 Parámetros de red para Router inalámbrico

PARÁMETRO	DESCRIPCIÓN	OBSERVACIONES
Red	192.168.0.0	Red en la que se está trabajando
Máscara	255.255.255.0	Máscara de red
IP Router	192.168.0.1	Dirección Ip del router inalámbrico
Rango de direcciones ip reservadas	192.168.0.2 – 192.168.0.19	Direcciones reservadas para servidor IP-PBX, Gateway de voz, equipos de prueba.
Rango de direcciones ip para host	192.168.0.21 – 192.168.0.254	Direcciones disponibles para los clientes que se conecten a la central IP-PBX

Utilizando los parámetros de la Tabla 2.10 se configura el router inalámbrico. Para ello se accede a la interfaz web de configuración y una vez allí se dirige hacia el menú y se ingresa a la configuración de red siguiendo la ruta *Basic >> LAN.* En la Figura 2.45 se observa la interfaz web de configuración y los parámetros de red que se han ingresado.

En cuanto a la interfaz inalámbrica este router trabaja con el protocolo IEEE 802.11b/g/n, el protocolo de seguridad es WPA – PSK / WPA2-PSK y la encriptación AES.

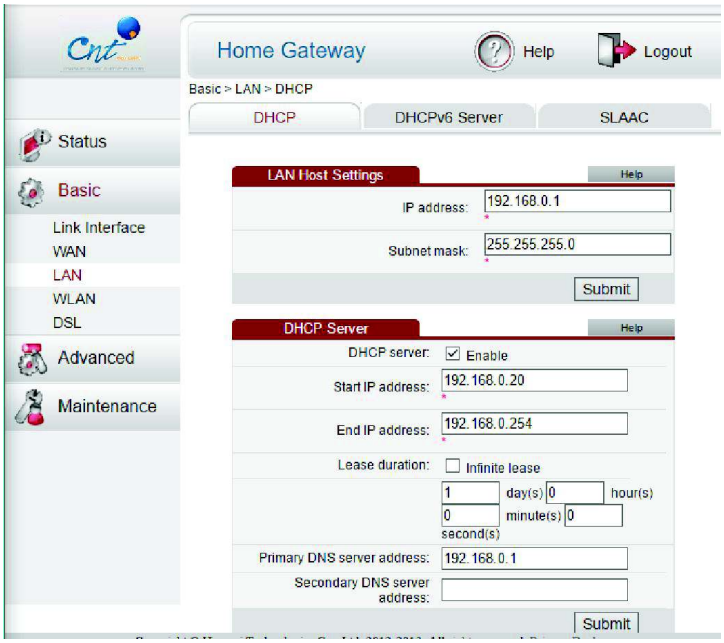


Figura 2.45 Configuración de router inalámbrico

Los parámetros de configuración de la interfaz inalámbrica se describen en la Tabla 2.11.

Es recomendable que en una instalación en un lugar fijo como una oficina de tipo SOHO[31] o casa se realice un *site survey* de manera que se pueda determinar el canal inalámbrico con menor interferencia. En este proyecto no se realiza este paso ya que no se tiene un lugar fijo donde se vaya a instalar el prototipo.

Tabla 2.11 Parámetros de la interfaz inalámbrica del router

Interfaz Inalámbrica	
Parámetro	Descripción
Modo	802.11b/g/n
Canal	11
Potencia de transmisión	20 dBm
SSID	HPELÁEZ

[31] SOHO: Small Office Home Office

Interfaz Inalámbrica	
Máximo número de dispositivos conectados	32
Seguridad	WPA – PSK / WPA2-PSK
Método de encriptación	AES

2.16 FUNCIONALIDADES ADICIONALES DE LA IP PBX

El software uElastix también permite tener algunas funcionalidades adicionales que se pueden habilitar o deshabilitar según sea la necesidad de los usuarios del a central IP PBX. Estas funcionalidades se utilizan en base a algunos códigos como los que se muestran en la Figura 2.46.

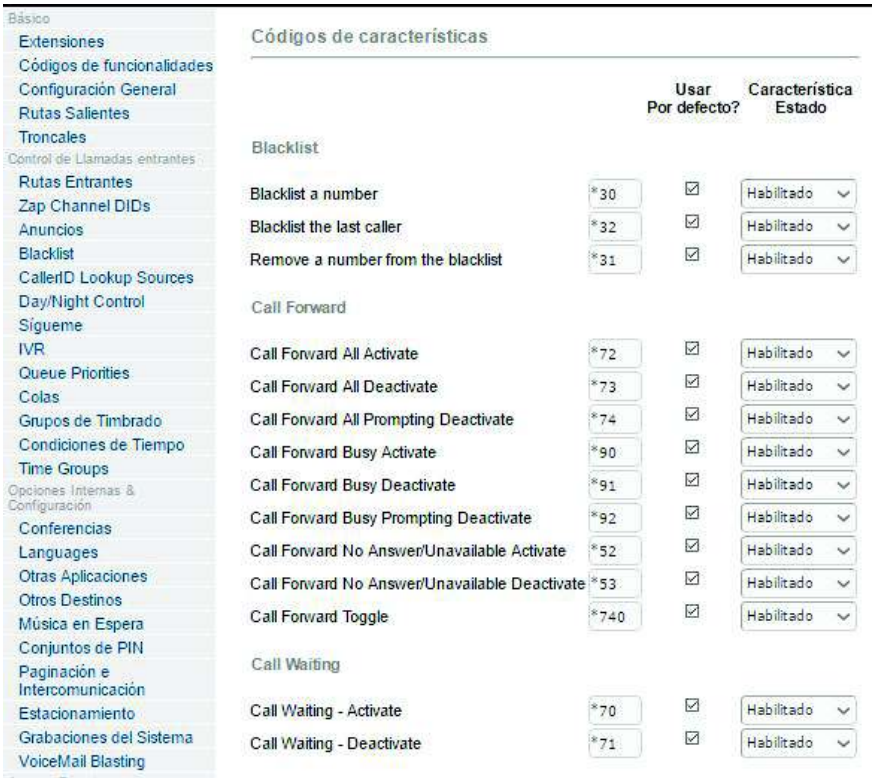


Figura 2.46 Códigos de funcionalidades

Para habilitar o deshabilitar estas funcionalidades se ingresa al *menú PBX >> Configuración de PBX >> Códigos de funcionalidades*.

Algunas de las funcionalidades que se pueden habilitar están las correspondientes al desvío de llamadas:

- Call Forward All Activate: Activa el desvío de llamadas a otra extensión.
- Call Forward Busy Activate: Activa el desvío de llamadas a otra extensión cuando la extensión del usuario se encuentra ocupada.

Otras funcionalidades que se pueden activar se observan en la Figura 2.47

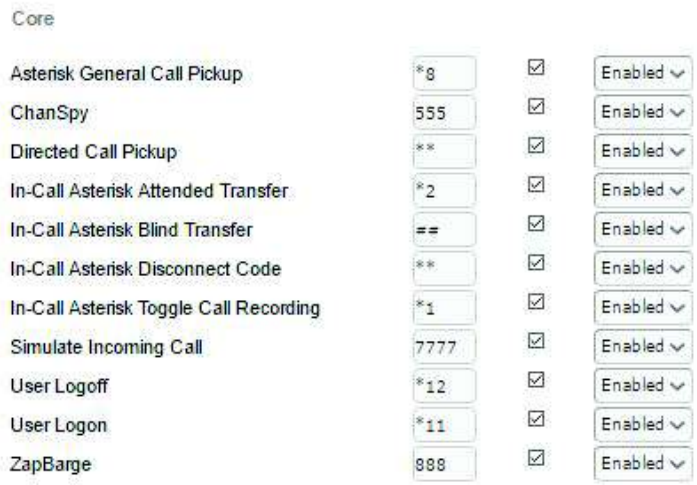


Figura 2.47 Funcionalidades adicionales de IP PBX

Entre las funcionalidades de este grupo están:

- Directed Call Pickup: Captura de la llamada de una extensión que se encuentre sonando.
- In-Call Asterisk Attended Tranfer: Transferir una llamada con una confirmación previa.
- In-Call Asterisk Blind Tranfer: Transferir una llamada sin confirmación previa.
- Simulate Incoming Call: Simula una llamada entrante a la IP PBX.

CAPÍTULO 3

3 PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS OBTENIDOS

En este capítulo se muestra la realización de las pruebas de funcionamiento de la central IP – PBX. Estas pruebas cubren varios aspectos del funcionamiento del sistema como: los servicios configurados en la central IP PBX, el rendimiento del sistema en cuanto al uso de la CPU y memoria RAM, la medida de la opinión de los usuarios sobre la calidad del sistema telefónico, y finalmente las pruebas de los elementos de seguridad implementados en el sistema.

3.1 PRUEBAS DE ACEPTACIÓN DEL SISTEMA TELEFÓNICO

En la Tabla 3.1 se detallan las pruebas de funcionalidad de los servicios que se han configurado en el desarrollo del prototipo. Se realizaron pruebas utilizando el softphone, se realizaron llamadas entre extensiones y hacia a la PSTN para determinar si las mismas fueron exitosas o fallaron de acuerdo al protocolo de pruebas descrito en la Tabla 3.1.

Tabla 3.1 Pruebas de aceptación del sistema telefónico

Prueba No.	Descripción de la Prueba	Criterio de Aceptación	Resultado	No Aplica
1	Los teléfonos obtienen una extensión:	Verificar número de extensión	Pasó	
2	Realizar llamada desde extensión IP a la PSTN destino Servicios Públicos.			
2.1	Servicio de Emergencia 911	La llamada se completa exitosamente.	Pasó	
2.2	Servicios de CNT	La llamada se completa exitosamente.	Pasó	
3	Pruebas de Saturación de canales con llamadas salientes	Se realizará 1 llamada	Pasó	
4	Pruebas de Saturación de canales con llamadas entrantes	Se realizará 1 llamada simultánea a una extensión	Pasó	
5	Verificar las funciones básicas de la central IP-PBX en el cliente SIP instalado (Zoiper)			
5.1	Aceptar las llamadas entrantes		Pasó	

Prueba No.	Descripción de la Prueba	Criterio de Aceptación	Resultado	No Aplica
5.2	Funciona correctamente el altavoz		Pasó	
5.3	Permite llamada en espera		Pasó	
5.4	Permite captura de llamada	Con una llamada entrante a una extensión marcar ** desde otra extensión y contestar la llamadas	Pasó	
5.5	Verificar opción de transferencia de llamadas	Con una llamada en curso desde una extensión marcar ## y transferir la llamada a otra extensión.	Pasó	
5.6	Historial de llamadas	El softphone permite ver un registro de todas las llamadas que se han realizado	Pasó	
5.7	Funciona correctamente la opción mute		Pasó	
6	Comprobar servicio de múltiples llamadas simultánea en Teléfono IP. Teniendo una llamada activa en un Teléfono IP generar una nueva llamada a la misma extensión	El sistema advierte de una nueva llamada entrante cuando se tiene una llamada activa.	Pasó	
7	**Pruebas de operadora.**			
7.1	Realizar llamada al número piloto del IVR.	Se debe escuchar el mensaje de la operadora	Pasó	

3.2 PRUEBAS DE RENDIMIENTO DEL SISTEMA

En esta sección se realizaron las pruebas para evaluar el rendimiento del sistema de acuerdo a parámetros como el uso del CPU, uso de memoria RAM y tiempos de respuesta de llamadas y llamadas fallidas. Estas pruebas se realizaron con la

intensión de comprobar el impacto que tienen las tareas de digitalización, codificación y decodificación de la voz en el rendimiento del sistema.

La metodología que se ha seguido consiste en implementar un ambiente de laboratorio donde el servidor de telefonía IP está conectado a un terminal que simula un *softphone* con el objetivo de determinar el límite en el cual el sistema queda sobrecargado. Se ha utilizado la herramienta SIPp para generar llamadas dinámicamente, a través del servidor IP-PBX instalado en la Raspberry Pi.

Para evaluar el rendimiento de la tarjeta Raspberry Pi como plataforma hardware para la central IP-PBX se ha realizado una prueba de estrés en base a la carga generada sobre la Raspberry Pi por la codificación y decodificación de voz. Se han utilizado diferentes códecs de voz y se han simulado varias llamadas a través del servidor de telefonía para determinar el impacto que tienen estas tareas en el consumo del CPU y memoria RAM. Al tener el procesador del sistema sobrecargado, se evidencia que empiezan a existir retardos en los procesos que se están ejecutando en ese momento, incluso algunos de ellos pueden detenerse por completo y finalizarse. Es por ello que se han tomado en cuenta dos parámetros adicionales como el tiempo de respuesta o tiempo que le toma a la central IP-PBX para contestar una llamada, y las llamadas fallidas. Los tiempos de respuesta se incrementan cuando la carga sobre el procesamiento del sistema aumenta y en última instancia las llamadas empiezan a fallar cuando el sistema está sobrecargado.

3.2.1 METODOLOGÍA DE EVALUACIÓN Y HERRAMIENTAS

Para medir las capacidades del sistema telefónico, se utiliza la Raspberry Pi modelo B que se encuentra conectada a un router donde también se conecta una laptop desde donde se inicia la prueba de estrés hacia el sistema. Se utiliza la herramienta SIPp instalada en una laptop como cliente SIP para generar automáticamente varias llamadas para sobrecargar al sistema telefónico. [52]

SIPp es una herramienta que trabaja con el protocolo SIP y es capaz de generar mensajes SIP para iniciar y terminar una llamada de voz a través de una IP-PBX. Adicionalmente SIPp puede enviar paquetes multimedia de voz o video cuando la llamada ha sido establecida. La herramienta también entrega un reporte de las estadísticas generadas durante y al finalizar las pruebas.

La metodología utilizada para realizar las pruebas consiste en generar un máximo de 200 llamadas desde SIPp hacia la IP-PBX. Las llamadas son generadas en pasos de 2 llamadas simultáneas, cada una con una duración alrededor de 10 segundos. La prueba finaliza cuando se hayan generado las 200 llamadas.

Durante cada llamada se envía contenido multimedia de voz, el mismo que fue generado previamente usando un softphone y capturando el contenido de la llamada con Wireshark[32], usando diferentes codecs de voz como G.711, GSM, G.722 e iLBC. Por cada códec se realizaron 9 pruebas, modificando el número de llamadas simultáneas desde 2 hasta 18 en intervalos de 2. En la Figura 3.1 se puede ver un gráfico que representa el flujo de llamada que se siguió para la realización de esta prueba.

Del lado de la central IP-PBX se ha configurado una extensión de prueba con un flujo de llamada simple que se ejecuta cada vez que se recibe una nueva llamada.

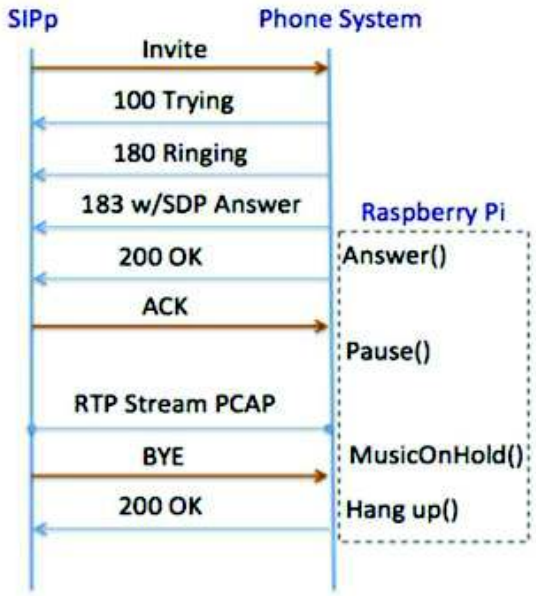


Figura 3.1 Flujo de llamada para prueba de rendimiento

El flujo que se sigue consiste básicamente en contestar la llamada, luego se reproduce sonido durante 10 segundos para finalmente colgar la llamada.

[32] WireShark es un software que permite capturar paquetes que circulan por una red y analizar algunos protocolos de comunicación.

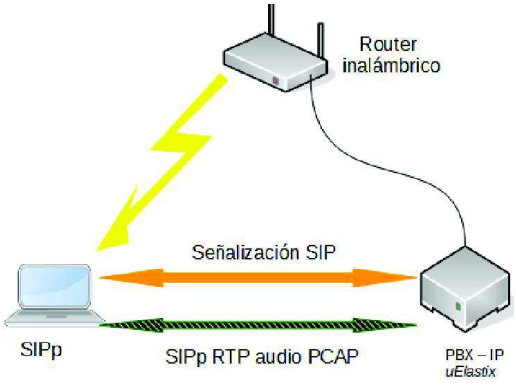


Figura 3.2 Diagrama de red para pruebas de rendimiento

En la Figura 3.2 se observa el esquema del escenario de laboratorio utilizado para estas pruebas.

3.2.2 RESULTADOS PRUEBAS DE RENDIMIENTO DEL SISTEMA

Los resultados de las pruebas de rendimiento se analizan en base a los datos de tráfico obtenidos desde SIPp y de las mediciones del consumo de CPU y memoria RAM obtenidas desde la herramienta *htop*[33] instalada en el servidor IP-PBX.

En base a las mediciones realizadas se puede determinar que la utilización de los recursos del hardware (Procesador de la Raspberry Pi) no depende solamente del códec de voz que se utilice para la compresión de la voz, sino que además depende de la carga que se añada al sistema en función del número de llamadas simultáneas. Se realizaron mediciones del uso del CPU y la memoria RAM para cada uno de los códecs luego de ejecutar las 9 pruebas incrementando el número de llamadas como se menciona en la sección anterior.

3.2.2.1 Uso de CPU y memoria RAM

En la Figura 3.3 se aprecia la variación en el porcentaje de uso del CPU en función del número de llamadas concurrentes. Aquí se puede evidenciar de manera general como el porcentaje de consumo del CPU se incrementa junto con el incremento en el número de llamadas simultáneas, independientemente del códec de voz que se utilice. Se puede señalar que en términos del consumo de CPU el códec GSM es el que menos recursos consume, llegando a un porcentaje máximo del consumo de CPU de 61.3% con 18 llamadas concurrentes como se evidencia en la Tabla 3.2; a

[33] Htop es un visualizador de procesos interactivo en modo texto para sistemas operativos Linux

diferencia de iLBC y G.729 que requieren de mayor procesamiento y por ende consumen más recursos del CPU. En el caso del códec G.729 se llega a un 71.7% de consumo del CPU con 16 llamadas concurrentes, mientras que en el caso de iLBC se llega a un 61.1% de consumo del CPU con 10 llamadas concurrentes. Estos datos son razonables dada la buena calidad de audio que ofrecen estos códecs.

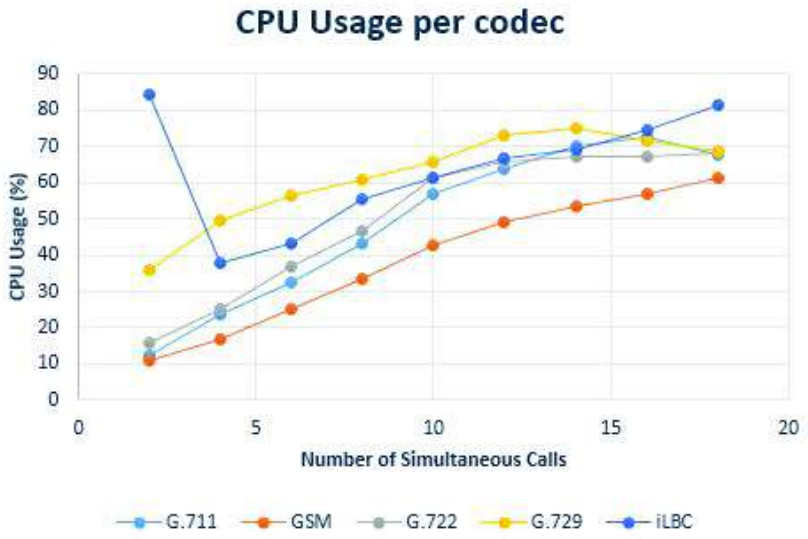


Figura 3.3 Porcentaje de uso del CPU por cada códec

En el caso de G.711, aun cuando este códec no consume mayores recursos del CPU dado que no hace poca compresión de voz, éste no obtiene los mejores resultados en este aspecto. Con G.711 el porcentaje del procesador pasa del 60% llegando a un máximo de 72.3% con 16 llamadas consecutivas.

Tabla 3.2 Valores de porcentaje de consumo de CPU y memoria RAM

Llamadas simultáneas	G.711		GSM		G.722		G.729		iLBC	
	% CPU	% RAM	% CPU	% RAM	% CPU	% RAM	% CPU	% RAM	% CPU	% RAM
2	12.3	7.1	10.8	6.3	15.7	6.3	35.8	6.4	84	6.4
4	23.5	7.1	16.6	6.5	24.9	6.5	49.7	6.6	37.9	6.6
6	32.3	7.1	24.9	6.7	36.8	6.7	56.3	6.8	43.1	6.8
8	43.2	7.2	33.2	6.9	46.8	6.9	60.9	7	55.3	7
10	56.8	7.4	42.6	7.1	61.2	7.1	65.9	7.2	61.1	7.4
12	63.7	7.7	49.2	7.2	65.5	7.3	73.2	7.4	66.8	7.5
14	69.9	7.8	53.7	7.4	67.3	7.5	74.9	7.6	69.1	7.7
16	72.3	7.9	56.7	7.6	67.1	7.4	71.7	7.8	74.6	7.4
18	67.6	7.8	61.3	7.7	67.9	7.5	68.8	7.8	81.2	7.6

En cuanto al consumo de memoria RAM, los resultados que se observan en la Figura 3.4 muestran que la Raspberry Pi necesita menos del 10% del total de memoria RAM disponible (512 MB) para mantener 18 llamadas consecutivas.

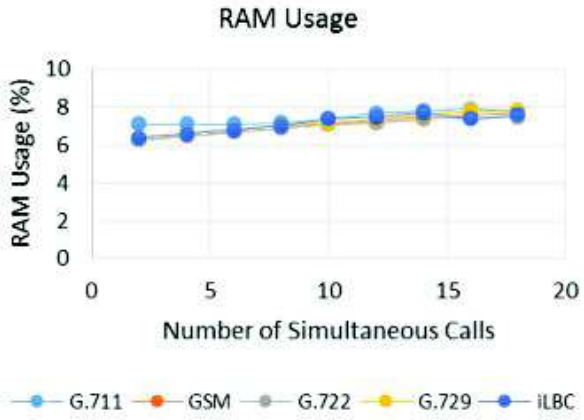


Figura 3.4 Porcentaje de utilización de memoria RAM por cada códec

3.2.2.2 Tiempos de respuesta y llamadas fallidas

De acuerdo a los datos generados por la herramienta SIPp, es posible obtener información del tiempo de respuesta que le toma a la central IP PBX en responder una llamada entrante.

Como se observa en la Figura 3.5 para un número dado de llamadas entrantes a la central IP PBX, el tiempo de respuesta se incrementa hasta llegar a un punto donde logra estabilizarse.

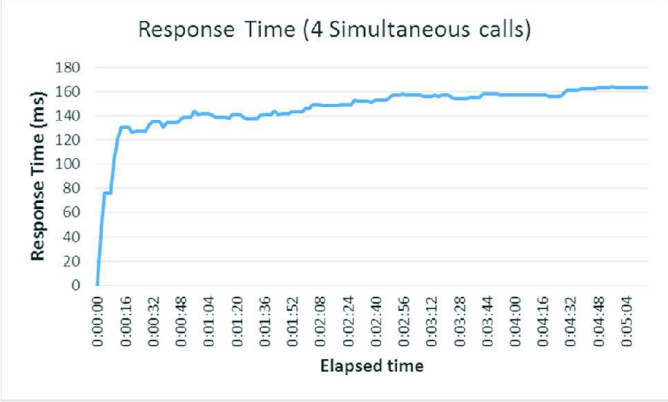


Figura 3.5 Variación del tiempo de respuesta para 4 llamadas simultáneas usando G.729

Por otro lado, cuando el número de llamadas entrantes sobrecarga al sistema el tiempo de repuesta varía de manera errática como se aprecia en la Figura 3.6. Aquí el tiempo de respuesta puede aumentar tanto que algunas llamadas ya no pueden ser procesadas.

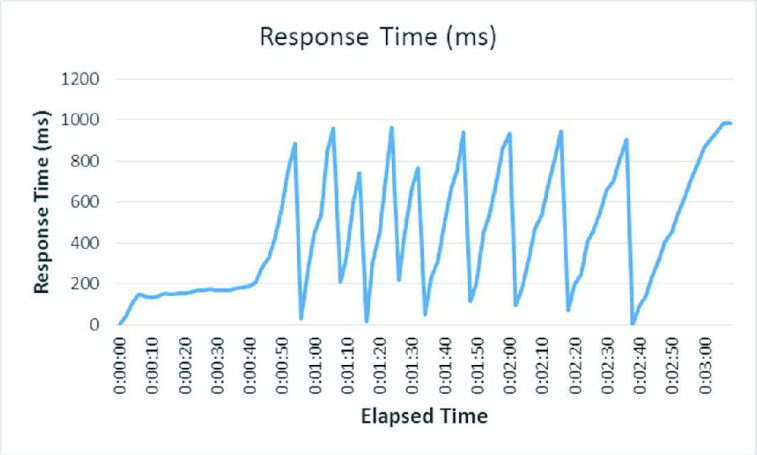


Figura 3.6 Variación del tiempo de respuesta para 16 llamadas simultáneas usando G.729

En la Figura 3.7 se observa cómo varía el tiempo de respuesta que le toma a la central IP-PBX contestar una llamada en función del número de llamadas simultáneas para cada uno de los códecs utilizados. Aquí se observa que utilizando iLBC, el tiempo que le toma a la central IP-PBX en responder una llamada es mayor en comparación con los demás códecs utilizados.

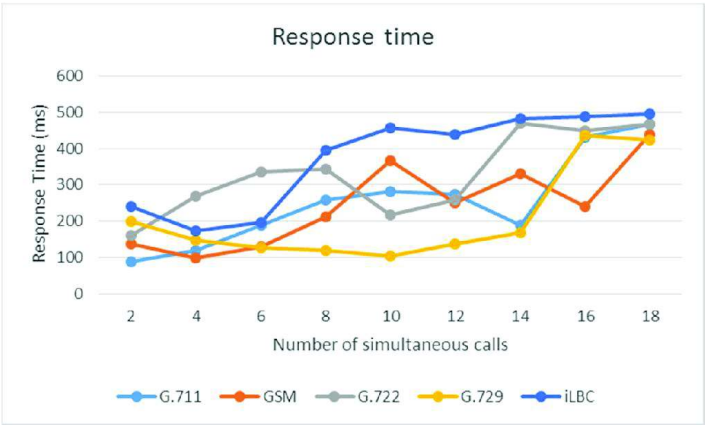


Figura 3.7 Variación del tiempo de respuesta para cada códec en función del número de llamadas simultáneas

Aquellos códecs que presentan mayor porcentaje de uso del CPU para el procesamiento son los que provocan que mayor cantidad de llamadas fallen. Como se observa en la Figura 3.8, con iLBC las llamadas empiezan a caerse cuando se tienen 10 llamadas simultáneas.

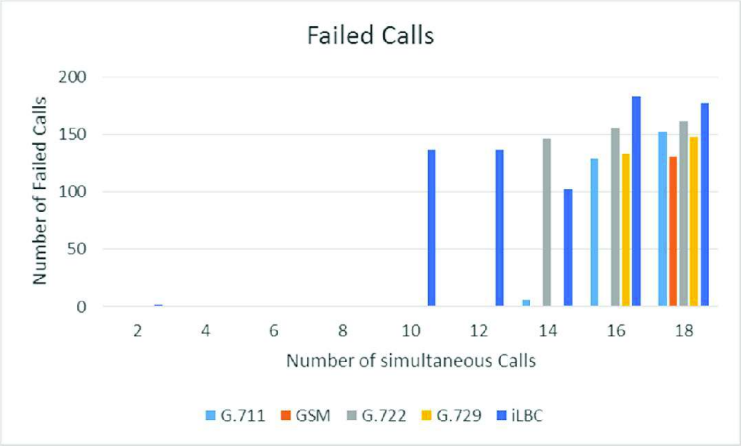


Figura 3.8 Llamadas fallidas por cada códec en función del número de llamadas simultáneas

En las tablas contenidas en el Anexo C se encuentra el resumen de los datos tomados de todos los parámetros que se tienen en cuenta para realizar el análisis de rendimiento del sistema telefónico.

De acuerdo a los datos obtenidos se puede establecer que cuando el sistema trabaja con un códec GSM puede soportar el tráfico de 16 llamadas simultáneas con un porcentaje de utilización del CPU de 56.7% (Ver Tabla 3.2).

3.2.3 PRUEBA MOS – NOTA MEDIA DE OPINIÓN

La Nota Media de Opinión o MOS[34] por sus siglas en inglés se define en la UIT-T P.800.1 como *"el valor de una escala predefinida que un sujeto asigna a su propia opinión sobre la calidad de funcionamiento del sistema de transmisión utilizado para una conversación o únicamente para una escucha de material hablado"* [53]. Este método subjetivo de evaluación puede ser utilizado en diferentes áreas de aplicación. Algunas de estas áreas se identifican a través de una letra junto a las siglas MOS, así se utiliza la letra N para banda estrecha (narrow band), W para banda ancha (wide band), LQ para calidad de la escucha (listening Quality), CQ

[34] Mean Opinion Score

para calidad de la conversación (conversational quality), S para una estimación Subjetiva, O para una estimación Objetiva y E para un valor calculado. Para el caso del presente proyecto el área de aplicación de la prueba MOS es de calidad de la conversación MOS-CQS.

El esquema de pruebas consiste en la central telefónica IP implementada con uElastix. El sistema está cargado con 4 llamadas simultáneas generadas con SIPp y que comprimen la voz utilizando el códec GSM. Dos smartphones con la aplicación Zoiper instalada como cliente SIP se conectan a la Raspberry Pi a través de un router inalámbrico como se observa en el esquema de la Figura 3.9.

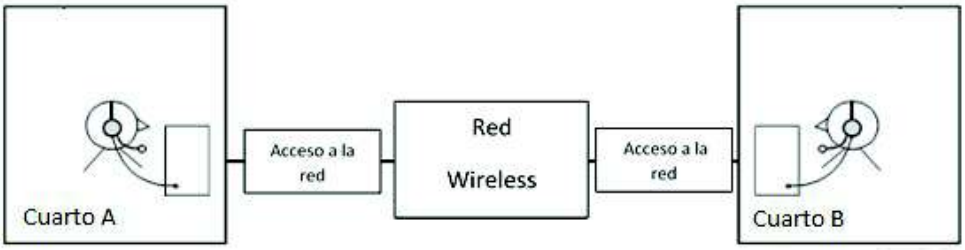


Figura 3.9 Esquema de laboratorio para prueba MOS

Cada prueba tiene un tiempo de 3 minutos en la que dos usuarios ubicados en diferentes habitaciones mantienen una conversación telefónica en la que se les ha pedido que sigan un esquema establecido. Las instrucciones de este esquema se pueden ver en el Anexo D donde se encuentran las recomendaciones dadas por la UIT para este tipo de pruebas y el material que se utiliza [54]. Luego de finalizar la conversación los participantes califican su experiencia mediante una encuesta siguiendo los parámetros establecidos en la recomendación ITU-T P.805 [55].

La metodología de evaluación para esta prueba consiste en realizar un grupo de preguntas a los participantes quienes responden con una calificación entre 1 (mala calidad o mala experiencia) y 5 (excelente calidad).

Para la estimación del número de entrevistas a realizar para la prueba de MOS se considera que debido a que el sistema está pensado para la pequeña y mediana empresa y que de acuerdo a la normativa implantada por la Comunidad Andina en su resolución 1260 [56], se clasifica a las PYMES según la Tabla 3.4.

Basado en esta clasificación se toma el caso de una mediana empresa en la que se dispone de un universo de 100 personas.

Tabla 3.3 Clasificación de las PYMES en el Ecuador

Variables	Micro Empresa	Pequeña Empresa	Mediana Empresa	Grandes Empresas
Personal Ocupado	1 - 9	10 - 49	50 - 199	> 200
Valor Bruto de las ventas anuales US $	< 100.000	100.001 - 1.000.000	1.000.001 - 5.000.000	> 5.000.000
Montos Activos	Hasta US $ 100.000	De US $ 100.001 hasta US $ 750.000	De US $ 750.001 hasta US $ 3.999.999	> US $ 4.000.000

El cálculo del tamaño de la muestra se realiza aplicando la siguiente fórmula [57]:

$$n=\frac{N\cdot Z^2\cdot \sigma^2}{(N-1)\cdot e^2+Z^2\cdot \sigma^2} \qquad [3.1]$$

Donde:

- n = El tamaño de la muestra que se desea calcular.
- N = Tamaño del universo.
- Z = Es la desviación del valor medio que se acepta para lograr el nivel de confianza deseado para el cálculo de la muestra y se ha establecido en 95%, por lo que Z va a ser igual a 1,96 [57].
- e = Es el margen de error máximo permitido. (± 1)
- σ^2 = Es la varianza que se espera encontrar en la población.

Como ya se mencionó anteriormente se tiene un universo de 100, con un margen de error del ± 1 y una varianza de 10[35]. Se reemplazan estos datos en la fórmula 3.1 y se tiene:

$$n=\frac{100\cdot 1{,}96^2\cdot 10}{(100-1)\cdot 1+1{,}96^2\cdot 10}=28$$

Por lo tanto se realiza un total de 28 encuestas con el modelo de preguntas que se muestra en el Anexo D.

La prueba de opinión se realiza en base al códec GSM que es aquel que presenta menor porcentaje de consumo del CPU y menor número de llamadas fallidas. Como resultado de esta prueba se obtuvo una puntuación de 3.917 (Buena calidad). Este

[35] Medida de la variabilidad de la población

resultado es consistente con los valores para la prueba MOS encontrados en la literatura sobre este tema [58].

Tabla 3.4 Resultados de prueba MOS

MOS	MÍNIMO	3.22
	PROMEDIO	3.917
	MÁXIMO	4.533

De acuerdo a la información mostrada en la Tabla 3.4, la prueba MOS refleja resultados satisfactorios. Ya que la calidad de la voz se ve afectada por los retardos en la transmisión y que la interfaz inalámbrica que se utilizó también impacta en la calidad de voz, el resultado obtenido está dentro de lo esperado.

3.3 PRUEBAS DE SEGURIDAD DEL SISTEMA

En esta sección se presentan las pruebas realizadas y los resultados obtenidos de la evaluación de los mecanismos de seguridad implementados en la central IP-PBX. Los mecanismos de seguridad que se implementan son para proteger a la central IP-PBX contra ataques de fuerza bruta a los protocolos SIP y SSH mediante la modalidad de *password cracking*[36]. También se busca evitar las conexiones remotas a la IP-PBX usando los protocolos HTTP[37], HTTPS[38] y SSH.

Inicialmente se realizan las pruebas sin ningún tipo de seguridad y luego se repiten las pruebas con los mecanismos de seguridad funcionando, para poder realizar una comparación y determinar si las políticas implementadas funcionan como se espera.

El esquema que se utiliza para realizar estas pruebas, y que se puede observar en la Figura 3.10 consiste en la central IP-PBX conectada a un router inalámbrico y como elemento atacante se utiliza una Raspberry Pi 2 modelo B con un sistema operativo apropiado para *pentesting* como es *Kali Linux* que también cuenta con una versión para plataformas ARM compatible con la Raspberry Pi [59]. La versión ARM contiene las herramientas básicas para pruebas de penetración, pero de ser necesario es posible descargar todas las herramientas disponibles en la versión de escritorio.

---

[36] Realizar intentos repetitivos de registro tratando de adivinar la contraseña de una extensión.
[37] Hypertext Transfer Protocol.
[38] Hypertext Transfer Protocol Secure

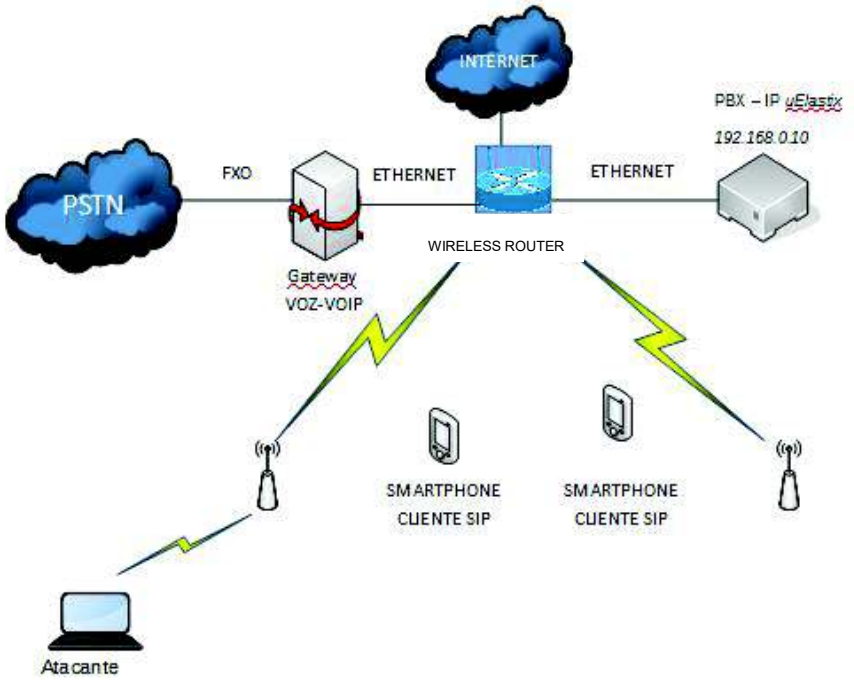


Figura 3.10 Esquema de red para laboratorio de pruebas de seguridad

Las herramientas que se utilizan en estas pruebas se describen a continuación:

- Nmap: “Network Mapper” es una herramienta de código abierto utilizada para recopilar información de una red, para esto envía diferentes tipos de paquetes a cada puerto y en función de cada respuesta Nmap recopila información de la red o host al que ataque.
- Sipvicious: Son un conjunto de herramientas que pueden ser utilizadas para auditar sistemas de VoIP, se utilizan las siguientes: svmap (escaneo de protocolo SIP), svwar (Identifica extensiones de una PBX), svcrack (Password craking).
- Hydra: Es una herramienta que permite asociar un usuario con una contraseña generada mediante un diccionario. Soporta numerosos protocolos a los que se puede atacar como SIP, SSH, HTTPS entre otros.
- Wireshark: Es un analizador de protocolos de red, con esta herramienta se puede capturar cada paquete que atraviese la red para luego poder analizarlo.

Como se menciona al inicio de esta sección, las pruebas realizadas constan de dos partes. En la primera parte se realizan algunos ataques a la central para identificar cuáles son los puertos abiertos y desprotegidos de la central IP-PBX para posteriormente realizar intentos de conexión a estos puertos y ataques a los protocolos SIP y SSH. La segunda parte de estas pruebas consiste en repetir los ataques, pero esta vez con el Firewall IP-TABLES y Fail2Ban habilitados.

3.3.1 PRUEBAS DE ATAQUE A LA CENTRAL IP-PBX SIN HABILITAR LOS MECANISMOS DE SEGURIDAD

La primera prueba de este grupo consiste en realizar un escaneo de la red en busca de alguna central IP-PBX disponible. Para ello utilizamos la herramienta “svmap” disponible en la suit de sipvicius. En la Figura 3.11 se observa que al realizar el escaneo de la red se descubre la existencia de la central IP-PBX activa y la herramienta devuelve en el resultado la dirección IP (192.168.0.10) y el puerto con el que trabaja (5060), junto con el *User Agent.*

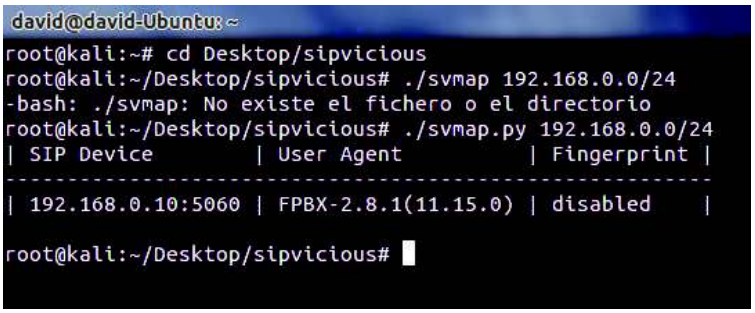


Figura 3.11 Escaneo de red con sipvicious en busca de una IP-PBX activa

Luego de realizar este escaneo de la red se puede utilizar “nmap” para descubrir cuáles son los puertos abiertos que pueden ser vulnerables a ataques. En la Figura 3.12 se observa que al realizar un escaneo de puertos en la central IP-PBX se descubren todos los puertos y servicios habilitados.

De este grupo de servicios, los que más interesan proteger son los que se encuentran en los recuadros amarillos dentro de la Figura 3.12, éstos son: SSH (puerto 22), HTTP (puerto 80), HTTPS (puerto 443), ya que estos servicios son los que pueden presentar más vulnerabilidades ante ataques o intentos de conexiones remotas.

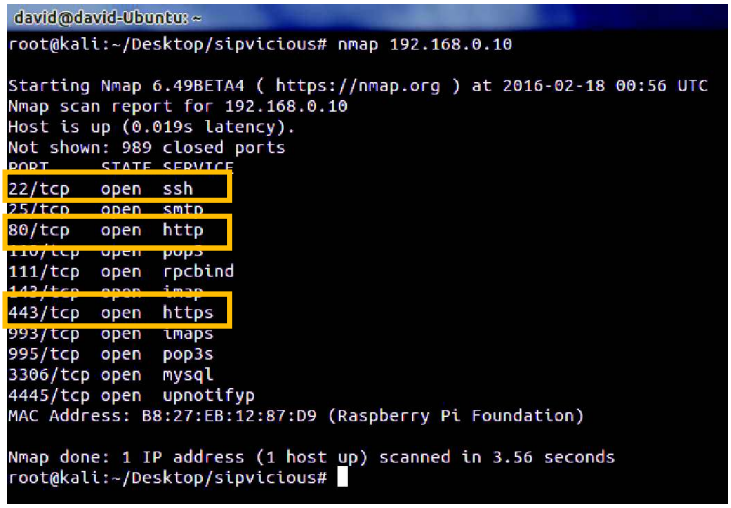


Figura 3.12 Escaneo de puertos abiertos usando nmap

Ya que se ha identificado esta vulnerabilidad es posible intentar un ataque a alguno de estos protocolos. Para realizar una prueba se elige a SSH como blanco de ataque. Mediante la utilización de "Hydra" y con la ayuda de un diccionario[39] es posible intentar un ataque. El método de este ataque consiste en realizar una serie de intentos de registro utilizando un usuario previamente dado y las contraseñas que se encuentran en el diccionario. Cuando una de estas contraseñas coincide con la configurada para el usuario "root" se muestra una respuesta con el usuario y la contraseña para el registro como se muestra en la Figura 3.13 dentro del recuadro amarillo.

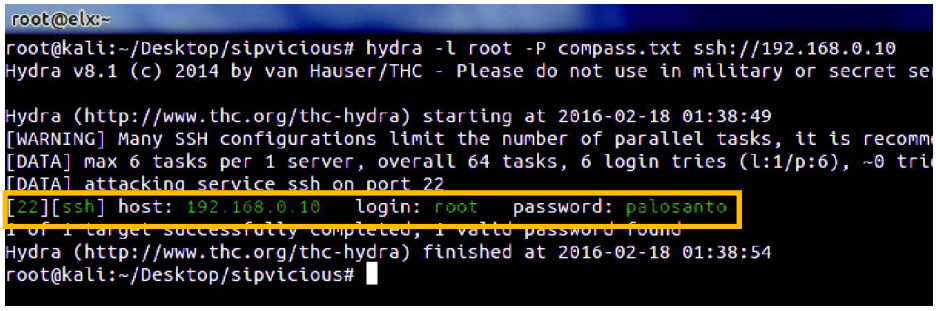


Figura 3.13 Ataque de password cracking con Hydra y sin implementar los mecanismos de seguridad

[39] Archivo de texto plano con una lista de contraseñas comunes

La siguiente prueba que se realiza es un ataque de fuerza bruta al protocolo SIP. El objetivo de este ataque es descubrir cuáles son las extensiones habilitadas en la central IP-PBX e identificar las que requieren autenticación.

En la Figura 3.14 se observa que utilizando la herramienta “svwar” se realiza un ataque hacia la IP-PBX identificada a través de su dirección IP (192.168.0.10) y como resultado se observa una lista con todas las extensiones configuradas en la central IP-PBX y adicionalmente una descripción de aquellas que requieren autenticación (*reqauth*) y aquellas que no lo requieren (*weird*).

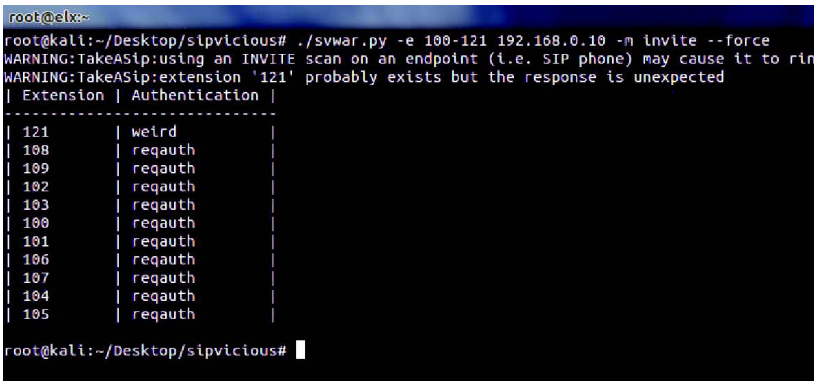


Figura 3.14 Ataque de escaneo a extensiones en la central IP-PBX sin aplicar mecanismos de seguridad

Con esta herramienta se debe realizar algunos intentos hasta encontrar un grupo de extensiones disponibles ya que la prueba se realiza ingresando un rango de extensiones como parámetro para la búsqueda. Ya que se ha obtenido las extensiones configuradas en la IP-PBX y debido a que se ha descubierto que una no tiene una contraseña asociada, entonces se utiliza “svcrack” para atacar a esta extensión.

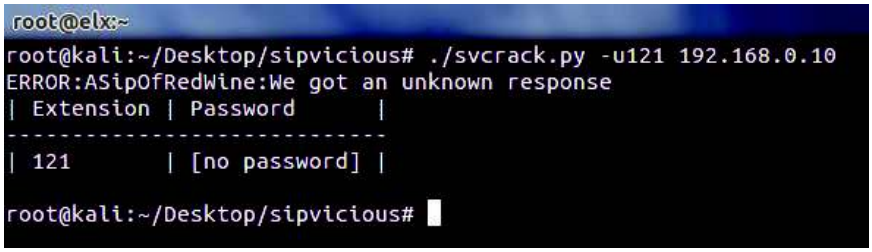


Figura 3.15 Ataque de *password cracking* al protocolo SIP con svcrack sin aplicar los mecanismos de seguridad

En la Figura 3.15 se observa el resultado de realizar este ataque. Como se puede ver “svcrack” devuelve como respuesta del ataque la extensión y la confirmación de que no se tiene una contraseña asociada. Esta es una vulnerabilidad que puede ser explotada por los atacantes ya que al descubrir esta extensión fácilmente se pueden registrar dentro de la central y empezar a realizar llamadas y perjudicar económicamente al dueño de la central IP-PBX.

Para atacar a las extensiones que si requieren de una contraseña nuevamente se hace uso de la herramienta “svcrack” y se utiliza una vez más un diccionario de contraseñas comunes para encontrar una que coincida con la que está asociada a la extensión que se está atacando. En la Figura 3.16 se observa que para la extensión 109, una de las contraseñas en el diccionario coincide y se devuelve en el resultado de “svcrack”. Una vez que se tiene esta coincidencia el atacante ya puede registrarse en la central y empezar a realizar llamadas fraudulentas.

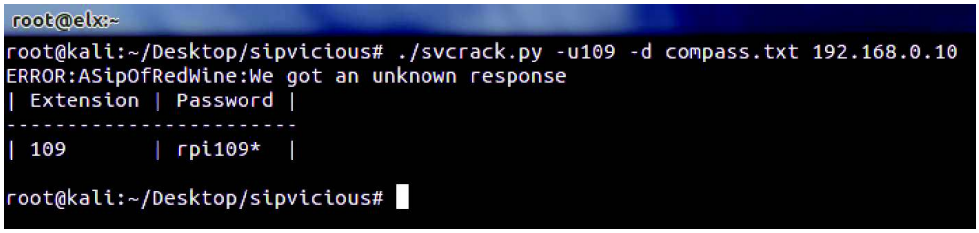


Figura 3.16 Ataque de password cracking a la extensión 109

Cabe recalcar que para esta última prueba se ha añadido al diccionario de contraseñas aquella que corresponde con la extensión que se está atacando de manera que haya una coincidencia. Esto es para demostrar como sería el ataque en un ambiente real ya que el atacante no conoce la contraseña y pretende encontrar una coincidencia con el uso del diccionario realizando tantos intentos como le sea posible. De ahí la gran importancia de utilizar contraseñas fuertes compuestas de varios caracteres entre mayúsculas, minúsculas, números y símbolos, que no sean comunes ni fáciles de romper.

3.3.2 PRUEBAS DE ATAQUE A LA CENTRAL IP-PBX CON LOS MECANISMOS DE SEGURIDAD HABILITADOS

Una vez que se han realizado las pruebas de seguridad para evaluar cuales son las vulnerabilidades a las que puede estar expuesta la central IP-PBX, se activan los servicios de seguridad que se implementaron durante la configuración de la

central telefónica IP para ver de qué manera mitigan estas vulnerabilidades de seguridad presentes en el sistema telefónico IP.

Como primer paso se activa el Firewall IP TABLES que viene dentro de *uElastix* desde la interfaz web y luego se pone en marcha el servicio Fail2ban digitando el comando >> *service fail2ban start*.

En la Figura 3.17 se observa dentro del cuadro azul, que las reglas de IP-TABLES ahora se encuentran activas para los protocolos SSH y el servicio Asterisk.

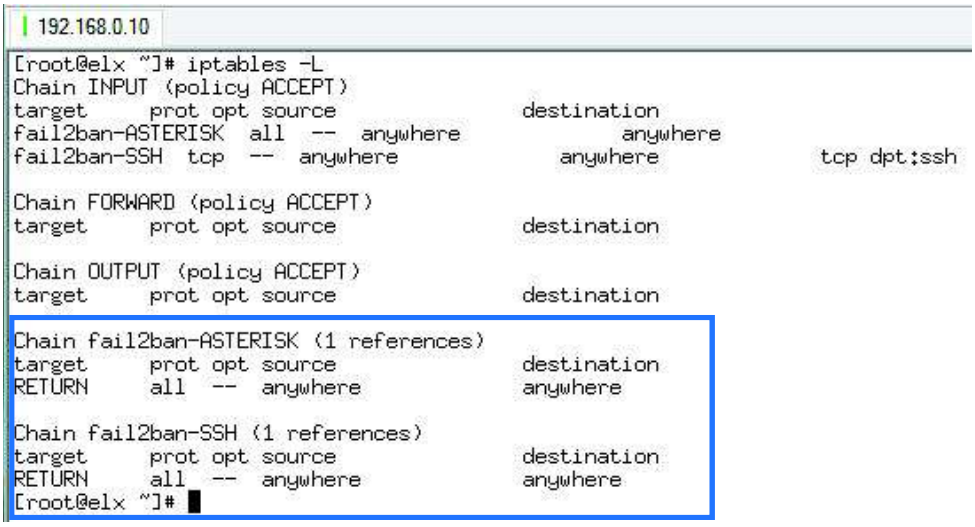


Figura 3.17 IP TABLES con las reglas de Fail2Ban implementadas

En la Figura 3.18 se observa que al realizar un escaneo de puertos en la central IP-PBX utilizando “nmap”, ya no aparecen los puertos 22, 80, y 433 correspondientes a los servicios SSH, HTTP y HTTPS respectivamente. Esto debido a que según las políticas implementadas en el Fairewall IP-TABLES descrito en la sección 2.12, el acceso a estos puertos solamente se encuentra habilitado para la dirección IP del administrador (192.168.0.101) del servidor IP-PBX.

Ya que los puertos 22, 80 y 443 se encuentran escondidos frente a escaneo o descubrimiento de servicios se previene los ataques de fuerza bruta o intento de conexiones remotas no autorizadas a través de estos ya que los atacantes no saben si estos servicios están habilitados o si los puertos han sido cambiados por algunos diferentes a los establecidos por defecto.

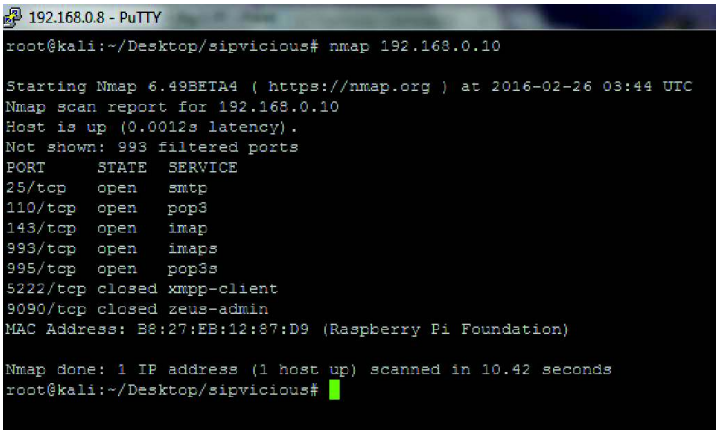


Figura 3.18 Escaneo de puertos abiertos luego de aplicar los mecanismos de seguridad

Si se repite el ataque al protocolo SSH con la herramienta “Hydra” se observa en la Figura 3.19 que aun cuando la contraseña esta listada en el diccionario no es posible encontrar la correspondiente al usuario “root” y se muestra un mensaje de error avisando que no fue posible conectarse mediante SSH como el mostrado en el cuadro amarillo en la Figura 3.19

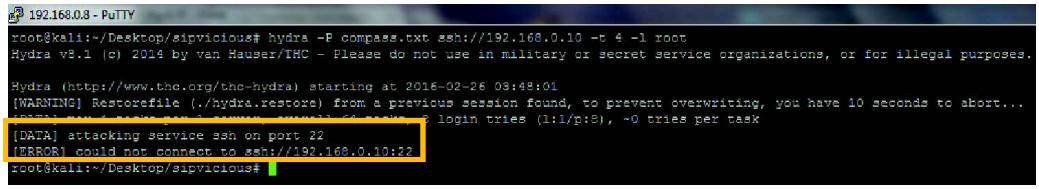


Figura 3.19 Ataque de password cracking al protocolo SSH bloqueado por Fail2Ban

Esto se debe a dos causas. Debido al tráfico restringido al protocolo SSH o al puerto 22 por parte del Firewall IP-TABLES las peticiones de conexión desde cualquier dirección IP diferente a la del administrador será denegado, y adicionalmente la herramienta fail2ban bloquea los ataques de “*password cracking”* dando de baja cualquier dirección IP que intente conectarse a la IP-PBX mediante SSH cuando se ha ingresado la contraseña incorrectamente más de 3 ocasiones.

Si ahora se realiza un ataque contra el servicio de telefonía Asterisk (protocolo SIP) y se escanea de las extensiones configuradas, en la Figura 3.20 se observa que aún es posible obtener el rango de extensiones configuradas.

Pero cuando se intenta hacer un ataque de Pasword Cracking ya no es posible encontrar una contraseña que coincida con la extensión atacada aun cuando esta

consta en el diccionario de contraseñas comunes. En la Figura 3.21 se observa que el ataque con svcrack devuelve una respuesta desconocida y queda sin efecto.

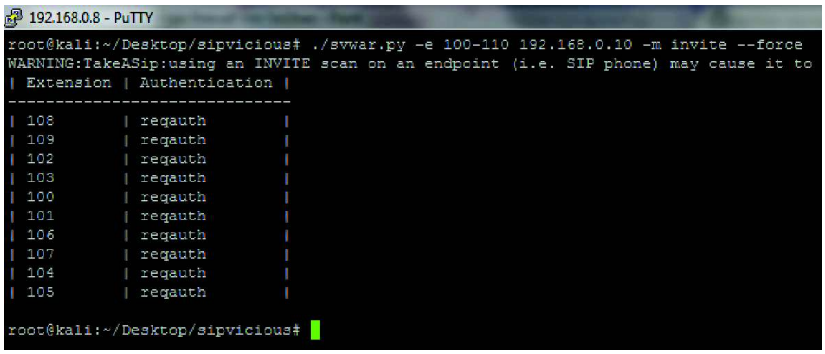


Figura 3.20 Escaneo de extensiones en la IP-PBX con mecanismos de seguridad habilitados

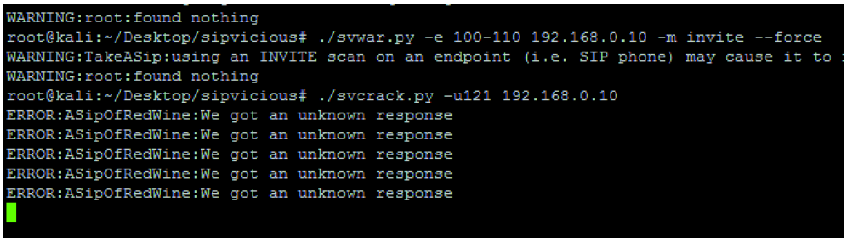


Figura 3.21 Ataque de password cracking a una extensión, bloqueado por Fail2Ban

El servicio Fail2ban ha bloqueado la IP del atacante al haber realizado más de 3 intentos fallidos de adivinar la contraseña tanto para el servicio de telefonía como para los intentos de conexión a través de SSH.

En la Figura 3.22 se observa como han sido bloqueadas las direcciones IP del atacante.

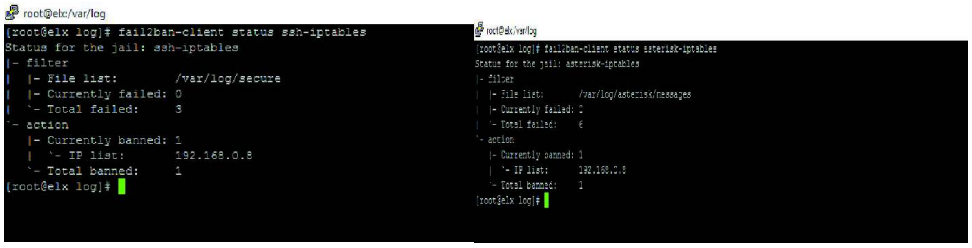


Figura 3.22 Reporte de dirección IP bloqueada por Fail2Ban

Mediante la realización de todas las pruebas de penetración se han descubierto algunas de las vulnerabilidades a las que está expuesta la central IP-PBX. La

implementación de algunos mecanismos de seguridad como la activación del Firewall IP-TABLES que viene incluido en uElastix y la implementación de la herramienta Fail2ban ha ayudado a mitigar algunas de estas vulnerabilidades. A continuación se realiza un resumen de las vulnerabilidades encontradas y las acciones preventivas o correctivas llevadas a cabo.

1. ***Servicios de seguridad inicialmente dados de baja***

   Mediante una revisión del servicio de IP-TABLES se observa que inicialmente el Firewall no viene activado en la instalación del servidor IP-PBX, por lo que es necesario configurar el servicio y darlo de alta antes de que el servidor entre en producción. Sin esta configuración mínima de seguridad el servidor es blanco fácil de ataques principalmente cuando está expuesto a Internet.

2. ***Descubrimiento de servicios.***

   Mediante un escaneo de la red utilizando “svmap” fue posible descubrir la existencia de una central IP-PBX. Ésta es una vulnerabilidad muy común en implementaciones de VoIP y conlleva a que el servicio de telefonía sea víctima de muchos ataques. Para evitar este problema es necesaria la utilización de un equipo firewall propiamente dicho para denegar estas intrusiones [60].

3. ***Puertos de comunicación sensibles abiertos.***

   Utilizando “nmap” y la dirección IP del servidor de telefonía IP se encontraron todos los puertos disponibles entre ellos el 22 (SSH), 80 (HTTP) y 443 (HTTPS), los cuales son los más apetecidos por los atacantes para intentar penetrar al servidor. Mediante la activación del firewall IP-TABLES y la implementación de Fail2Ban estos puertos han quedado ocultos y en el caso de SSH el protocolo ha quedado protegido contra ataques de fuerza bruta como “*password cracking*”.

4. ***Pasword cracking al protocolo SIP.***

   Mediante un escaneo hacia la central IP-PBX utilizando “svwar” fue posible encontrar las extensiones configuradas en el servidor y descubrir cuales requieren de una contraseña para el registro y cuales no tienen una

contraseña asociada. Con esta información es posible realizar ataques de fuerza bruta o "*password cracking"* de manera que se puedan romper estas contraseñas y utilizar esa información para que un atacante se registre en la central telefónica IP y pueda realizar llamadas fraudulentas. Tal como en el caso del servicio SSH mediante la implementación de Fail2ban el protocolo SIP queda protegido contra este tipo de ataques.

CAPÍTULO 4

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Luego del desarrollo de este proyecto de titulación se puede concluir lo siguiente:

- El uso de estándares definidos por instituciones como la IETF, brindan todas las pautas, recomendaciones y definiciones necesarias para las implementaciones de aplicaciones como VoIP. Esto hace que las plataformas de software libre que utilizan estos estándares sean más versátiles a la hora de desplegar las implementaciones de VoIP y las hace compatibles con otros sistemas aun cuando sean desarrolladas por otras personas, haciendo de las comunicaciones un mundo más accesible para todos.
- Las plataformas de hardware libre como la Raspberry Pi permiten el desarrollo de muchas aplicaciones en varios ámbitos como la domótica, control, redes de sensores inalámbricos, y como se ha visto en este proyecto de titulación también se puede utilizar en VoIP. En todas estas aplicaciones el desarrollador es quien decide la forma de cómo utilizar el hardware.
- Este proyecto presenta la implementación de un prototipo de central telefónica IP que utiliza una Raspberry Pi como plataforma de hardware y *uElastix* como el software para la IP-PBX. En base a esto se realizaron mediciones del rendimiento del sistema de telefonía IP implementado en la Raspberry Pi, obteniendo como resultado que la plataforma realiza un muy buen trabajo en el procesamiento de las llamadas al ver que el porcentaje de uso del CPU no supera el 80 % cuando existen 18 llamadas concurrentes.
- Al evaluar parámetros más específicos como el tiempo de respuesta y las llamadas fallidas se observa el comportamiento real del sistema telefónico. Los tiempos de respuesta se elevan significativamente llegando a los 500 ms de retardo cuando se tienen 16 llamadas simultáneas. Esto genera un amplio impacto en cuanto a la calidad de transmisión de voz que ofrece el sistema ya que ésta es una aplicación en tiempo real muy susceptible a los retardos. En cuanto a los codecs de audio, G.729 es el códec que mejor

funciona en términos del tiempo de respuesta, pero GSM es el códec que mejor trabaja cuando se evalúa las llamadas fallidas.

- La prueba subjetiva MOS muestra resultados satisfactorios para el códec GSM cuando se trabaja con 5 llamadas simultáneas dentro de la infraestructura inalámbrica que se ha propuesto en este proyecto de titulación, obteniendo una calificación promedio de 3.9 sobre 5 lo cual se establece como buena para este tipo de sistemas.
- Debido a su buena calidad de voz durante la conversación, baja tasa de tiempos de respuesta y menor número de llamadas fallidas se ha escogido a GSM como el códec a utilizar en la central telefónica IP.
- EL software que se utiliza para la implementación de la IP-PBX brinda grandes prestaciones y permite tener muchas de las funcionalidades de una PBX, como el establecimiento y recepción de llamadas desde y hacia la PSTN, llamadas entre extensiones, captura y transferencia de llamadas, llamada en espera, IVR, entre otros servicios que se describen en este trabajo. Así el prototipo presentado muestra versatilidad y puede ser una muy buena alternativa como sistema de comunicaciones para pequeñas empresas o poblaciones rurales con baja densidad poblacional, siendo además un sistema de bajo costo.
- La infraestructura de telecomunicaciones para este proyecto utiliza una interfaz inalámbrica WLAN, a través de la cual los clientes SIP se comunican con el servidor IP-PBX. Al realizar las pruebas de rendimiento, servicios implementados y la prueba MOS se observa que la infraestructura inalámbrica no afecta a la calidad de voz del sistema (mientras se encuentre dentro del área de cobertura del router inalámbrico), permitiendo así que todas las funcionalidades implementadas funcionen correctamente.
- El prototipo de central IP-PBX implementado en el presente proyecto ha pasado con éxito todas las pruebas de servicio que se le han realizado. Permite realizar llamadas a la PSTN, recibir llamadas desde la PSTN hacia cualquier extensión de usuario configurada. El prototipo tiene las capacidades de una PBX para transferir llamadas a otras extensiones, capturar llamadas, intercomunicación entre varias extensiones, permite la

configuración de distintas rutas para llamadas nacionales, números móviles. Adicionalmente cuenta con un IVR que realiza las funciones de contestadora automática para llamadas que ingresen a la IP-PBX y luego redirigirlas a la extensión correspondiente.

- Adicionalmente el software utilizado para este proyecto permite tener funcionalidades adicionales a los configurados. Estas funcionalidades se habilitan marcando unos códigos de funcionalidades con los que cuenta el software. Así para el desvío incondicional de llamadas a otro número de extensión se puede marcar el código (*72) y marcar la extensión a la que se quiere desviar las llamadas entrantes.
- El cliente SIP utilizado en ente proyecto es Zoiper. Este cliente permitió probar todas las funcionalidades del sistema IP-PBX, es compatible con cualquier teléfono inteligente en el que fue probado e inclusive es compatible con los sistemas operativos Windows y Linux donde también fue probado, permitiendo así la comunicación entre distintas plataformas haciendo que el sistema sea más versátil todavía.
- Los sistemas VoIP son vulnerables a una gran cantidad de ataques, por lo que establecer políticas de seguridad es muy necesario en este tipo de implementaciones. A lo largo de las pruebas de seguridad que se realizaron se detectaron algunas de las vulnerabilidades que tenía el servidor IP-PBX y luego de aplicar las políticas de seguridad necesarias se ha logrado mitigar muchos de los ataques a los que podía estar expuesto el servidor IP-PBX. El ámbito de la seguridad en implementaciones de VoIP es un campo que presenta muchas oportunidades de estudio y desarrollo.

4.2 RECOMENDACIONES

- Luego de la evaluación del rendimiento del sistema se recomienda que este prototipo se utilice para un máximo de 10 usuarios y con 5 llamadas concurrentes como tope, ya que entre estos límites se saca el mejor provecho de los recursos del hardware, es decir de la Raspberry Pi, y todos los servicios que se han configurado en la central IP-PBX pueden ser utilizados.

- Debido a que el sistema utiliza una interfaz inalámbrica WLAN para la conexión de los usuarios con el servidor; se recomienda que para lugares con un área de cobertura muy amplia se utilice un sistema WDS (Wireless Distribution System) para ampliar la cobertura de la red inalámbrica y que todos los usuarios puedan acceder al servicio de telefonía IP. Esta solución depende también de las necesidades del cliente y del costo de los equipos que se puedan utilizar.
- En el ámbito de la seguridad del sistema IP-PBX, aun cuando los mecanismos de seguridad implementados en este trabajo han cumplido su objetivo de proteger al sistema frente a varios tipos de ataques, se debe pensar en mecanismos de seguridad para cada escenario. Herramientas como TLS (Transport Layer Security) y SRTP (Secure Real-Time Transport Protocol) que son protocolos estándares y que se soportan en Elastix brindan confidencialidad en las comunicaciones que ayudan a minimizar el riesgo de ataques.

Bibliografía

[1] E. LANDIVAR, Comunicaciones Unificadas con Elastix, 2012.

[2] J. CIAORA y M. VALENTINE, CCNA VOICE 640-461 Official Cert Guide, Cisco Pess, 2012.

[3] S. JIMÉNEZ, Comunicación Digital, Quito, 2012.

[4] A. TANENBAUM, REDES DE COMPUTADORAS, Pearson Education, 2012.

[5] M. ESCOBAR, Telefonía y Conmutación, Red Tercer Milenio, 2012.

[6] ASTERISK, «What is a VoIP Gateway?,» 2015. [En línea]. Available: http://www.asterisk.org/get-started/applications/gateway. [Último acceso: 21 Noviembre 2015].

[7] 3CX, «Información sobre VoIP gateway,» 2015. [En línea]. Available: http://www.3cx.es/voip-sip/pasarela-voip/. [Último acceso: 21 Noviembre 2015].

[8] R. FREEMAN, Telecomunication System Engineering, Hoboken, New Jersey: John Wiley & Sons, Inc, 2004.

[9] ELASTIX TECH, «Interconexion a la PSTN,» 2014. [En línea]. Available: http://elastixtech.com/fundamentos-de-telefonia/interconexion-a-la-pstn/. [Último acceso: 16 Diciembre 2015].

[10] ITU-T, H.323 Packet-based multimedia communications systems, 2009.

[11] ELASTIX, «Protocolo IAX,» 2014. [En línea]. Available: http://elastixtech.com/protocolo-iax/. [Último acceso: 21 Junio 2015].

[12] IETF, «RTP: A Transport Protocol for Real-Time Applications,» 2003. [En línea]. Available: https://tools.ietf.org/html/rfc3550. [Último acceso: 5 Marzo 2015].

[13] VoIP Foro, «Funcionamiento de un Codec G.711,» 2015. [En línea]. Available: http://www.voipforo.com/codec/codec-g711--ley.php. [Último acceso: 15 Mayo 2015].

[14] VoIP-Info, «ITU G.711,» 2015. [En línea]. Available: http://www.voip-info.org/wiki/view/ITU+G.711. [Último acceso: 15 Mayo 2015].

[15] ITU-T, UIT-T G.711. MODULACION POR IMPULSOS CODIFICADOS (MIC) DE FRECUENCIAS VOCALES, 1993.

[16] ETSI, «Digital cellular telecommunications system (Phase 2+) (GSM); Full rate speech; Transcoding,» 1999.

[17] VoIP-Info, «GSM Codec,» 2015. [En línea]. Available: http://www.voip-info.org/wiki/view/GSM+Codec. [Último acceso: 18 Mayo 2015].

[18] IETF, «RFC 3951: Internet Low Bit Rate Codec (iLBC),» 2004. [En línea]. Available: http://www.ietf.org/rfc/rfc3951.txt. [Último acceso: 22 05 2015].

[19] UIT-T, G.722: 7 kHz audio-coding within 64 kbit/s, 2012.

[20] Open Source Hardware Association , «Definición de Hardware Libre,» 2014. [En línea]. Available: http://www.oshwa.org/definition/spanish/. [Último acceso: 19 Enero 2014].

[21] ELASTIX, «uElastix, a big solution on micro devices,» 2012. [En línea]. Available: http://www.uelastix.com. [Último acceso: 20 Septiembre 2015].

[22] ASIRI, «Acerca de Asiri,» 2015. [En línea]. Available: http://asiri.ec/index.php/es/acerca-de-asiri.html. [Último acceso: 15 Enero 2015].

[23] P. ESTRELLA, «Hardware abierto desde Latinoamérica,» 2013. [En línea]. Available: http://es.slideshare.net/elastixorg/asiri-minga13ss. [Último acceso: 15 Enero 2015].

[24] Raspberry Pi Foundation, «What is a Raspberry Pi?,» 2012. [En línea]. Available: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/. [Último acceso: 15 Enero 2015].

[25] Elinux, «Raspberry Pi Hub,» 2015. [En línea]. Available: http://elinux.org/RPi_Hub. [Último acceso: 15 Enero 2015].

[26] MINI BOX, «pico-SAM9G45,» 2015. [En línea]. Available: http://www.mini-box.com/pico-SAM9G45-X. [Último acceso: 15 Enero 2015].

[27] MINI BOX, «pico-SAM9G45 highly extensible low cost, low power, pico-ITX compatible board designed and manufactured by mini-box,» 2015. [En línea]. Available: http://arm.mini-box.com/index.php?title=Main_Page#Powering_up_the_pico-SAM9G45_board. [Último acceso: 15 Enero 2015].

[28] OPENDIREITO, «Un procedimiento de instalación de la ISO generada para plataformas basadas en ARM.,» 2013. [En línea]. Available: http://opendireito.com/2013/07/13/probando-uelastix-1/. [Último acceso: 19 Agosto 2015].

[29] FEDORA, «Fedora Documentation,» [En línea]. Available: https://docs.fedoraproject.org/en-US/Fedora/18/html/Installation_Quick_Start_Guide/Requirements.html. [Último acceso: 10 Septiembre 2015].

[30] A. CERTAIN, Trixbox al descubierto, Gecko EU, 2006.

[31] Fonality, *Trixbox reference guide,* 2006.

[32] F. ERLEWEIN, «Architecture Paper,» 2004.

[33] KOLAB Community, «Kolab Overview,» 2015. [En línea]. Available: http://kolab.org/overview. [Último acceso: 08 Marzo 2015].

[34] FREESWITCH, «THE WORLD'S FIRST CROSS-PLATFORM SCALABLE FREE MULTI-PROTOCOL SOFT SWITCH,» 2014. [En línea]. Available: https://freeswitch.org. [Último acceso: 6 Septiembre 2015].

[35] FREESWITCH, «Raspberry Pi,» 2015. [En línea]. Available: https://freeswitch.org/confluence/display/FREESWITCH/Raspberry+Pi. [Último acceso: 6 Septiembre 2015].

[36] FREESWITCH, «Raspberry PI-specific documentation,» 2014. [En línea]. Available: https://wiki.freeswitch.org/wiki/Raspberry_PI-specific_documentation. [Último acceso: 06 Septiembre 2015].

[37] DEBIAN, «Meeting Minimum Hardware Requirements,» 2015. [En línea]. Available: https://www.debian.org/releases/stable/amd64/ch03s04.html.en. [Último acceso: 16 Septiembre 2015].

[38] FONALITY, «Simple Pricing,» 2015. [En línea]. Available: http://www.fonality.com/pricing. [Último acceso: 7 Julio 2016].

[39] SD Association, «Speed Class,» 2014. [En línea]. Available: https://www.sdcard.org/developers/overview/speed_class/index.html. [Último acceso: 11 Noviembre 2015].

[40] PANASONIC, «Speed Class and Performance,» 2014. [En línea]. Available: http://panasonic.net/avc/sdcard/industrial_sd/performance.html. [Último acceso: 11 Noviembre 2015].

[41] SD Association, «Bus Speed (Default Speed/ High Speed/ UHS),» 2014. [En línea]. Available:

https://www.sdcard.org/developers/overview/bus_speed/index.html. [Último acceso: 11 Noviembre 2015].

[42] Cambridge University, «mkdosfs - create an MS-DOS file system under Linux,» 1995. [En línea]. Available: https://www.cl.cam.ac.uk/cgi-bin/manpage?8+mkfs.vfat. [Último acceso: 9 Marzo 2016].

[43] Zoiper, «Zoiper IAX SIP VOIP Softphone,» 2015. [En línea]. Available: https://play.google.com/store/apps/details?id=com.zoiper.android.app. [Último acceso: 10 Marzo 2015].

[44] ITU-T, «VARIOUS TONES USED IN NATIONAL NATWORKS (ACORDING TO ITU-T RECOMMENDATION E.180),» ITU-T, Génova, 2003.

[45] ELASTIXTECH, «Troncales SIP para PBX-IP-Elastix,» 2013. [En línea]. Available: http://elastixtech.com/troncales-sip-para-pbx-ip-elastix/. [Último acceso: 10 Octubre 2015].

[46] J. ROSENBERG, «What is a Session Initiation Protocol (SIP) Trunk Anyway?,» 2008. [En línea]. Available: https://tools.ietf.org/id/draft-rosenberg-sipping-siptrunk-00.txt. [Último acceso: 12 Junio 2016].

[47] FreePBX, «Trunk Sample Configurations,» 2014. [En línea]. Available: http://wiki.freepbx.org/display/FPG/Trunk+Sample+Configurations. [Último acceso: 2 Marzo 2015].

[48] www.fail2ban.org, «www.fail2ban.org,» 2013. [En línea]. Available: http://www.fail2ban.org/wiki/index.php/Main_Page. [Último acceso: 5 Septiembre 2014].

[49] ElastixTech, «Seguridad Básica en Elastix,» 2013. [En línea]. Available: http://elastixtech.com/seguridad-basica-en-elastix/. [Último acceso: 10 Octubre 2015].

[50] VOZTOVOICE.ORG, «Instalar y configurar fail2ban para Asterisk,» [En línea]. Available: https://www.voztovoice.org/?q=node/135. [Último acceso: 2015 08 20].

[51] ELASTIX, «Fail2Ban installation,» [En línea]. [Último acceso: 2015 Agosto 20].

[52] Source Forge, «SIPp Documentation,» 2015. [En línea]. Available: http://sipp.sourceforge.net/doc/reference.html. [Último acceso: 15 Junio 2015].

[53] UIT-T, Terminología de las notas medias de Opinión, Unión Internacional de Telecomunicaciones, 2006.

[54] ITU-T, Recommendation ITU-T P.800.2 - Mean opinion score interpretation and reporting, 2013.

[55] ITU-T, Subjective evaluation of conversational quality, 2007.

[56] SUPERINTENDENCIA DE COMPAÑÍAS, «Clasificación de las PYMES,» 2010. [En línea]. Available: http://www.russellbedford.com.ec/images/Boletines%202010/12.%20Resolucion%20SUPER%20CIAS%20PYMES%20-%20SC-INPA-UA-G-10-005.pdf. [Último acceso: 02 Junio 2015].

[57] A. PAZ, «Tamaño de una muestra para una investigación de mercado,» 2013. [En línea]. Available: http://www.tec.url.edu.gt/boletin/URL_02_BAS02.pdf. [Último acceso: 02 06 2015].

[58] O. SALCEDO, «A comparative study of bandwith usage running SIP and IAX.,» *Tecnura,* pp. 171 - 187, 2012.

[59] Offensive Security, «Kali Linux on a Raspberry Pi (A/B+/2) with Disk Encryption,» [En línea]. Available: https://www.offensive-security.com/kali-linux/raspberry-pi-luks-disk-encryption/. [Último acceso: 06 Marzo 2016].

[60] J. OLIVA, «Seguridad en Implementaciones de voz sobre IP,» ELASTIX, 2014.

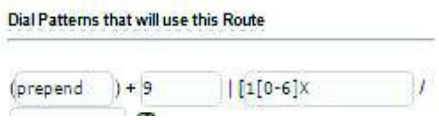
[61] O. GROMETA, «Mis libros de Natworking,» 2009. [En línea]. Available: http://librosnetworking.blogspot.com/2009/04/fxs-fxo.html. [Último acceso: 02 Febrero 2016].

ANEXOS

ANEXO A: CONFIGURACIONES ADICIONALES

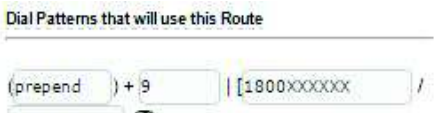
A.1 Configuración de patrones de marcado para rutas de salida

- **Servicios:**



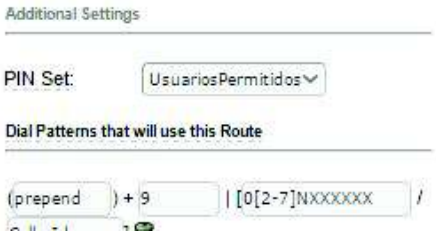
A.1-1 Patrón de marcado para ruta de salida Servicios

- **1800:**



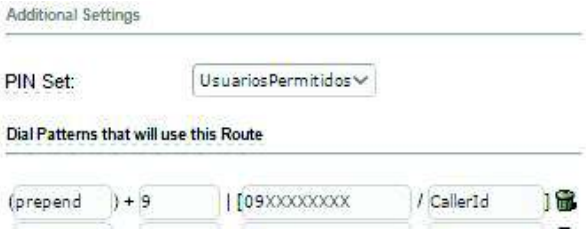
A.1-2 Patrón de marcado para ruta de salida 1800

- **Nacional:**



A.1-3 Patrón de marcado para ruta de salida Nacional

- **Móvil:**



A.1-4 Patrón de marcado para ruta de salida Móvil

A.2 Configuración de perfil para prueba con SIPp

En el capítulo 3 en la sección 3.2 se muestran las pruebas de rendimiento del sistema IP-PBX. Para la realización de estas pruebas se menciona que se configuró una extensión de prueba con un flujo de llamada simple que se sigue para contestar y procesar las llamadas que provienen de la herramienta SIPp.

En el archivo sip_additional.conf que se encuentra en el directorio: /etc/Asterisk/ se adicionan las siguientes líneas para crear una extensión dentro del contexto [sipp] que tiene el flujo que va a seguir la llamada.

```
[sipp]
type=friend
context=sipp
host=dynamic
port=6000
user=sipp
canreinvite=no
disallow=all
;allow=g729
;allow=ilbc
allow=gsm
;allow=g722
;allow=alaw
;allow=ulaw
```

En el archivo extensions_custom.conf se configura el contexto que contiene el flujo que siguen las llamadas provenientes de SIPp. Esta configuración se realiza agregando las siguientes líneas en este archivo.

```
[sipp]
exten => 1001,1,Answer
;exten => 1001,n,WaitExten
exten => 1001,n,SetMusicOnHold(default)
exten => 1001,n,WaitMusicOnHold(20)
exten => 1001,n,Hangup
;end of [sipp]
```

Estos archivos se anexan en formato digital en el CD de anexos.

ANEXO B: UTILIZACIÓN DE SIPp

B.1 Opciones de la línea de comandos de SIPp

En esta sección se anexan los comandos utilizados con la herramienta SIPp para generar las llamadas que saturan al sistema, los cuales se muestran en la Tabla B.1.1.

```
david@david-Ubuntu:~$ sudo ./sipp -sf uac_pcap_ilbc.xml -s 1001
192.168.0.110 -l 18 -r 2 -m 200 -trace_stat -fd 2
```

B.1-1 Opciones del comando sipp utilizado para la prueba de rendimiento

PARÁMETRO	VALOR	SIGNIFICADO
./sipp		Ejecutar prueba de sipp
-sf	uac_pcap_G729.xml	Carga un escenario xml para la prueba
	uac_pcap_G722.xml	
	uac_pcap_GSM.xml	
	uac_pcap_ilbc.xml	
-sn	uac_pcap	Carga un escenario precargado para la prueba
-s	1001 / 192.168.0.110	Extensión de destino para las llamadas de prueba
		IP del servidor Elastix
-l	2 – 18	Número máximo de llamadas simultáneas que se ejecutan en la prueba
-r	2	Número de llamadas que se incrementan por segunda hasta llegar al máximo de llamadas
-m	200	Número máximo de llamadas que se ejecutan durante la prueba
-trace_stat		Guarda un reporte de las estadísticas de la prueba en un archivo .csv
-fd	2	Incremento de la llamadas simultáneas cada 2 segundos

ANEXO C: DATOS DE RENDIMIENTO DEL SISTEMA

C.1 Resumen de datos de las pruebas de rendimiento del sistema

C.1-1 Porcentaje de uso de CPU y memoria RAM por códec en función del número de llamadas simultáneas

Llamadas simultáneas	G.711		GSM		G.722		G.729		iLBC	
	% CPU	% RAM	% CPU	% RAM	% CPU	% RAM	% CPU	% RAM	% CPU	% RAM
2	12,3	7,1	10,8	6,3	15,7	6,3	35,8	6,4	84	6,4
4	23,5	7,1	16,6	6,5	24,9	6,5	49,7	6,6	37,9	6,6
6	32,3	7,1	24,9	6,7	36,8	6,7	56,3	6,8	43,1	6,8
8	43,2	7,2	33,2	6,9	46,8	6,9	60,9	7	55,3	7
10	56,8	7,4	42,6	7,1	61,2	7,1	65,9	7,2	61,1	7,4
12	63,7	7,7	49,2	7,2	65,5	7,3	73,2	7,4	66,8	7,5
14	69,9	7,8	53,7	7,4	67,3	7,5	74,9	7,6	69,1	7,7
16	72,3	7,9	56,7	7,6	67,1	7,4	71,7	7,8	74,6	7,4
18	67,6	7,8	61,3	7,7	67,9	7,5	68,8	7,8	81,2	7,6

C.1-2 Llamadas fallidas por códec en función de las llamadas simultáneas

Llamadas simultáneas	G.711	GSM	G.722	G.729	iLBC
2	0	0	0	0	2
4	0	0	0	0	0
6	0	0	0	0	0
8	0	0	0	0	0
10	0	0	0	0	137
12	0	0	0	0	137
14	6	0	146	0	102
16	129	0	156	133	183
18	152	131	162	148	177

C.1-3 Tiempo de respuesta [ms] en función del número de llamadas simultáneas

Llamadas simultáneas	G.711	GSM	G.722	G.729	iLBC
2	87,32	137,25	159,38	199,25	238,95
4	119,39	98,22	268,36	146,3	172,5
6	188,68	128,42	334,8	126,97	194,97
8	258,58	212,4	343,86	119,48	395,05
10	280,95	367,22	216,79	104,5	457,8
12	274,89	249,81	257,89	138,14	439,12
14	189,45	329,81	468,87	166,73	481,82
16	430,51	239,17	448,5	435,47	487,57
18	467,99	437,66	467,94	422,25	495,76

ANEXO D: RECOMENDACIONES PARA REALIZAR PRUEBA MOS (MEAN OPINION SCORE)

D.1 Respuestas de prueba MOS

Remitirse al archivo: Test MOS_Responses.xlsx

D.2 Ejemplos de escenarios de conversación para la prueba MOS

Remitirse al apéndice VIII del archivo: T-REC-P.805-200704-I!!PDF-E.pdf

D.3 Ejemplos de preguntas para la prueba MOS

Remitirse al Anexo B del archivo: T-REC-P.82-198811-W!!PDF-S.pdf

D.4 Tipos de prueba MOS

Remitirse a la sección 6 del archivo: T-REC-P.800.2-201305-S!!PDF-E.pdf

ANEXO E: DIAGRAMA ESQUEMÁTICO DEL HARDWARE DE LA RASPBERRY PI

Se adjunta en formato digital el diagrama esquemático de la Raspberry Pi. Referirse al archivo Raspberry-Pi-Rev-2.0-Model-AB-Schematics.pdf