Decanato de Ingeniería e Informática
Escuela de Ingeniería

Trabajo de Grado Optar por el Título de:
Ingeniero Electrónico, Mención Comunicaciones.

Propuesta de aplicación de un sistema de voz sobre IP Utilizando Raspberrry pi, Asterisk, Open-VPN y DNS dinámico en campus II, UNAPEC

Sustentantes:

Joan Eliet Heiliger Ruiz	2013-0771
Henry Francisco Hiraldo Santos	2013-0554

Asesor:

Carlos Guzmán Hernández

Distrito Nacional
República Dominicana
Agosto 2018

RESUMEN

Hoy en día tecnología IP esta involucradas en gran parte de la vida cotidiana de las personas. Este trabajo de grado se centra en el estudio de la tecnología VoIP. Esta permite comunicación de voz en tiempo real utilizando como medio una red IP. El objetivo de este trabajo de grado es diseñar un sistema VoIP que permita realizar llamadas a números de líneas fijas y móviles. Con el fin de reducir gastos en llamadas a largas distancia o servicio de *roamig* o itinerancia. El diseño de dicho sistema está basado en hardware de bajo costo y softwares de código abierto.

DEDICATORIAS

Henry Francisco Hiraldo Santos

A mis padres, Henry de Jesús Hiraldo y María Victoria Santos.

A mis hermanas, Anabel Hiraldo, Sofia Hiraldo y mis amigos.

A mi asesor Carlos Guzmán.

A todas las personas que me apoyaron.

Joan Eliet Heiliger Ruiz

A mis padres.

A la familia reyes medina.

A mis 32 compañero de la promoción 2008-2012 de electrónica en ITESA, en especial a Carlos cruz.

A todos los profesores que de algún modo u otro han aportado en mi formación profesional

Agradecimientos

Henry Francisco Hiraldo Santos

Agradezco a mis padres Henry de Jesus Hiraldo y Maria Victoria Santos; a mis hermanas Anabel Hiraldo y Sofía Hiraldo por el apoyo de que me han brindado en todo momento y por siempre estar para mí en todo momento. Gracias a mi compañeros y amigos Joan Eliet Heiliger, Stalyn A. Abreu Perez, Manuel Isaías Peña, Rolando S. Mata, Karla Giselle Perez por todo el soporte, apoyo y crecimiento que me han brindado.

Joan Eliet Heiliger Ruiz

A mis padres ya que ellos fuero los propulsores de que pudiera terminar mis estudios.

A nuestro asesor Carlos Guzmán que siempre estuvo dispuesto para aporta sus conocimiento y experiencia para hacer posible este trabajo de grado

A todos mis compañeros y profesores con los que compartir durante el transcurso de mis estudios en UNAPEC.

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

INTRODUCCIÓN

Actualmente vivimos en una época en la que viajes por negocios, vacaciones, estudios e incluso trabajo son bastante común. Parte fundamental de estas estadías fuera de nuestro entorno hogareño, es establecer medios de comunicación de largas distancias con nuestros familiares, compañeros de trabajos y negocios.

Hoy en día existes diversas formas de comunicación a larga distancia, algunas tienen un gran costo como el servicio de roamig o itinerancia; otras son gratis como las redes sociales video, videollamadas pero tienen la desventaja que no permiten a llamadas a líneas fijas o líneas móviles; también estas los servicios de telefonía IP que nos permiten tener una línea telefónica a través de internet con la desventaja de que estos servicios están sujetos a ciertas restricciones como a los lugares donde se puede llamar, el país origen del número a contratar.

En el presente trabajo se centrará en el diseño de un sistema de voz sobre IP (VoIP) utilizando recursos de hardware de bajos costos y software de código libre, que permita realizar llamadas a líneas fijas y móviles a través de internet. Y que sirva para reducir gatos en roaming y supere las desventajas de los servicios de Telefonía IP convencionales.

1 Sistemas VoIP

1.1 Marco histórico

La primera red de pautes fue creada en 1969 por el departamento de defensa de los estados unidos, en ese entonces llamada ARPANET(Advanced Research Projects Agency Network / Red de la Agencia de Proyectos de Investigación Avanzada). Fue utilizarla como medio de comunicación entre las diferentes instituciones académicas y estatales.

La primera forma de transmisión de voz en forma de paquete se diseñó en 1973, por Bob McAuley y Charlie Radar en el Instituto de Tecnología de Massachusetts (MIT). Para 1974 hicieron la primera prueba exitosa entre una institución en california y el MIT sobre la red ARPANET.

Ya en 1974 con la publicación del primer documento RFC sobre el protocolo TCP se usa el termino internet, que significa interconexión de redes. Se utilizo por primera vez para interconectar la red APARNET con NSFNET. Ya luego en los 80 con la suite de protocolos TCP/IP el internet se globalizo.

En 1988 aparecieron los primeros códecs de audio de ancho de ancho de banda. Estos fueron los G722 y G711.

VOIP inicia a finales del siglo XX a mediados de los años 90 más exactamente en 1995 por un grupo de jóvenes israelíes que pretendían codificar la voz para ser transmitida de un ordenador a otro. En el mismo año se realizó el lanzamiento del primer *softphone* por

la empresa Vocaltec, Inc. para ser utilizado en un computador que debía tener tarjeta de sonido, micrófono, parlantes y modem para su funcionamiento. El *softphone* llamado "Internet Phone Software" no tuvo mucho auge porque en esa época aún no había evolucionado la internet en el tema del ancho de banda por consiguiente fue un fracaso para la empresa e igualmente para los usuarios [1].

En marzo de 1997 la compañía MCI de origen estadunidense desarrollo un proyecto sobre VoIP llamado VAULT, este consistía en interconectar y combinar redes tradicionales de telefonía (PSTN) con redes de datos. Ese mismo año Jeff Pulver decide realizar la primer feria llamada VON con usuarios, fabricantes e interesados sobre la tecnología VoIP. En 1998 un grupo de emprendedores comenzó a fabricar los primeros ATA y Gateways que permitían la comunicación de computador a teléfono convencional (PSTN) y también teléfono convencional a teléfono convencional mediante el uso de ATA en cada extremo.

En 1999 la empresa Cisco vende sus primeras plataformas corporativas para VoIP, principalmente utilizaban el protocolo de señalización H323.

En el año 2000 el estudiante Mark Spencer de la Universidad de Auburn crea Asterisk. La primera central telefónica basada en Linux con un código fuente abierto.

En 2002 surge el protocolo de señalización SIP y este comienza a desplazar a H323.

Hoy en día la telefonía IP se ha vuelto una pieza clave para el desarrollo de una nación en la parte de las telecomunicaciones haciendo más fácil y práctico el uso de esta tecnología para los usuarios finales y a su vez converge con otros servicios de valor agregado.

1.2 ¿Qué es VoIP?

"VoIP (Voice over IP, voz sobre IP) es una tecnología que permite comunicación en tiempo real de voz sobre el protocolo IP. La telefónica IP es una comunicación de voz que solo participan protocolos de VoIP sin traducirlo a otros o convertir la comunicación a telefónica análoga" [2].

"VoIP se refiere a una categoría de hardware y software que permite la transmisión de la voz y otras formas de información sobre de red de transmisión de paquetes. Asociado protocolos, dispositivos, aplicaciones que permiten que la voz transmitida tradicionalmente en una red PSTN ser enviada en forma de paquete en una red de datos IP" [3].

La VoIP es técnica de comunicación en la que se digitaliza la voz y luego es transmitida en forma de paquete de datos a través de una red IP en tiempo real. VoIP también permite brindar servicios de telefonía con las tecnologías IP.

La suite de protocolos TCP/IP es la base de la VoIP y la hace compatible con redes corporativas, privadas, públicas, cableadas e incluso con redes inalámbricas. VoIP puede unir organizaciones en diferentes ubicaciones, incluyendo los trabajadores móviles, en una única red de comunicaciones convergentes [4].

1.3 El modelo TCP/IP

El término modelo de red, o arquitectura de red, se refiere a un conjunto de documentos y normas que describen la función de cada parte de una red de telecomunicaciones. Estos documentos pueden definir protocolos, la lógica que los dispositivos deben usar para comunicarse o incluso las características físicas de la red, como conectores, tipo de medio, voltaje y corriente. Todos los documentos en conjunto describen cómo construir una red funcional [5].

Por su sigla en inglés (*transmission control protocol*/*internet protocol*, protocolo de control de trasmisión/protocolo internet), el modelo TCP/IP es un conjunto de protocolos que rigen cómo deben ser las comunicaciones en una red de datos IP y es lo que hace posible el internet hoy en día. El modelo TCP/IP no es financiado o controlado por un gobierno o persona en particular, las políticas del modelo son dirigidas por conjuntos de comité y organizaciones.

Entre las instituciones más importante están:

- **Internet Society (ISOC):** guía el futuro del internet, supervisando lo estándares y políticas públicas, educando y entrenando.
- **Internet Corporation for Assigned Names and Numbers (ICANN):** se encarga del direccionamiento a nivel mundial del internet.
- **Internet Activities Board (IAB):** se encarga de la arquitectura del internet.
- **Internet Engineering Task Force (IETF):** es un grupo de más de 70 comités informales que ayudan al desarrollo de nuevos estándares para el internet

1.3.1 Los documentos RFC

Los documentos RFC (*request for comments*, solicitud de comentario) son la manera formal en que se documenta informaciones técnicas de tecnologías relacionadas con el internet. Estos documentos no tienen un autor específico, sino que son escritos por grupos de personas pertenecientes a la IETF, IRTF, IAB o personas independientes. Estos están publicados en internet sin ninguna restricción y son identificados por un número único consecutivo a la última publicación. Cuando un documento RFC es publicado nunca se modifica; para hacer una actualización a un RFC es necesario escribir uno nuevo que deje desactualizado el anterior [6].

Los documentos RFC pueden ser descargados y creados en el sitio web [7]. Cada EFC tiene asignado un estatus: es informativo cuando contiene simplemente una nota o idea para la comunidad; experimental si está pensado para experimentar o representar el posible

inicio de un futuro estándar. Otros estados posibles son el estándar borrador (para ser inspeccionado), estándar propuesto (propuestas para un estándar), estándar (estándar oficial) o histórico (en desuso).

1.3.2 Las capas del modelo TCP/IP

TCP/IP funciona dividiendo las funciones de la red en capas y define cómo cada una debe interactuar con la otra. Cada capa depende de la capa que está debajo o por encima de ella. Cada capa tiene una función especial; la capa física y enlace están orientadas al funcionamiento del *hardware*, y las capas de red, transporte y aplicación proveen de servicios a los usuarios.

Al proceso del paso de la data a través de las capas se le llama encapsulación. En cada capa del modelo TCP/IP se le agrega una cabecera a la data original, que son datos con información del protocolo usado en la capa, también se le conoce como PDU (*protocol data unit*, unidad de protocolo). La data generada en la capa aplicación simplemente se llama «data», cuando la data pasa a la capa de transporte es llamada «segmento», luego en la capa red es llamada «paquete» y finalmente en la capa de enlace y física la encapsulación toma el nombre de «trama» y es la representación final en bits de que se transmite o se recibe. Para establecer una comunicación entre dos *hosts* en el modelo TCP/IP la data generada en un *host* A debe de pasar desde la capa aplicación hasta la capa física; por otro lado, un *host* B que hace de receptor de la data debe pasar desde la capa física hasta la capa aplicación [7], como se ilustra en la figura 1.

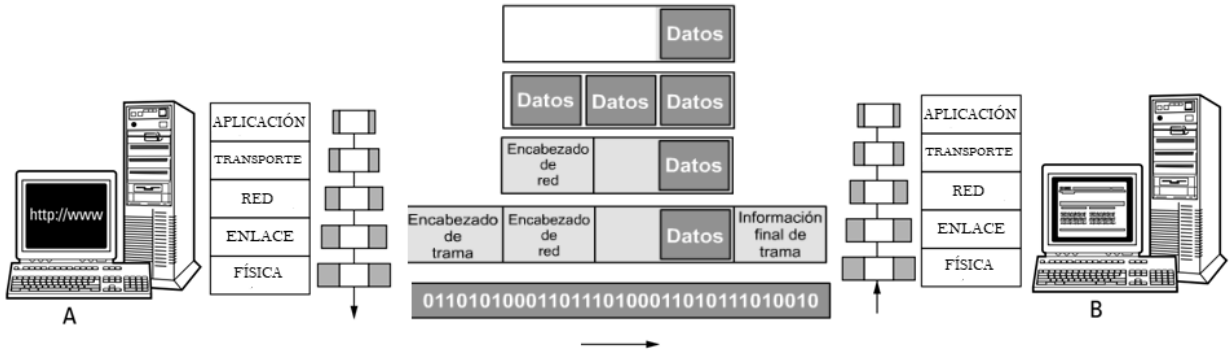


Figura 1: representación gráfica del proceso de encapsulación del modelo TCP/IP **[5] [8]**

La función de las capas del modelo TCP/IP son las siguientes:

La capa física: es la primera capa del modelo TCP/IP; en esta se define el medio de transmisión, que puede ser aire, cable o luz. También define las caracterices de *hardware* en la red como el tipo de interface y características del bus. En esta capa se transforman las señales eléctricas en bits, y viceversa. Los protocolos de la capa física y de enlace no pertenecen al modelo TCP/IP. Los protocolos de esta capa manejan la información en bits y el método de conexión.

El componente principal en esta capa es la NIC (*network interface carda*, interfase de red). Es un circuito que hace la función de transductor, convierte las señales eléctricas en bits, y viceversa. Cada NIC viene con un identificador único fijado por el fabricante llamado MAC Address (media *access control address*, dirección de control de acceso al medio) [8].

Capa de enlace: en esta capa la información en bits se convierte en paquetes de datos, y viceversa. Los protocolos de esta capa están orientados al control medio de transmisión y correcciones de errores en la transmisión. También incluye protocolos que trabajan con las direcciones MAC.

El estándar ethernet (IEEE802) define las características físicas de cableado, señalización en la capa física y controla cómo se maneja el medio de transmisión en la capa de enlace. Esta función se hace con el standard CSMA/CD (*carrier sense multiple access with collision detection*, acceso múltiple con escucha de portadora y detección de colisiones), que define como los *frame* viajan a través de del medio de transmisión [9]. Debido a que solo una señal puede ser trasmitida en el medio de transmisión, cada dispositivo que se encuentra en la red revisa si el medio está siendo ocupado por otro dispositivo en la red. Si no se detecta nada, entonces cualquier dispositivo puede usar el medio para transmitir. También en el caso de que un dispositivo esté transmitiendo este debe de seguir revisando el medio de comunicación debido a que si hay una colisión con otro equipo este debe de retransmitir la información. Otro factor que es manejado por *ethernet* es la velocidad de transmisión; en la actualidad las velocidades más comunes que se encuentran en el mercado son 10 y 100 Mb/s, 1 y 10 Gb/s.

Capa de red: recibe los paquetes de la capa de enlace y los redirecciona al *host* correspondiente en la red. Los protocolos de estas capas y de la capa de transporte y aplicación son *software*. Los protocolos de capa de red son capaces de determinar la mejor ruta por donde enviar la información.

Los protocolos de esta capa también son llamados protocolos de enrutamiento. El protocolo más usado en esta capa es el IP (*internet protocol*, protocolo internet). Los protocolos de enrutamiento realizan dos funciones básicas: enrutamiento y fragmentación [5].

La fragmentación consiste en que cuando hay dos *hosts* con MTU diferentes, el *host* de mayor MTU debe de fragmentar la trama y el *host* de menor MTU debe de rehacerla. El parámetro MTU (*maximum transmission unit*, unidad máxima de transmisión) es el tamaño máximo en bits que debe tener la trama y que soporta el protocolo o la tecnología de red. El MTU se expresa en *bytes* [10]. El proceso de entregar el paquete a la dirección correspondiente se llama enrutamiento.

Capa de transporte: esta capa asegura una comunicación transparente y segura entre dos *host*s. Consta de dos tipos de protocolos orientados a conexión, como el TCP. Entre las funciones primarias de este están reparar errores en la comunicación, determinar si la transferencia de data fue completada y el control de flujo de datos. Por otro lado, los protocolos no orientados a conexión, como el UDP, no hacen reparación de errores ni control de flujo ni determinan si la transferencia de la data fue completa. Estos transmiten la información más rápido debido a que solo se centran en entregar la data al destino sin tomar en cuenta la integridad de esta [8].

Capa de aplicación: esta capa sirve de interface. En esta capa los paquetes se transforman a un formato legible por el usuario en el caso de que se esté recibiendo. En esta capa también se genera data por el usuario para ser transmitida en forma de paquete.

1.3.3 Multiplexación y puertos

Multiplexación es una técnica que permite a un *host* transmitir múltiples sesiones de comunicación a la vez en un mismo medio. El receptor hace el proceso contrario llamado demultiplexación. Estas operaciones se realizan empleando el concepto de puertos (o conexiones), es decir, un número vinculado a algún protocolo que, cuando se combina con una dirección de IP, permite a los *hosts* determinar con qué protocolo procesarán la información [11].

1.3.4 Segmentación

En el proceso de encapsulación existe un subproceso en la capa de transporte y de red llamado segmentación. La segmentación consiste en que los protocolos de la capa de transporte y de red agregan un encabezado con parámetros relevantes para la comunicación; cada capa agrega su propio encabezado.

A continuación, se describen las cabeceras del segmento de los protocolos TCP, UPD e IP.

1.3.4.1 Segmento TCP

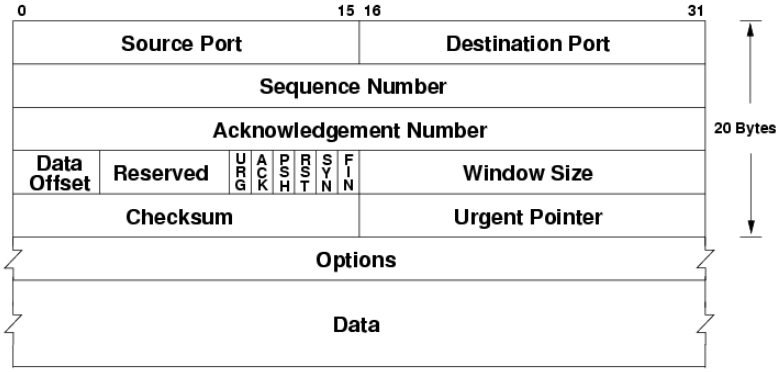


Figura 2: representación gráfica de un segmento TCP [11]

Según IETF [12], los campos del segmento TCP tienen las siguientes funciones:

- ***Souce port/Destinatios port*** **(puerto origen/puerto destino):** el número de puerto por donde los *host*s enviarán y recibirán los datos.
- ***Sequence Number*** **(número de secuencia*):*** es un número de 32 bits indica cuántos segmento serán enviados durante la sección TCP.
- ***Acknowledgment Number*** **(número de acuse de recibo):** Es un número de 32 bits usado por el emisor para solicitar el próximo segmento; este número al final de la sección será igual a *sequence number* + 1.
- ***Data Offset*** **(desplazamiento de datos):** Es un número de 4 bits que indica el tamaño del encabezado TCP.
- ***Reserved*** **(reservado):** son tres bits reservados no utilizados.
- ***Flags*** **(banderas):** son 9 bits de control utilizados para establecer conexión, enviar data y terminar conexión:
    - **URG:** (*urgent pointer* / punto de urgencia) cuando este bit está en uno la data debe procesarse con prioridad a otros datos.
    - **ACK:**

    - **PSH:** bit que controla la función *push*. Este bit le dice al receptor que no llene todos los campos del encabezado para enviar el paquete.
    - **RST:** (*reset*/reinicio) bit usado para reiniciar una sección establecida.
    - **RYN:** bit que indica el inicio se la secuencia de inicialización de sección.
    - **FIN:** bit que indica finalización de la sección.

- ***Windows size* (tamaño de ventana):** es un campo de 16 bits utilizado para regular cuánta data será enviada por el emisor, antes de recibir un *acknowledgment*.
- ***Checksun*:** es un campo de 16 bits, que es generado por el protocolo que envía la data; se utiliza para la detección de error en la comunicación.
- ***Urgent pointer*:** este campo se utiliza para asignar la prioridad del paquete. Solo se utiliza si la bandera URG está activada.
- ***Options*:** es un campo opcional utilizado para optimizar el rendimiento del protocolo TCP en casos especiales.

1.3.4.2 Segmento UDP

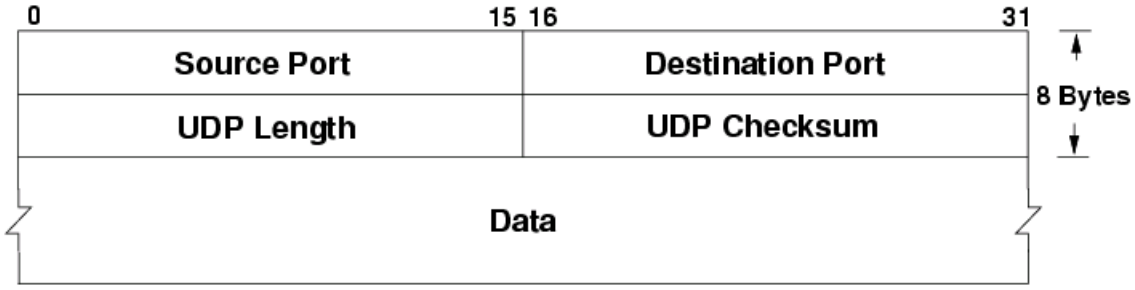


Figura 3: representacion gráfica del segmento UDP [12].

Según IETF [13], los campos del segmento UDP tienen las siguientes funciones:

- ***Source port/destinatios port:*** especifica el puerto de origen y el puerto destino.
- ***UDP length*****:** 16 bits, indica en *bytes* el tamaño del encabezado UDP y de la data encapsulada. El valor mínimo de este campo es 8.
- ***UDP Checksum*****:** es un campo de 16 bits, que es generado por el protocolo que envía la data; es utilizado para la detección de error en la comunicación.

1.3.4.3 Segmento IP

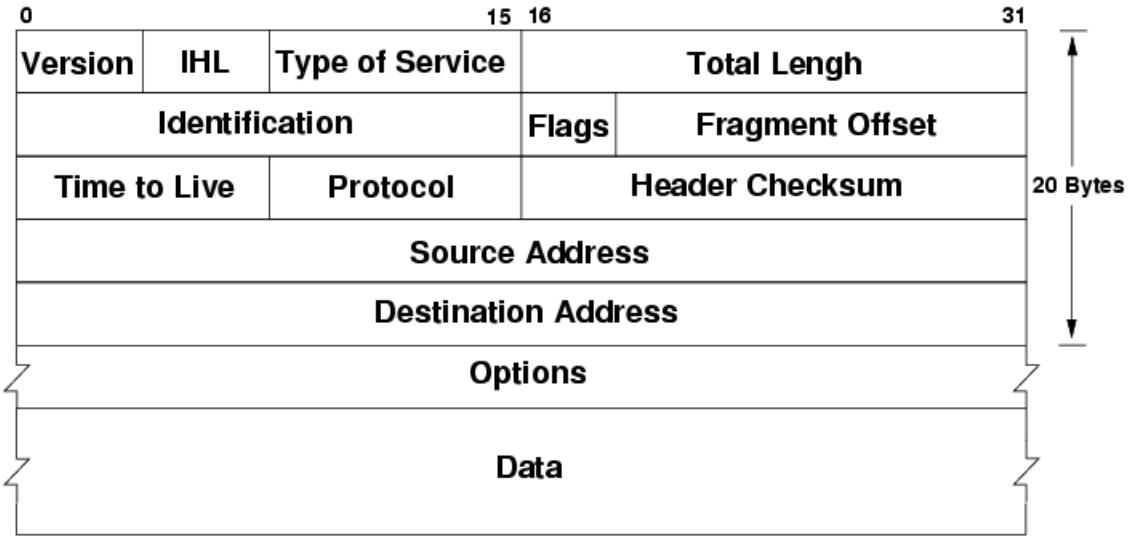


Figura 4: Representación gráfica del segmento IP [13].

Tal como indica IETF [14], los campos del segmento UDP tienen las siguientes funciones:

- ***Version:*** 4 bits versión del protocolo IP a usar en la comunicación.
- ***IHL:*** IP *header length*, campo usado para especificar el tamaño de encabezado IP; consta de 32 bits.
- ***Type of service*****:** los tres primeros bits se denominan prioridad; hoy en día no se utilizan. Los próximos cuatro bits son usados para establecer el tipo de servicio y estos servicios son factor de calidad en la transmisión de datagrama.
- ***Total lengh*** (longitud total): este campo consta de 16 bits; representa la longitud total del datagrama IP en *byte*. Es utilizado para calcular la longitud de la data.

- ***Identification*:** es un identificador asignado al datagrama. Este valor aumenta en 1 cada vez que un datagrama es enviado a un receptor. La longitud máxima que puede alcanzar es de 16 bits.

- ***Flags* (bandera):** este campo es usado para controlar el proceso de fragmentación, de ser necesario.

    - **DF:** es puesto a uno si se debe fragmentar el datagrama.

    - **MF:** es puesto a cero cuando termina el proceso de fragmentación.

- ***Fragment offset* (posición de fragmento):** en caso de que el datagrama sea fragmentado, este campo especifica en términos de 8 *byte* la posición del fragmento enviado respecto del primero. Este campo sirve como parámetro para reensamblar la trama.

- ***Time to live* (tiempo de vida):** indica el máximo número de enrutadores que un paquete puede atravesar. Cada vez que algún nodo procesa este paquete, disminuye su valor en, como mínimo, una unidad. Cuando llegue a ser cero, el paquete será descartado. Típicamente toma el valor 64 o 128 en los datagramas.

- ***Protocol* (protocolo):** Indica el protocolo de la capa superior, que debe procesar el paquete en el receptor.

- ***Header checksum:*** es un campo de 16 bits, generado por el protocolo que envía la data. Es utilizado para la detección de error en la comunicación.

- ***Source address*:** dirección IP del emisor.
- ***Destination adrress:*** dirección IP del receptor.
- ***Options:*** aunque no es obligatoria la utilización de este campo, cualquier *host* debe ser capaz de interpretarlo. Puede contener un número indeterminado de opciones.

1.3.5 Direccionamiento

Direccionamiento consiste en el otorgamiento de las direcciones IP a los *host*s de una red. El otorgar una dirección IP a un *host* requiere tomar en cuenta una serie de parámetros, como se describe a continuación.

Una dirección IP es un número binario de 32 bits que consta de dos secciones: número de red y número de *host*. La forma más común de escribir una dirección IP es en 4 campos llamados octetos, de 8 bits separados por un punto. Cada campo puede ser un número de 0 a 255. Este estilo es llamado notación decimal [8].

Todos los *host*s en una misma red o subred deben de usar el mismo número de red, y cada *host* debe tener un número de *host* único. Para describir la sección de red y *host* de una dirección IP se utiliza un componente llamado mascara de subred, es un número de 32 bits representado de la misma manera de una dirección IP, los bits puesta a 1 de manera sucesiva representarían la sección de red y la cantidad de subred posible y los bits puestos a cero representan la cantidad de *host* que puede haber en la red [5]

Ejemplo de una dirección IP

Notación binaria	**Notación decimal**	**Máscara de subred**
10100110.00010100.11111010.00000001	192.20.250.1	255.255.255.0

Tabla 1: Ejemplo de una direccion IP, autoría propia.

Se logra identificar el prefijo haciendo una operación *and* entre los octetos de la dirección y de la máscara. A continuación, un ejemplo:

Dirección	10100110.00010100.11111010.00000001
Máscara	11111111.11111111.11111111.00000000
Operación *and*	10100110.00010100.11111010.00000000
Resultado en decimal	l92.20.250.0

Tabla 2: Operación *and* para determinar parte de red de una dirección IP, autoría propia.

1.3.6 Clases de direcciones IP

En teoría en una red en el estándar IP se podrían formar más de 4 billones de direcciones. Para tener más control de las direcciones asignadas, el estándar IP subdivide las direcciones IP en 5 rangos llamados clases (A, B, C, D, E) y dos tipos (públicas y privadas). El primer octeto es el identificador de cada clase y tipo [15].

	Rango de IP privadas	Rango de IP públicas	Máscara de red
Clase A	10.0.0.0 – 10.255.255.255	1.0.0.0 – 126.255.255.255	255.0.0.0
Clase B	172.16.0.0 – 172.31.255.255	128.0.0.0 – 191.255.255.255	255.255.0.0
Clase C	192.168.0.0 –192 .168.255.255	192.0.0.0 – 223.255.255.255	255.255.255.0
Clase D	224.0.0.0 – 239.255.255.255		255.255.255.0
Clase E	240.0.0.0 – 255.255.255.255		255.255.255.0
Loopback	127.0.0.0 – 127.255.255.255		255.0.0.0

Tabla 3: Clases y tipos de direcciones IP, autoría propia

Según [16], el rango A es para redes con grandes cantidades de *host*. En la clase A pueden existir 126 y redes y cada una puede albergar 16,777,214 *host*s. El rango ocupa alrededor de la mitad de las direcciones. El rango B toma una cuarta parte de las direcciones, en él rango se pueden formar 16,328 redes con capacidad de 65,534. El rango C toman un octavo del direccionamiento y tiene intervalos de redes pequeños de 255 *host*;se pueden formar 2,097,157 redes de rango C y soportan 254 *host*s.

Los D y E son rangos reservados para usos específicos. La red de Loopback es un estándar reservado para pruebas de equipos de red.

Cuando en el direccionamiento se utilizan las máscaras de red ya establecidas según su rango, se le llama direccionamiento *full clase*. Existen técnicas para alterar las máscaras

que permiten modificar el tamaño de las redes en cualquier rango. Esto permite hacer un uso óptimo de las direcciones y se le llama “direccionamiento de longitud de máscara variable”. Esas técnicas no serán mostradas en este estudio porque no competen a esta investigación.

Las direcciones IP públicas son asignadas a *host*s con acceso directo a internet, y hacen posible comunicarse a dicho *host* a través de internet. Estas son otorgadas por las compañías proveedoras de servicios de internet. Las IP públicas son únicas alrededor del mundo y no pueden ser repetidas.

IP privadas son usadas para redes privadas, como la de una oficina, compañía u hogar. Estas son únicas dentro de la red en la que se esté usando y pueden ser usadas libremente.

Cuando se calcula un direccionamiento para red, la primera dirección IP disponible es el identificador de la red, que no se asigna a ningún *host*. La última es la dirección de broadcast usada para enviar mensajes a todos los *host*s de la red; tampoco se asigna a un *host* en específico [5].

1.3.7 Protocolo NAT

Debido a que las direcciones IP de los servicios que se encuentran en internet deben tener una dirección IP pública, a los usuarios de una empresa, hogar u oficina que está en una red privada les sería imposible acceder a servicios en internet ya que no están direccionados dentro de la misma red. Para dar solución a esto se utiliza el protocolo NAT (*network address translation*/ traducción de direcciones de red), protocolo que hace posible que dos *host*s en redes distintas puedan comunicarse [17] [18].

1.3.8 Enrutamiento

El enrutamiento es el proceso de llevar los paquetes a través de una red TCP/IP, desde el *host* emisor hasta el *host* receptor. La capa de red es la responsable del enrutamiento. Cuando un *host* emite un paquete a un receptor, este primero calcula si el *host* destino está en la misma subnet [19], y dependiendo el resultado pueden suceder uno de los siguientes escenarios:

1) En la misma subred según [5].

    A. Busca la MAC *address* del *host* destino. Usando en el protocolo ARP, que básicamente envía un mensaje a cada *host* de la subred preguntado la MAC *address* de cada uno, posteriormente almacena cada MAC *address* dentro de una tabla asociada con la IP.

    B. El *host* emisor encapsula el paquete en una trama, con la dirección destino. Cuando este paquete pase al medio, será escuchado y procesado por el *host* destino.

2) En otra subred según [5].

    A. El paquete debe ser enviado a un dispositivo llamado *router* (enrutador) también conocido como *default gateway* (puerta de enlace), que básicamente se encarga de interconectar subredes o redes completas, por lo cual el *host*

emisor busca con ARP la MAC *address* del *router*, y envía el paquete para *host* destino al *router*.

B. El *router* compara la dirección destino con su tabla de enrutamiento para entonces calcular la mejor ruta por donde enviar el paquete al *host* destino, utilizando un protocolo de enrutamiento como el NAT.

1.3.9 Modelo cliente servidor

El modelo cliente servidor es un estilo de distribuir información y compartir recurso centralizados en un punto a otros dispositivos pertenecientes a la red. El cliente es la aplicación que solicita servicio de aplicación servidor [7].

1.3.10 Clasificación de las redes

Las redes pueden ser de diversas maneras, pero según su alcance pueden clasificarse básicamente en redes de área local (LAN) y en redes de área amplia (WAN). Un ejemplo de una red LAN sería una red típica de un hogar, de una pequeña empresa y el de un campus universitario. Por otro lado, una red WAN puede ser el internet como tal, la red formada por redes LAN en diferentes localidades [5].

1.4 Funcionamiento de VoIP a través de las capas del modelo TCP/IP

En la comunicación VoIP el audio de la voz es digitalizado y empaquetado en un proceso de encapsulación al igual que cualquier información que se quiera enviar a un *host* en una red IP. A continuación se muestra el proceso de encapsulación de la voz a través de las capas del modelo TCP/IP [7].

- **Aplicación:** Un protocolo de VoIP como SIP, digitaliza la señal de la voz captada a través de transductor como un micrófono, luego la señal captada pasa un proceso de conversión de análogo a digital. El audio digitalizado y codificado será la data que transmitirá al otro extremo.

- **Transporte**: En esta capa los protocolos no orientados a conexión suelen ser los más idóneos para la tarea; como el protocolo UDP, ya que transmite los datos sin tomar en cuenta la integridad del paquete enviado al emisor. Además VoIP es una aplicación en tiempo real, se requiere que los datos fluyan de la manera más rápida posible. También se comienza a segmentar el paquete.

- **Red**: La capa de red se encarga del enrutamiento de los paquetes de voz al destino correcto. El protocolo más usado en esta capa es el IP. También se decide si el paquete será fragmentado (es la mejor práctica evitar la fragmentación).

- **Enlace:** La capa de enlace trabaja con el control del medio. Se puede configurar la red para que los datos de VoIP tenga preferencia sobre las otras comunicaciones.

- **Física:** Se convierte la información a señales eléctricas y se envía al medio de transmisión.

1.4.1 Parámetros VoIP

Hay una serie de parámetros que deben ser tomando en cuenta a la hora de diseñar un sistema VoIP para garantizar una comunicación clara y que otros servicios en la red no sean interferidos [20].

1.4.1.1 Ancho de banda

"El ancho de banda se define como la cantidad de información que puede fluir a través de una conexión de red en un período dado. El ancho de banda de una red generalmente se describe en términos de miles de bits por segundo (kbps), millones de bits por segundo (Mbps), miles de millones de bits por segundo (Gbps)" [21].

1.4.1.2 Códecs

En un principio el ancho de banda necesario para la transmisión de voz en tiempo real era considerablemente elevado, lo que hacía imposible este tipo de comunicaciones sobre redes de datos que no garantizaran una calidad de servicio, como por ejemplo Internet o redes basadas en protocolo IP [22].

Por eso es necesario el uso de códecs en la comunicación VoIP. Estos tienen la tarea de reducir el uso del ancho de banda del audio. Con el uso procesos matemáticos que permiten digitalizar el audio, comprimir el audio y empaquetarlo de manera eficaz [23].

La voz es una señal análoga, mientras que las redes de datos son digitales. Para transformar la señal de la voz a pulsos digitales primero se requiere transformar la voz a una señal eléctrica con el uso de un componente transductor, como un micrófono. Luego la señal eléctrica y aun análoga pasa por un proceso llamado conversión analógico-digital.

Este proceso de conversión analógico digital o modulación por impulsos codificados (PCM) se hace mediante 3 pasos: muestreo, cuantificación y codificación.

Muestreo: El proceso de muestreo consiste en tomar valores instantáneos de una señal analógica, a intervalos de tiempo iguales. A los valores instantáneos obtenidos se les llama muestras [24].

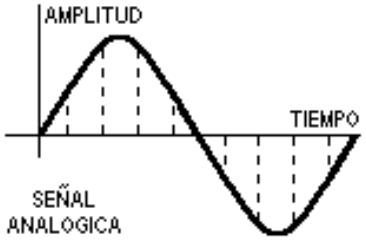


Figura 5: Ilustra el proceso de muestreo de una señal análoga **[25]**

El muestreo se efectúa siempre a un ritmo uniforme, que viene dado por la frecuencia de muestreo fm o *sampling rate*.

La condición que debe cumplir fm viene dada por el teorema del muestreo "Si una señal contiene únicamente frecuencias inferiores a f, queda completamente determinada por muestras tomadas a una velocidad igual o superior a 2f.

De acuerdo con el teorema del muestreo, las señales telefónicas de frecuencia vocal (que ocupan la banda de 300Hz a - 3400Hz), se han de muestrear a una frecuencia igual o superior a 6800Hz [26].

En la práctica, sin embargo, se suele tomar una frecuencia de muestreo igual a 8000Hz. Es decir, se toman 8000Hz muestras por segundo que corresponden a una separación entre muestras de:

$$t = \frac{1}{f} = \frac{1}{8000Hz} = 125\mu s$$

Por lo tanto, dos muestras consecutivas de una misma señal están separadas por 125 µs que es el periodo de muestreo [24].

Cuantificación*:* es el proceso mediante el cual se asignan valores discretos, a las amplitudes de las muestras obtenidas en el proceso de muestreo. El objetivo de esto es que cada valor instantáneo pueda ser representado como data.

La cuantificación se puede clasificar en dos tipos:

Cuantificación uniforme*:* Se utiliza un número finito de valores discretos para representar en forma aproximada la amplitud de las muestras. Para ello, toda la gama de amplitudes que pueden tomar las muestras se divide en intervalos iguales y a todas las muestras cuya amplitud cae dentro de un intervalo se les da el mismo valor [27].

El proceso de cuantificación introduce necesariamente un error, ya que se sustituye la amplitud real de la muestra por un valor aproximado. A este error se le llama error de cuantificación. El error de cuantificación se podría reducir aumentando el número de intervalos de cuantificación, pero existen limitaciones de tipo práctico que obligan a que el número de intervalos no sobrepase un determinado valor.

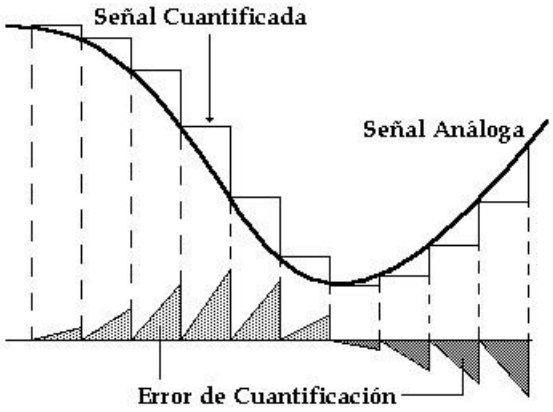


Figura 6: Muestra del proceso de cuantificación uniforme **[28]**

Cuantificación no uniforme: En una cuantificación uniforme la distorsión es la misma cualquiera que sea la amplitud de la muestra. Por lo tanto, cuanto menor es la amplitud

de la señal de entrada mayor es la influencia del error. La situación se hace ya inadmisible para señales cuya amplitud analógica está cerca de la de un intervalo de cuantificación.

Una solución es el uso de la cuantificación no uniforme que toma un número determinado de intervalos y se distribuyen de forma no uniforme aproximándolos en los niveles bajos de señal, y separándolos en los niveles altos. De esta forma, para las señales débiles es como si se utilizase un número muy elevado de niveles de cuantificación, con lo que se produce una disminución de la distorsión. Sin embargo, para las señales fuertes se tendrá una situación menos favorable que la correspondiente a una cuantificación uniforme, pero todavía suficientemente buena.

El proceso de cuantificación no uniforme responde a una característica determinada llamada ley de codificación o de compresión.

Hay dos tipos de leyes de codificación: las continuas y las de segmentos.

- **Continuas**: los intervalos de cuantificación son todos de amplitud distinta, creciendo ordenadamente desde valores muy pequeños (correspondientes a las señales de nivel bajo) a valores grandes (correspondientes a las señales de nivel alto).
- **Segmentos:** la gama de funcionamiento se divide en un número determinado de grupos y dentro de cada grupo los intervalos de cuantificación tienen la misma amplitud, siendo distinta de unos grupos a otros. Son las de uso más común.

Ley A (*a-law*) y ley μ (*u-law*)

Según [29], actualmente las dos leyes de compresión de segmentos más utilizadas son la ley A (*a-law*) y la ley μ (*u-law*). La ley A (*a-law*) se utiliza principalmente en los sistemas PCM europeos, y la ley μ (*u-law*) se utiliza en los sistemas PCM americanos y japonés.

Estos dos estándares internacionales, conocidos formalmente como recomendación G.711, están documentados en ITU-T 89. Emplean logaritmos basados en funciones para codificar las muestras de audio.

Indican que las amplitudes bajas de las señales del habla contienen más información que las amplitudes altas. Es por esto por lo que la cuantificación no lineal tiene sentido. Imagine una señal de audio enviada por una línea telefónica y digitalizada en muestras de 14 bits. Cuanto más fuerte sea la conversación, tanto mayor será la amplitud, y mayor el valor de la muestra. Puesto que las amplitudes altas son menos importantes, pueden ser toscamente cuantificadas. Si la muestra más grande, que es de $2^{14} - 1 = 16{,}383$, se cuantifica a 255 (el número más grande de 8 bits), entonces el factor de compresión es de $14/8 = 1.75$.

Cuando es decodificado, un código 255 será muy diferente del original. Debido a la tosca cuantificación, las muestras grandes generan un ruido de cuantificación alto.

Las muestras más pequeñas deben ser finamente cuantificadas, por lo que acaban con un bajo ruido de cuantificación

El algoritmo de la a ley μ se representa matemáticamente de la manera siguiente:

$$y = F(x) = sgn(x)\frac{\ln(1+\mu|x|}{ln(1+\mu)}\ para - 1 \leq x \leq 1$$

Se usa la función inversa para descomprimir:

$$y^{-1} = sgn(x)\left(\frac{1}{\mu}\right)\left((1+\mu)^{|y|} - 1\right) para\ -1 \leq x \leq 1$$

Donde:

μ = 255 (8 bits) es el estándar usado en Norteamérica y Japón.

sgn = es la función signo.

Para calcular y producir un código de 8 bits en el mismo intervalo [−1, +1].

La salida es posteriormente escalada al rango [−256,255]. La Figura 7 muestra esta salida como una función de la entrada para los tres valores de μ: 25, 255, y 2555. Es evidente que los valores grandes de μ producen una cuantificación tosca para grandes amplitudes.

Tales valores asignan más bits para las amplitudes más pequeñas e importantes. El estándar G.711 recomienda el uso de μ = 255. El diagrama muestra sólo los valores de entrada no negativos. La zona negativa del diagrama tiene la misma forma, pero con entradas y salidas negativas [29].

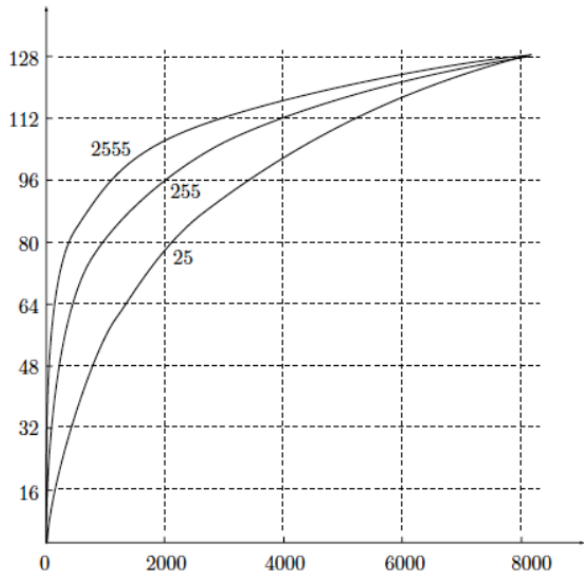


Figura 7: μ-law para valores de μ de 25, 255, y 2555 **[29]**

Codificación*:* es el proceso mediante el cual se representa una muestra cuantificada, mediante una sucesión de "1's" y "0's", es decir, mediante un número binario. La codificación es necesaria ya que esta traduce la información de la cuantificación a un formato legible para un receptor [30].

1.4.1.3 Cancelación de eco

Es el sistema por el cual el dispositivo emisor guarda la información que envía en memoria y es capaz de detectar en la señal de vuelta la misma información (tal vez atenuada y con ruido). El dispositivo filtra esa información y cancela esas componentes de la voz. Requiere mayor tiempo de procesamiento. Además de la

ejecución de la conversión de analógico a digital, el Códec comprime la secuencia de datos, y proporciona la cancelación del eco [31].

1.4.1.4 Latencia

"Es el tiempo necesario para que la voz viaje de un extremo al otro, incluyen los tiempos necesarios para la compresión, transmisión y descompresión. Este tiempo tiende a minimizarse, pero jamás podrá ser suprimido. Se recomienda que la latencia ten un valor menor a los 120ms." [32]

1.4.2 Protocolos VoIP

1.4.2.1 Protocolo Sip

[33]El protocolo de inicio de sesiones (SIP, *session initiation protocol*) desarrollado por el IETF es un protocolo de señalización de capa de aplicación que define la iniciación, modificación y la terminación de sesiones interactivas de comunicación multimedia. Inicialmente fue publicado en febrero del 1996 en el RFC 2543.

SIP hace posibles comunicaciones multimedia gracias a dos protocolos que son RTP y SDP. El protocolo RTP se usa para transportar los datos de voz en tiempo real, mientras que el protocolo SDP se usa para la negociación de las capacidades de los participantes, tipo de codificación, etc. Como consecuencia de eso el protocolo SIP usa una gran cantidad de cabeceras y puertos para funcionar.

1.4.2.2 IAX 2

La segunda versión del protocolo de comunicación entre Asterisk (Inter-Asterisk eXchange) se conoce como IAX2. IAX2 es una alternativa al protocolo de señalización SIP. Este fue creado como parte del desarrollo de Asterisk. A diferencia del SIP, que usa dos flujos de datos para voz y otros dos para señalización, IAX2 usa solo un par de flujos donde voz y datos coexisten. Esta forma de enviar tanto las conversaciones como la señalización por el mismo canal se conoce como *in-band*, en contraste con el método que usa SIP, el *out-of-band* [34].

Debido a su diseño, IAX2 es la opción más adecuada en regiones en desarrollo donde existe presencia de NAT. Además, IAX2 es capaz de empaquetar llamadas simultáneas en un solo flujo de paquetes IP. Este mecanismo es conocido como “trunking” y su implementación resulta en ahorros en el consumo de ancho de banda.

1.4.2.3 Funcionamiento de llamadas con el protocolo SIP

A continuación se desglosará el proceso que hace posible una comunicación VoIP con el protocolo SIP.

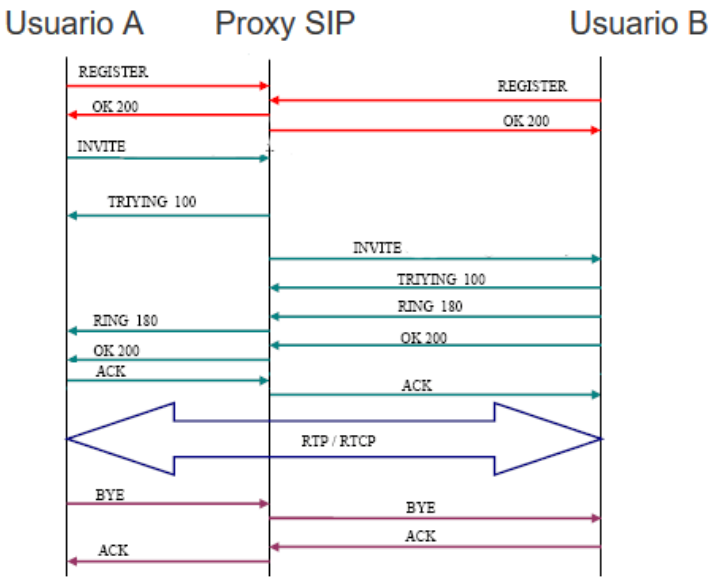


Figura 8: Proceso de llamada VoIP con el protocolo SIP **[35]**

[36]Las dos primeras transacciones corresponden al registro de los usuarios. Los usuarios deben registrarse para poder ser encontrados por otros usuarios. En este caso, los terminales envían una petición *register*. El servidor Proxy SIP, que actúa como *register*, consulta si el usuario puede ser autenticado y envía un mensaje de *OK* en caso positivo.

La siguiente transacción corresponde a un establecimiento de sesión**.** Esta sesión consiste en una petición INVITE del usuario al proxy. Inmediatamente, el proxy envía un TRYING 100 para parar las retransmisiones y reenvía la petición al usuario B. El usuario B

envía un *ringing* 180 cuando el teléfono empieza a sonar y también es reenviado por el proxy hacia el usuario A. Por último, el OK 200 corresponde a aceptar la llamada (el usuario B descuelga).

En este momento la llamada está establecida y pasa a funcionar el protocolo de transporte RTP con los parámetros (puertos, direcciones, códecs, etc.) establecidos en la negociación mediante el protocolo SDP.

La última transacción corresponde a una finalización de sesión**.** Esta finalización se lleva a cabo con una única petición BYE enviada al *proxy*, y posteriormente reenviada al usuario B. Este usuario contesta con un OK 200 para confirmar que se ha recibido el mensaje final correctamente.

1.4.2.4 Funcionamiento de llamadas protocolo IAX2

El protocolo IAX2 también consta de varias transacciones para operar.

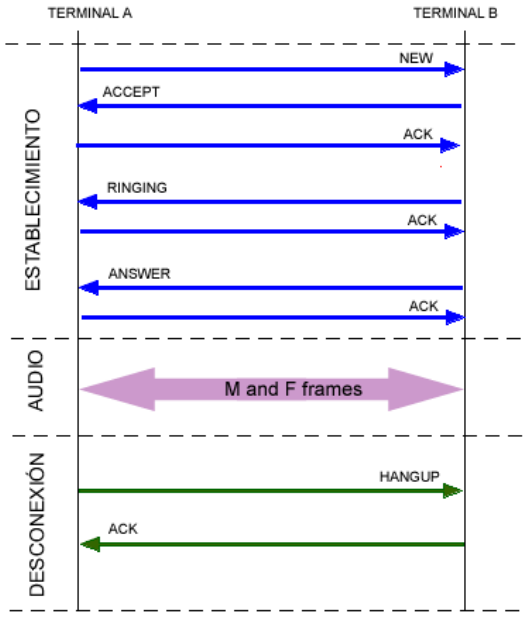


Figura 9: Proceso de llamada VoIP con el protocolo IAX2 **[25]**

[37]En la sección de establecimiento de la llamada, **e**l terminal A inicia una conexión y manda un mensaje “new”. El terminal llamado responde con un “accept” y el llamante le responde con un “Ack”. A continuación, el terminal llamado da las señales de “ringing” y el llamante contesta con un “ack” para confirmar la recepción del mensaje. Por último, el llamado acepta la llamada con un “answer” y el llamante confirma ese mensaje.

En el flujo de datos o flujo de audio **s**e mandan los frames M y F en ambos sentidos con la información vocal. Los frames M son mini-frames que contienen solo una cabecera de 4 *bytes* para reducir el uso en el ancho de banda. Los frames F son frames completos que

incluyen información de sincronización. Es importante volver a resaltar que en IAX este flujo utiliza el mismo protocolo UDP que usan los mensajes de señalización, evitando problemas de NAT.

Liberación de la llamada o desconexión: la liberación de la conexión es tan sencillo como enviar un mensaje de “hangup” y confirmar dicho mensaje.

1.4.3 Cálculo de consumo de ancho de banda VoIP

Según [38] es posible calcular el ancho de banda necesario para llamada VoIP. Es necesario saber el tamaño de las cabeceras de los protocolos que intervienen en la comunicación y algunos parámetros ofrecidos por códec a utilizar.

- 40 *bytes* para encabezados IP (20 *bytes*) / User Datagram Protocol (UDP) (8 *bytes*) / Real-time Transport Protocol (RTP) (12 *bytes*).
- 18 encabezados de los *bytes* para Ethernet L2, que incluyen 4 *bytes* de la secuencia de verificación de tramas (FCS) o de la verificación por redundancia cíclica (CRC).

Fórmulas para el cálculo del uso de ancho de banda:

1. Tamaño total del paquete(Tt) = (encabezado L2: MP o FRF.12 o Ethernet) + (encabezado IP/UDP/RTP) + (tamaño de carga útil de voz)
2. PPS (cantidad de paquetes por segundo) = (velocidad de bits en códec) / (tamaño de la carga útil de voz)
3. Ancho de banda(BW) = tamaño de paquete total * PPS

1.4.3.1 Ejemplo de cálculo de ancho de banda con el códec G711 (*u-law*)

Parámetros códec G711 (*u-law*)			
Tamaño de muestra (*byte*)	Intervalo de muestra (ms)	Tamaño carga útil de voz (byte)	MOS
80	10	160	4.1

$$velicidad\ bits\ kodec = \frac{tamaño\ de\ muestra * 8}{intervalo\ de\ muestras} = \frac{80\ bytes * 8}{10ms} = 64kbps$$

Calcular Tt según la fórmula 1

$$Tt = IP\ (20\ bytes) + \ UDP\ (8\ bytes) + RTP\ (12\ bytes) + L2(18bytes)$$
$$+ carga\ util\ codec(160bytes) = 218\ bytes$$

Hallas PPS según la fórmula 2

$$\frac{Vecida\ códec}{tamaño\ carga\ útil * 8} = \frac{64kbps}{160 * 8} = 50$$

Ya se puede calcular el BW necesario con la fórmula 3

$$BW = 218\ bytes * 50pps = 87.2\ kbps$$

El parámetro MOS es una escala para calificar la calidad de voz de las conexiones de teléfono. Con MOS, una amplia gama de oyentes juzga la calidad de un ejemplo de voz en

una escala que va del uno (mala) al cinco (excelente). Las calificaciones se hacen un promedio para proporcionar el MOS.

1.5 Componentes sistemas VoIP

1.5.1 *Gateway*

Gateway (puerta de enlace) en términos generales son dispositivos que interconectan redes o dispositivos que traban con tecnología de comunicación diferentes. Se usan estos dispositivos para interconectar centrales VoIP con la red PSTN o la red celular.

1.5.1.1 FXO-FXS Gateway

Es un dispositivo utilizado para interconectar una central VoIP con la red PSTN. Estos constan de uno o más puertos FXO (*foreign exchange office*) y son puertos por donde se recibe la señal de la PSTN. Puertos FXS (*foreign exchange suscribir*) son los puertos donde manda la línea telefónica PSTN. Los puertos FXS pueden usarse para conectar teléfonos análogos con la central telefónica IP; los puertos de red, comúnmente puerto RJ45, para la comunicación IP.

1.5.1.2 GSM Gateway

Es dispositivo utilizado para interconectar una central VoIP con la red celular GSM. Esta consta de uno o varios puertos para insertar tarjetas SIM, cada tarjeta representa una línea celular diferente. Los puertos para la comunicación con la central IP pueden ser puertos RJ45 o USB.

1.5.2 Teléfono IP/ *softphone*

Los teléfonos IP son teléfonos especialmente diseñados para trabajar con VoIP. Los *softphone* son *softwares* que emulan el funcionamiento de un teléfono IP.

1.5.3 Central VoIP

Una central VoIP es un servidor que gestiona toda la actividad de un sistema VoIP.

1.6 Asterisk

Asterisk un entorno de trabajo de código abierto para crear aplicaciones de comunicaciones VoIP. Convierte un computador ordinario en un servidor de telecomunicaciones. Creado por en 1999 por Mark Spencer y actualmente sigue desarrollándose por la empresa Digium. Originalmente diseñado para el sistema Linux, hoy en día se puede instalar en diversos sistemas operativos.

1.6.1 Estructura de Asterisk

Asterisk está basado en una estructura modular alrededor de un núcleo. El núcleo controla la actividad Asterisk, cada vez que requiere de realizar una función específica carga el módulo correspondiente. Por ejemplo, para usar líneas análogas con Asterisk se usa modulo para esa tarea, si se requiere que Asterisk lleve un registro de las llamadas realizadas, se requiere de otro módulo que haga esta función. Esto permite que Asterisk sea escalable porque permite quitar o agregar módulos según la aplicación y extensible porque para el diseño de nuevos módulos no es necesario conocer todo el código de Asterisk [39].

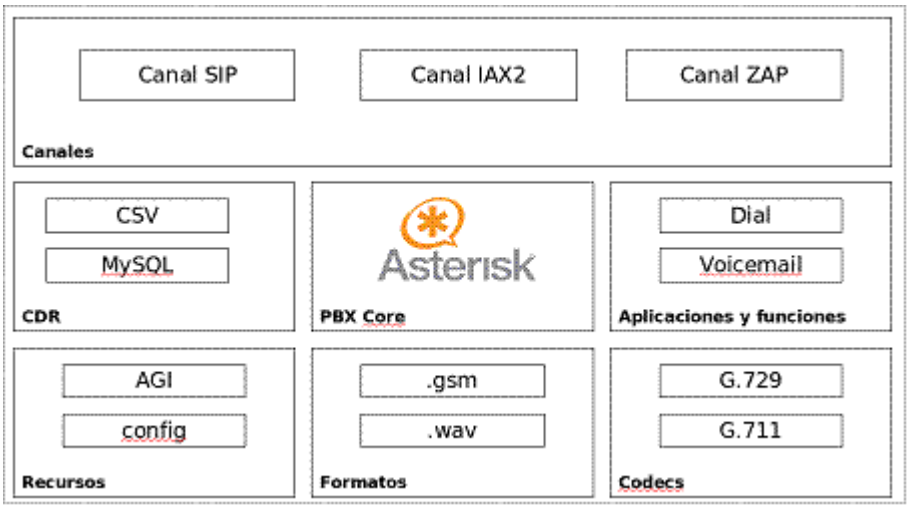


Figura 10: Estructura de Asterisk **[40]**

Los módulos de Asterisk pueden dividirse en 6 grupos:

- **Recursos:** Aportan funcionalidades adicionales al núcleo, como la posibilidad de leer archivos de configuración.

- **Canales**: permiten a Asterisk manejar dispositivos de una determinada tecnología. Por para manejar dispositivos que trabajan con el protocolo SIP se utiliza el módulo chan_sip, para IAX2 chan_iax y para canales análogos/digitales chan_zap.

- **Aplicaciones y funciones:** Estos módulos conforman la "la caja de herramientas" de Asterisk, ya que aportan las distintas herramientas de configuración de Asterisk. Por ejemplo, el Dial Plan que es la aplicación que ejecuta el plan de cómo se procesara las llamadas entradas y salientes.

- **CDR:** Estos módulos controlan la escritura de registros telefónicos generados por Asterisk a diferentes formatos, por ejemplo, un fichero CSV, una base de datos MySQL, etc.

- **Códecs:** permiten que Asterisk pueda codificar y decodificar la información de audio/video que envía y recibe.

- **Formatos:** estos módulos posibilitan a Asterisk trabajar con diferentes formatos de ficheros distintos, como formato mp2, alaw, wav, etc.

1.6.2 Gestión de Asterisk

Asterisk fue diseñado originalmente para trabajar en sistemas basados en Linux, típicamente en los ambientes Linux la interfaz para usuario está basado en líneas de comando. Asterisk tiene dos maneras de ser gestionado.

- **Línea de comandos:** la gestión a través de la línea de comando está basado en la escritura de las configuraciones de Asterisk en archivos de configuración, Asterisk lee los archivos de configuración y ejecutas las intrusiones. También Asterisk cuenta con una herramienta de gestión con el uso de comandos. Con dicha herramienta se puede lograr que Asterisk ejecute determinas acciones al instante y revisar el estado de las operaciones que Asterisk está ejecutando en tiempo real.

- **Interfaz web:** existen varias versiones de interfaces web (WEBUI), estas ofrecen una interface gráfica en una página web para la configuración y administración de Asterisk.

Estas básicamente funcionan presentando un formulario con las opciones de configuración de los módulos de Asterisk, luego que el usuario selecciona las opciones que les conviene la WEBUI se encarga de escribir los archivos de configuración correspondiente.

1.7 Raspberry Pi

Raspberry Pi (RPI) es una computadora de placa reducida. Lo que significa que es un computador completo en una sola placa de circuito impreso. Es desarrollado en el Reino Unido por la fundación también llamada Raspberry Pi. Con el objetivo de estimular el conocimiento de la programación y el diseño digital. La característica principal de la RPI es tamaño de apena 5.6cm/8.25cm, aproximadamente el tamaño de una tarjeta de crédito. El *hardware* del RPI es basado en una arquitectura SoC (*system on chip*, sistema en un chip); esta arquitectura centra todos componentes de un computador en un solo chip [41].

Existen varios modelos de RPI. En este trabajo investigación solo se menciona el modelo 3B de RPI. Este modelo es uno de lo más potentes en la actualidad y cumple con las necesidades de este proyecto de investigación.

Características del modelo Raspberry PI 3b	
SoC	Broadcom BCM2387 chipset Procesador 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1
Ram	1GB LPDDR2
Conectividad	10/100 Ethernet WiFi 40 GPIO Audio 3.5 Jack Bluetooth 4 puertos USB
Almacenamiento	Memoria Micro SD
Alimentación	Micro USB socket 5V1, 2.5A

Tabla 4: Características de 3B de Raspberry PI 3

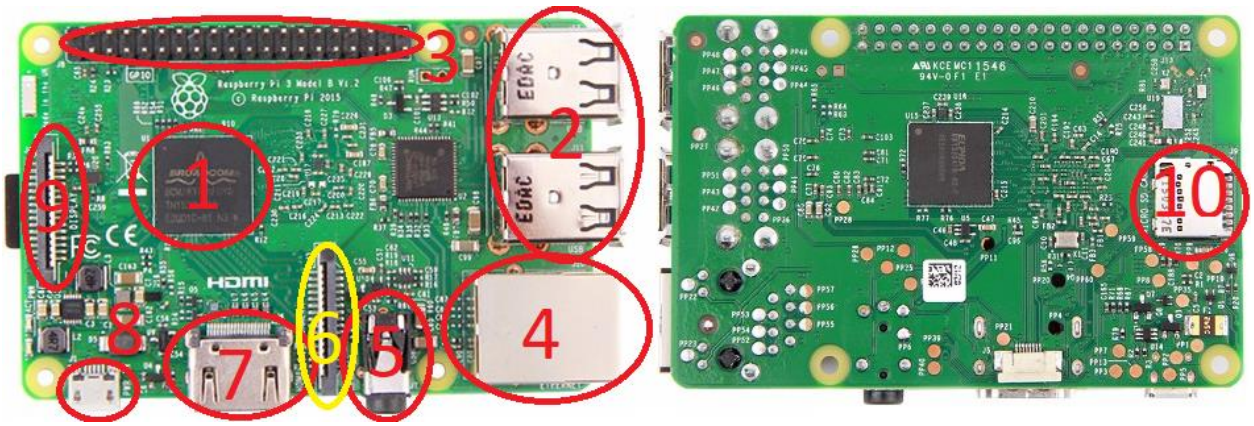


Figura 11: Vista de Raspberry PI 3B y sus partes

1. Soc
2. Puerto USB
3. Puerto GPIO
4. Puerto Ethernet
5. Jack 3.5/Audio
6. Puerto para cámara
7. Puerto HDMI
8. Puerto de alimentación micro USB
9. Puerto para pantallas LCD
10. Puerto para memoria micro SD

1.8 Linux

El sistema operativo más común para RPI está basando en el kernel (núcleo) Linux. El kernel es una parte fundamental del sistema operativo; se encarga de manejar los recursos y permitir que los programas hagan uso de estos, siendo los principales recursos CPU, memoria, dispositivos de entrada/salida, etc. Además, el kernel es el encargado de proporcionar protección mediante diferentes niveles de acceso compartido (multiplexado) a los recursos [42].

Antes de que el kernel sea utilizable es necesario acompañarlo con otros softwares que hagan de interface para el usuario y además definir cuáles servicios serán brindandos al usuario e instalar todo junto en un computador. Las distribuciones Linux son sistemas operativos basados en núcleo Linux. Hoy en día existe una gran variedad de comunidades dedicadas al desarrollo de las distribuciones.

Linux es distribuido bajo licencia GPLv2, lo significa que puede ser modificado o distribuido por cualquiera. Es muy popular en el mundo de las telecomunicaciones. Es fácil pensar que las telecomunicaciones son solo un conjunto de dispositivos electrónicos y antenas distribuidas para transmitir y recibir señal, pero en realidad esos dispositivos necesitan ser supervisados, administrados y operados. Ahí es donde entra Linux, un sistema moldeable a diversas necesidades [43].

Características principales de Linux:

- **Multiusuario**: Dependiendo del equipo disponible, Linux puede soportar desde uno hasta más de 100 usuarios, ejecutando cada uno de ellos un conjunto diferente de programas.

- **Multitarea**: Describe la habilidad de ejecutar, aparentemente al mismo tiempo, numerosos programas sin obstaculizar la ejecución de cada aplicación. Esto se conoce como multitarea preferente, porque cada programa tiene garantizada la posibilidad de correr.

- **Multiplataforma**: Corre en muchas CPU distintas (Intel 386/486/Pentium y como k6/7 de AMD, procesadores de arquitectura ARM, procesadores de la familia Motorola, Sun Sparc, etc.).
- **Shell**: Esta es la utilidad que procesa las peticiones de los usuarios. En los sistemas operativos basados en Linux es común usar instrucciones de líneas de comandos. Esta es una interfase en la que el usuario y el sistema solo interactúan con textos.

1.9 RasPBX

Existe diversas distribuciones para RPI; una de las más destacadas es Raspbian. Fue creado por Mike Thompson and Peter Green como un proyecto independiente a Raspberry pi y su desarrollo un sigue activo. Esta distribución está optimizada especialmente para *hardware* de RPI.

RasPBX es una distribución basada en Raspbian que viene con una versión optimizada para RPi de Asterisk.

1.10 VPN

En tiempo atrás, cuando se requería tener dos redes en localidades distante entre sí, la solución típica para ese entonces era hacer una conexión directa, cableando líneas de comunicación dedicadas solo para la interconexión de las redes entre las localidades. Otra manera que se utilizaba para que usuarios pudieran acceder a una red remota era con conexión dial up, que utilizaba como medio una llamada telefónica. Este método utiliza dispositivos módem que se encargan de codificar/decodificar la información en frecuencia de audio.

Las dos soluciones mencionadas anteriormente implicaban un alto costo de operación o implementación, pero con el avance del internet se desarrollaron tecnologías más económicas que permitían el acceso remoto a redes.

Hoy en día cuando se requiere conectar dos redes que están en localidades distantes o usuarios que están fuera de la localidad de una red, una herramienta eficaz para esta tarea son las VPN (*Virtual Private Network*, red privada virtual).

[44] Dice que VPN también se puede describir como un conjunto de conexiones lógicas aseguradas por *software* que establece privacidad salvaguardando los puntos finales de la conexión. Hoy Internet es el medio de red utilizado, y la privacidad se logra mediante métodos criptográficos. Por otro lado [45]define VPN como una tecnología de red que permite una extensión segura de una red LAN (*local area network*, red local) sobre una red pública o no controlada como el Internet, mediante un proceso de encapsulación y de encriptación de los paquetes de datos.

Una VPN es:

- **Virtual**: esto es, porque no es una conexión real directa de red en entre los *host*s, solo hay conexiones virtuales suplidas por los *softwares* de VPN, normalmente sobre conexiones públicas de internet.

- **Privada**: esto porque solo los usuarios con privilegios podrán tener acceso a leer los paquetes encriptados de la comunicación en la VPN.

El funcionamiento de una VPN básicamente se divide en tres fases:

Autenticación

En esta fase se verifica que el usuario o *host* que quiere acceder a la VPN esté autorizado. Para esta tarea primero se establece una sección entre el *host* cliente y el servidor VPN, luego el servidor verifica las credenciales del usuario o el *host*. Las credenciales pueden ser usuario/clave o un certificado encriptado, que un archivo que contiene credenciales de un usuario o *host*.

Tunelización (*tunneling*)

La tunelización es un método utilizado para encapsular paquetes de datos dentro de otros paquetes de datos, los cuales son enviados al *host* destino utilizando el mismo u otro protocolo al que fue encapsulado originalmente [46]. Esto tiene mucha utilidad ya que permite el transporte de datos con protocolos con diferentes esquemas de direccionamiento. Por ejemplo, para enviar un paquete desde una red LAN A otra B a través de internet es necesario encapsular los paquetes de A en paquetes NAT. Cuando el paquete llega al *host* destino es desencapsulado al paquete original.

La tunelización consta de tres partes:

- ***Passenger protocol***– Datos originales a transportar (IPX, NetBeui, IP).
- ***Encapsulating protocol***– El protocolo que envolverá (encapsulará) al paquete original.
- ***Carrier protocol***– El protocolo usado para el transporte de los datos.

Encriptación

[47]Encriptación es una técnica de criptografía que codifica una comunicación en texto plano en una comunicación cifrada, con el uso de algoritmos de encriptación. Los algoritmos de encriptación utilizan lo que se denomina llave; la llave es un archivo con los parámetros que determina la función de salida de un algoritmo de encriptación. Para decodificar la información encriptada el receptor debe de aplicar el algoritmo inverso con la llave adecuada. Los algoritmos de encriptación se pueden clasificar en:

- Simétrica: Cuando el receptor y el emisor usan llaves diferentes a descifrar la comunicación.
- Asimétrica: Cuando tanto el emisor como el receptor usan la misma llave para descifrar la comunicación.
- *Haching*: es el uso de algoritmos para encriptar texto, solo el contraalgoritmo puede descifrar el texto.

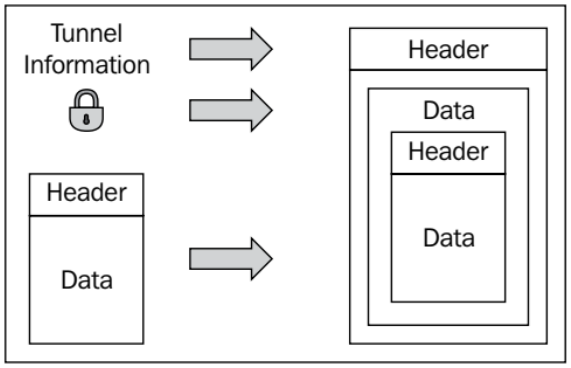


Figura 12: Proceso de tunelización de datos [44]

1.10.1 OpenVPN

OpenVPN es un *software* diseñado para la implementación de redes VPN. OpenVPN es de código libre diseñado originalmente por James Yonan en el año 2001. Hoy en día es desarrollado por una gran comunidad de programadores. OpenVPN utiliza una encriptación simétrica con el protocolo SSL/TLS junto con un sistema de llaves PKI. OpenVPN está disponible para una gran variedad de plataformas Android, Linux, Windows, MacOS [45] [44].

1.10.1.1 Sistema PKI

Un sistema PKI (public key infrastructure/Infraestructura de llaves públicas). Es un sistema en que se generan certificados y llaves firmados digitalmente por un *certificate authority* /certificado de autoridad (CA). Lo que significa que cada certificado y llave están encriptado en un *hash*, así se garantiza que el certificado o llaves no han sido modificados [48].

1.10.1.2 Funcionamiento SSL/TLS

El protocolo SSl/TLS básicamente realiza las siguientes fases en su operación según [45]:

1. El cliente inicia una sección con el servidor y solicita el certificado de este;
2. El cliente verifica el certificado del servidor, si es aceptado el cliente envía su certificado al servidor;
3. El servidor verifica el certificado del cliente;
4. Ambos envían sus llaves públicas, utilizando un *hash* llamado Diffie Hellman en la comunicación. Así se logra un intercambio de llaves seguros, y
5. Se establece el tune VPN entre los dos *host*s. Interfaces Tum/Tap.

Las interfaces TUN y TAP son interfaces virtuales de red. Las interfaces TUN solo trabajan con trafico IP y las TAP trabajan con tramas Ethernet. El uso de esta es necesario ya que las VPN emulan que el *host* esté conectado a través de un cable a una red [45].

2 Diseño de un sistema VoIP

2.1 Esquema del sistema voip

Para el diseño de un sistema VoIP con la característica que cumpla con las pautas definas en la investigación se utilizará el diagrama de la figura 13.

Este consta de los siguientes componentes:

1) *router*: encargado de enrutar las comunicaciones del sistema hacia el internet.
2) Gateway FXS-FXO: se usará el SPA300, este permite digitalizar una línea telefónica y además usar un teléfono análogo como teléfono IP. este se conecta a la línea telefónica en un puerto FXS, se le conecta el teléfono análogo en el puerto fijo y se conecta a red mediante el puerto Ethernet.
3) Raspberry Pi: este servirá como servidor VPN y central telefónica IP. Estará conectado al router mediante el puerto Ethernet con un cable.
4) Módem gsm: servirá como GSM gateway y va conectado a unos de los puertos USB del Raspberry.
5) Teléfono análogo: servirá de extensión
6) El internet: será la vía de acceso al sistema
7) Los usuarios móviles. Serán quienes usarán el sistema con sus dispositivos móviles, Celular, portátil, tableta, etc.

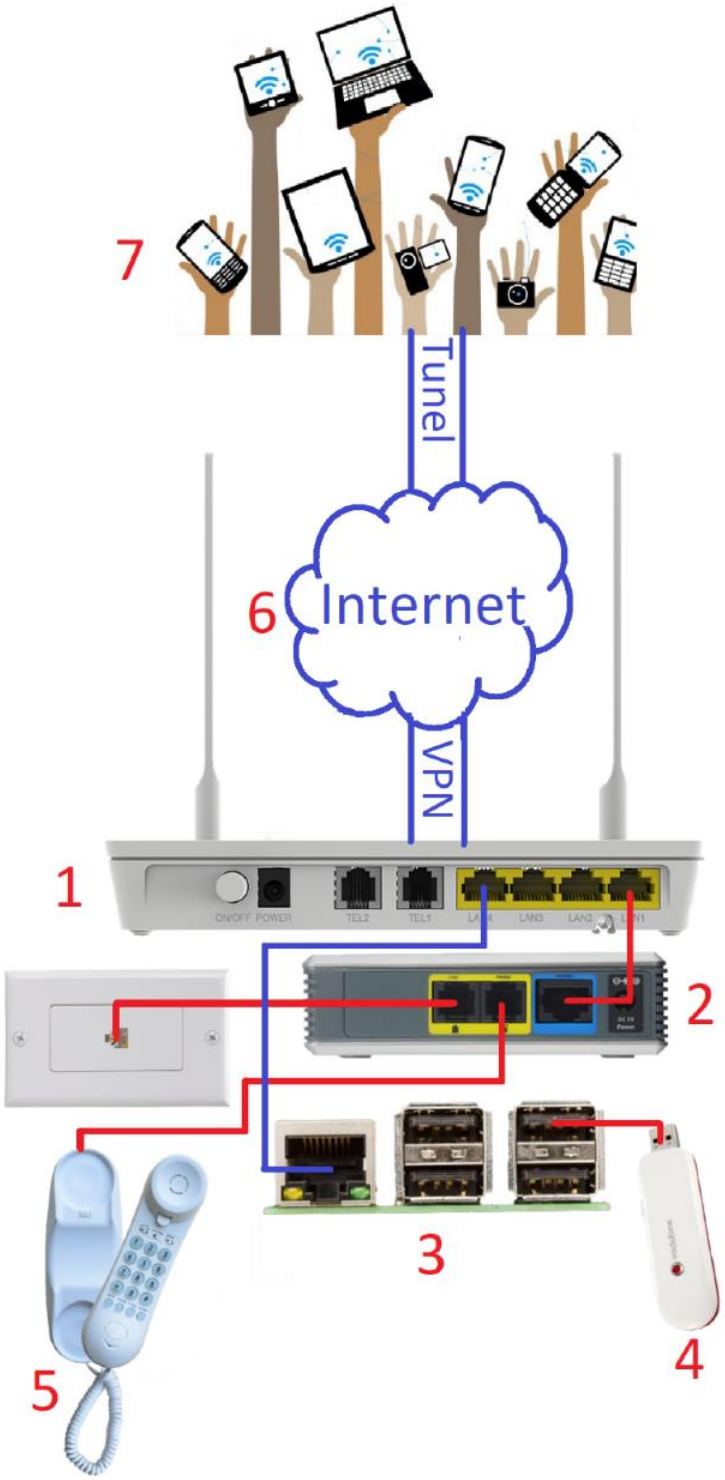


Figura 13: Diagrama pintoresco del sistema VoIP.

2.2 Instalación del Raspbx

Primero es necesario descargar de la sección de descarga de la web http://www.raspberry-asterisk.org/ la última versión del sistema operativo Raspbx que se instalará en la RPI, en este caso la versión raspbx-04-04-2018. Raspbx se descarga en un archivo comprimido en formato zip. Luego de descargado este debe ser descomprimido y se obtendrá como resultado un archivo con extensión .img, los archivos en este formato lo que hacen es contener en un solo archivo una copia de un medio de almacenamiento informático como disco duro, discos ópticos, memorias USB, etc. Del archivo .img se grabará una copia del Raspbx en la memoria SD del RPI con el *software* Win32, como se detallas a continuación:

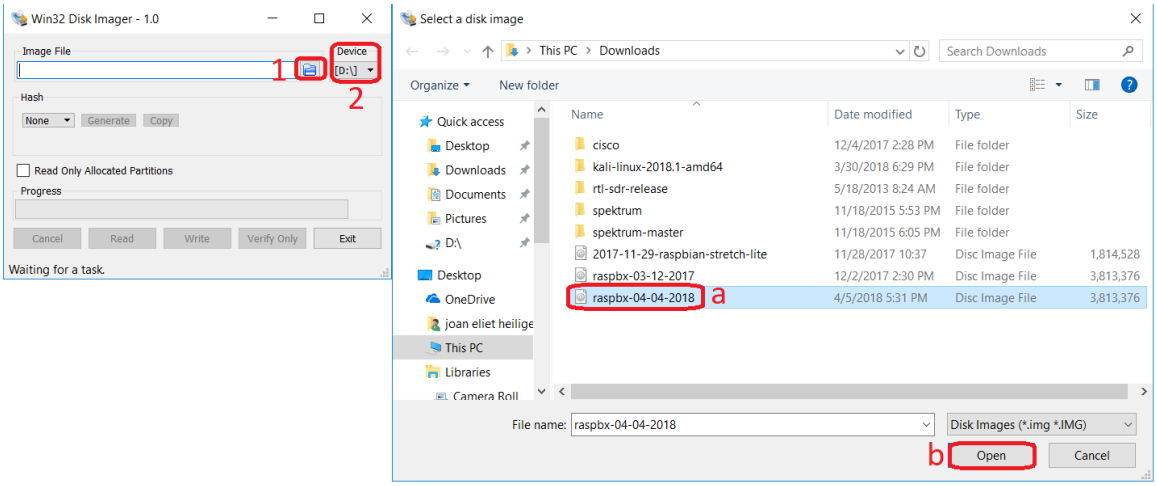


Figura 14: Captura de pantalla del *software* WIN32

Siguiendo los pasos de la figura 14:

1) Seleccionar imagen del sistema operativo a cargar.
   a. Se abre una ventana en la cual debemos buscar la ubicación donde se encuera el archivo .img.
   b. Seleccionar el archivo .img haciendo clic sobre este, luego clic en “open”.
2) Seleccionar dispositivo donde se grabará la imagen de sistema operativo.

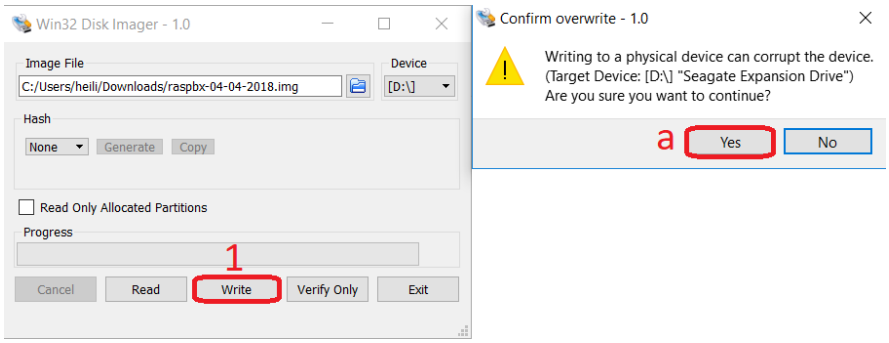


Figura 15: captura de pantalla del software WIN32

Luego se continúa con los pasos de la figura 15

1) clic en el botón Write para comenzar a transferir imagen de Raspbx a la memoria SD de la RPI
   a. al dar clic en el botón “Write” aparecerá una pantalla de advertencia indicándonos que se borraran todos los archivos de la imagen seleccionada, se debe hacer clic en el botón “Yes” para que comience el proceso. Cuando el proceso el programa termine lo indicará con un mensaje en pantalla.

2.3 Configuración de *router*

Es necesario realizar varias configuraciones en el *router* que instalan los proveedores de internet (PSI) cuando se les solicita un servicio. Estas configuraciones permitirán acceder a nuestro sistema desde internet, además de hacer un direccionamiento óptimo para el sistema.

2.3.1 Configuración de direccionamiento

Los *router* tienen la opción de hacer direccionamiento dinámico de la red del usuario con el protocolo DHCP. Tanto la RPI como el ATA requieren una dirección IP fija, porque si no cada vez que la RPI o el ATA cambiaran de dirección IP habría que configurar los *host*s asociados al sistema o el enlace y entre el ATA y la RPI.

El protocolo DHCP tiene una opción para otorgar siempre la misma dirección IP a los *host*s definidos en una lista de direcciones. A continuación de se mostrará cómo se logra dicha configuración. En este caso en un *router* modelo HG8245H de Huawei, instalado por el PSI Claro.

Este *router* se configura mediante WEBUI, para acceder a la interface de configuración es necesario conectar un computador a uno de los puertos ethernet del *router* o bien a mediante WIFI. En este caso se conectó un computador con el sistema operativo Windows 10 mediante wifi. Luego es necesario saber la dirección IP del *router*. Como este *router* viene por defecto configurado para otorgar direccionamiento dinámico mediante el

protocolo DHCP, usando la línea de comando de Windows se puede verificar que dirección IP tiene el *gateway* o el *router* en este caso de la siguiente marera.

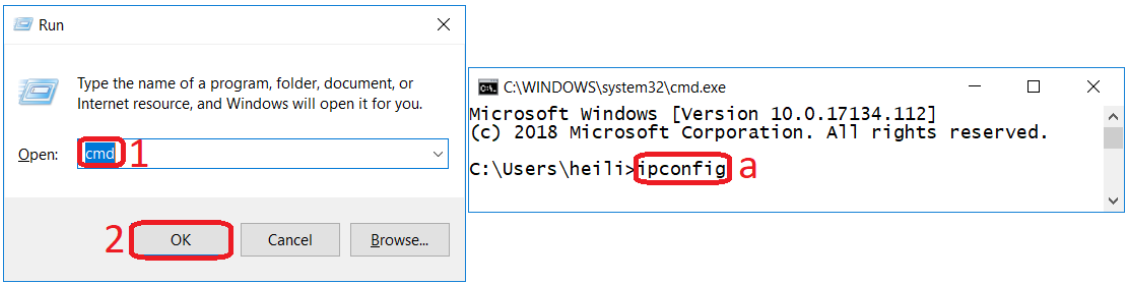


Figura 16: Abriendo terminal de Windows 10

La figura 16 indica los siguientes pasos

1) Oprimir de manera simultánea las teclas win y R. aparecerá una ventana llamada “Rum”. escribir cmd, para indicar que se requiere abrir la línea de comandos.
2) Oprimir la tecla “Enter” o clic en el botón “OK” y se abrirá una ventana como la que está a la derecha de la figura 16, esa es la línea de comando de Windows.
   a. Escribir el comando ipconfig y oprimir la techa “Enter” para ejecutarlo.

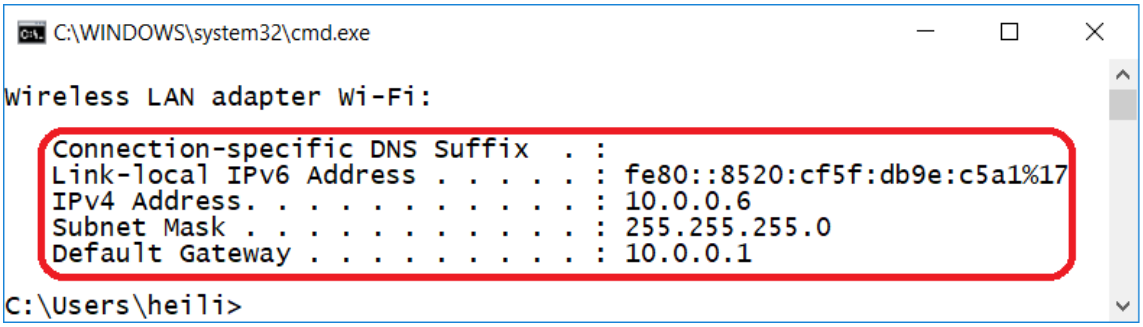


Figura 17: captura de pantalla de la terminal de Windows 10 ejecutando el comando ipconfig

En la figura 17 se puede ver el resultado del comando ipconfig. “IPv4 Address” indica la dirección asignada a la computadora, “Subnet Mask” indica la máscara de red. Con esa información se puede determinar el enrutamiento este hecho en la red 10.0.0.0/255.255.255.0; “Default Gateway” indica la dirección del router.

Basta con colocar la dirección del router es un explorador de internet para acceder las WEBUI del router. Al acceder a la WEBUI serán requeridos un usuario y una contraseña para acceder la configuración. Dichas credenciales son proveídas por el PSI a momento de la instalación del servicio.

Para configurar el direccionamiento estático con el protocolo DHCP se debe hacer el procedimiento indicado en la figura 18.

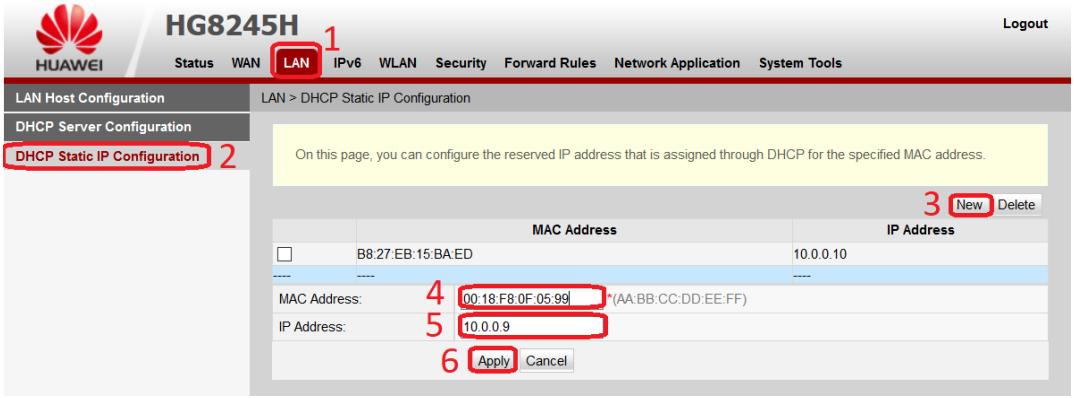


Figura 18: Captura de pantalla WEBUI del router HG8245H, seccion "DHCP Static IP Configuration".

1) Clic en la pestaña “Lan”.
2) Clic en la sección “DHCP stactic IP configuración”.
3) Clic en el botón new para agregar un *host* a la lista.

4) Especificar la dirección MAC del *host* que se requiere que tenga Dirección IP fija.
5) Especificar la IP destinada al *host*
6) Aplicar los cambios

Tanto como el ATA como el RPI se usa el mismo procedimiento.

2.3.2 Configuración de redireccionamiento de Puerto

Otra configuración necesaria en el *router* es el *port forwanding* o *port mapping* (redireccionamiento de puerto). El redireccionamiento de puerto es una aplicación del protocolo NAT que redirecciona la comunicación que recibe el *router* a un puerto especifico en la dirección IP pública del mismo, a un *host* en lado de la LAN, de esa manera es posible acceder a través de internet a los servicios de un *host* que este en una LAN.

Para este proyecto de investigación será necesario hacer redireccionamiento de Puerto en dos puertos. Uno para la VPN y otro para gestionaran la RPI por medio del protocolo SSH.

El redireccionamiento de Puerto se configura siguiendo los pasos de la figura 19 y 20.

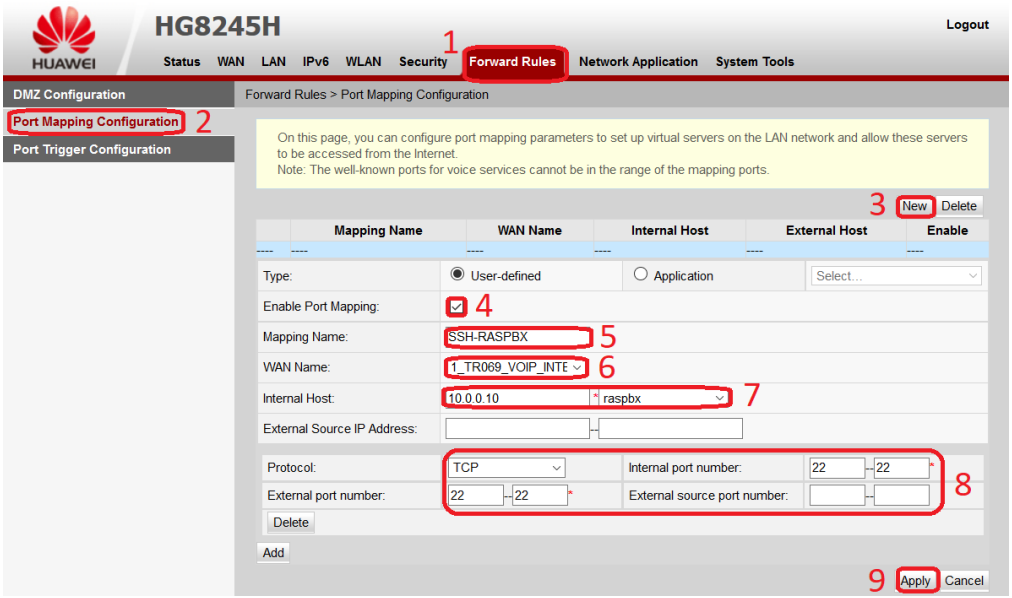


Figura 19: Captura de pantalla WEBUI del router HG8245H, sección "Port Mapping Configuration", configuración redireccionamiento puerto 22.

Paso para configuración del redireccionamiento puerto 22

1) Clic Pestaña “Foward Rules”
2) Clic en la Sección “Port Mapping Configuration”
3) Clic botón new para agregar nueva regla
4) Habilitar redireccionamiento de puerto
5) Agregar un nombre para regla
6) Elegir la WAN la cual servirá de entra para el redireccionamiento de puerto. en este paso el PSI tiene una WAN para cada servicio que ofrece IPTV, VoIP e Internet.
7) Especificar el *host* asociado con la regla, este caso la RPI siempre tendrá asignada la dirección IP 10.0.0.10
8) En este paso se especifica los puertos a redireccionar y protocolo con el que funcionara. En el campo “External port number” se especifica el puerto a redireccionar, en este caso el puerto 22 que con el que trabaja el protocolo SSH. En

el campo “Internal port number” específica a que puerto del *host* receptor será redireccionada la comunicación, también al puerto 22.

9) Aplicar los cambios

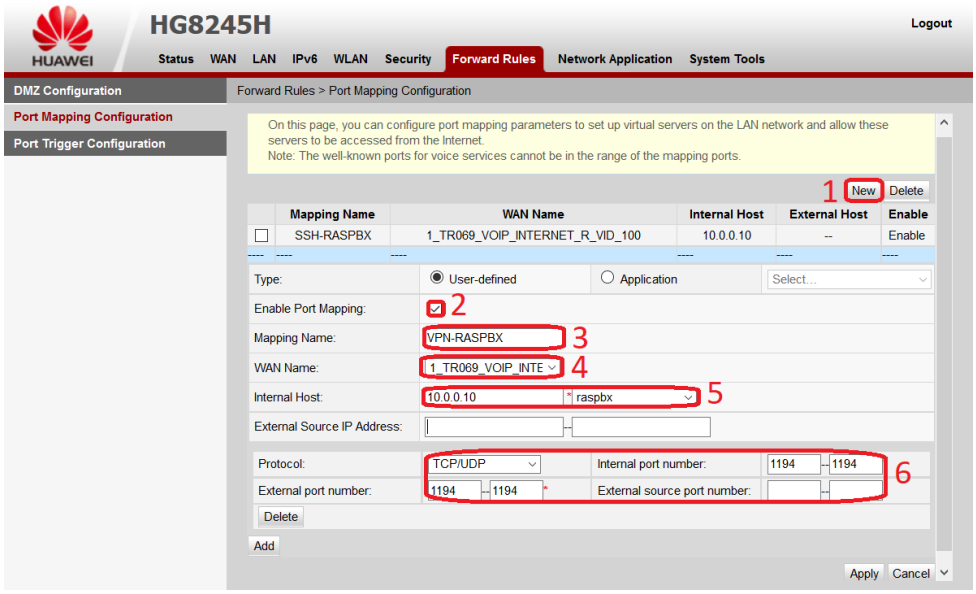


Figura 20: Captura de pantalla WEBUI del router HG8245H, sección "Port Mapping Configuration", configuración redireccionamiento puerto 1194.

Paso para configuración del redireccionamiento puerto 1194

1) Clic botón new para agregar nueva regla
2) Habilitar redireccionamiento de puerto
3) Agregar un nombre para regla
4) Elegir la WAN la cual servirá de entra para el redireccionamiento de puerto. en este paso el PSI tiene una WAN para cada servicio que ofrece IPTV, VoIP e Internet.
5) Especificar el *host* asociado con la regla, también será a la RPI

6) En este paso se especifica los puertos a redireccionar y protocolo con el que funcionara. En el campo “External port number” se especifica el puerto a redireccionar, en este caso el puerto 1194 que con el que trabaja el servicio OPENVPN. En el campo “Internal port number” específica a que puerto del *host* receptor será redireccionada la comunicación, también al puerto 1194.
7) Aplicar los cambios

2.3.3 Configuracion de DDNS

DNS es un protcolo que traduce de un nombre (*host*name o dominio) a una direcion IP o viceversa. Un ejemplo de uso de este protcolo es cuando acedemos a la una pagina web por ejemplo unapec.edu.do, en relidad lo que sucede que el explorado primero consultara a un servidor DNS a que direcion IP pertenece el *host*name eunepec.edu.do, luego este redidereciona al explorador a la direccion IP donde esta la pagina web que se requiere consultar.

El protcolo DDNS funciona de una manera similar pero con la diferencia de que este caso la direecion IP que esta asociada a un *host*ane no es fija, por lo tanto debe ser actualizada cada ves que el *host* dueño del *host*anme cambie de direccion IP.

Para acceser la sos servicion de la RPI se requiere hacer a traves de la Direccion IP publica del router, en este caso como el PSI la asigana dinamicamente lo que ocacionaria que cada ves que la direccion IP publica cambie los clientes que este fuera de la LAM donde esta la RPI pierda coneccion. Para resolver ese problema hay que registrar un serviocio de ddns para router .

2.3.3.1 Cuenta ddns

Existe diversos proverdores de DDNS, tanto gratis como de servicios pago. Para fines de este trabjajo de investigacion se eligio el provedor www.noip.com, este ofrece servio gratuito.

Figura 21: Captura de pantalla página web www.noip.com

Para crear una cuenta en noip.com se hace el procedimiento indicado en la figura 21

1. Espesificar el *host*name en este caso se asigno el nombre vozunapec
2. Se escge el dominio, en este caso se escogio hopto.org
3. Clic sigup
4. Ya realizado los pasos anteriores. Sera necesario llenar un formaulario con datos personales, para terminar el proceso de registro.

2.3.3.2 Configuracion de cuenta de DDNS en el *router*

Configurando la cuenta de DDNS en el *router*, este se encargará de asignar y actualizar la direccion IP pública que posea al *host* name creado previamente. Para ello se deben realizar los pasos de la figura 22

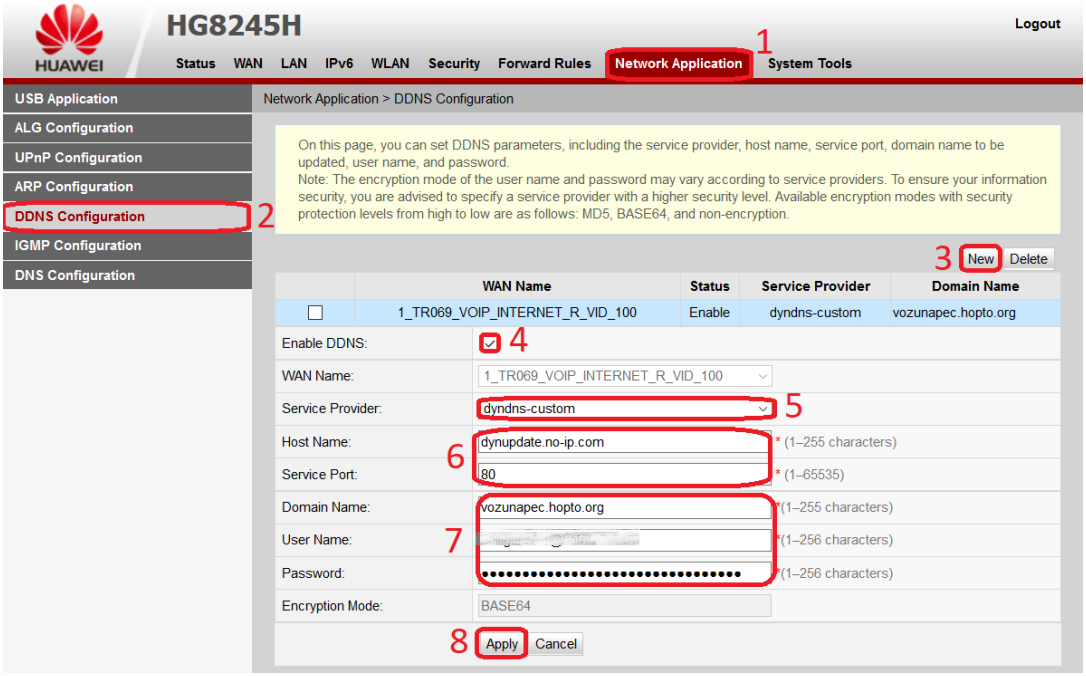


Figura 22: Captura de pantalla WEBUI del router HG8245H, sección "DDNS Configuration"

1) Clic en pestaña “Network Aplication”
2) Clic en sección “DDNS Configuration”
3) Clic en botón new para agregar una configuración de DDNS
4) Habilitar cuenta de DDNS
5) Elegir proveedor de DDNS, en este caso “dyndns-custon” porque este modelo de *router* contiene no-ip en su lista de selección

6) En el campo "*Host* Name" se escribe la dirección donde el *router* tiene que acceder para actualizar la dirección IP del dominio que deseamos. En el campo "Service Port" se coloca el puerto donde el proveedor de DDNS admite las solicitudes.
7) En este paso se agrega la información de la cuenta que el router utilizara para acceder a la cuenta en el proveedor de DDNS.
    a. En el campo "Domain Name" se coloca el dominio que queremos que el *router* actualice la dirección ip
    b. En los campos "User Name" y "Password" agregamos el usuario y la clave de la cuenta de noip.com
8) Aplicar los cambios

2.4 Configuración inicial Raspberry Pi

En RPI funcionara como servidor de varios servicios como son: servidor VPN y VoIP con Asterisk. Para acceder a la RPI primero se debe conectar la RPI a uno de los puertos ethernet del *router* y conectar la fuente de alimentación al puerto mini USB. Luego de unos segundos podemos usar el protocolo SSH para acceder la consola del RPI. En este caso se utilizó el software PuTTY.

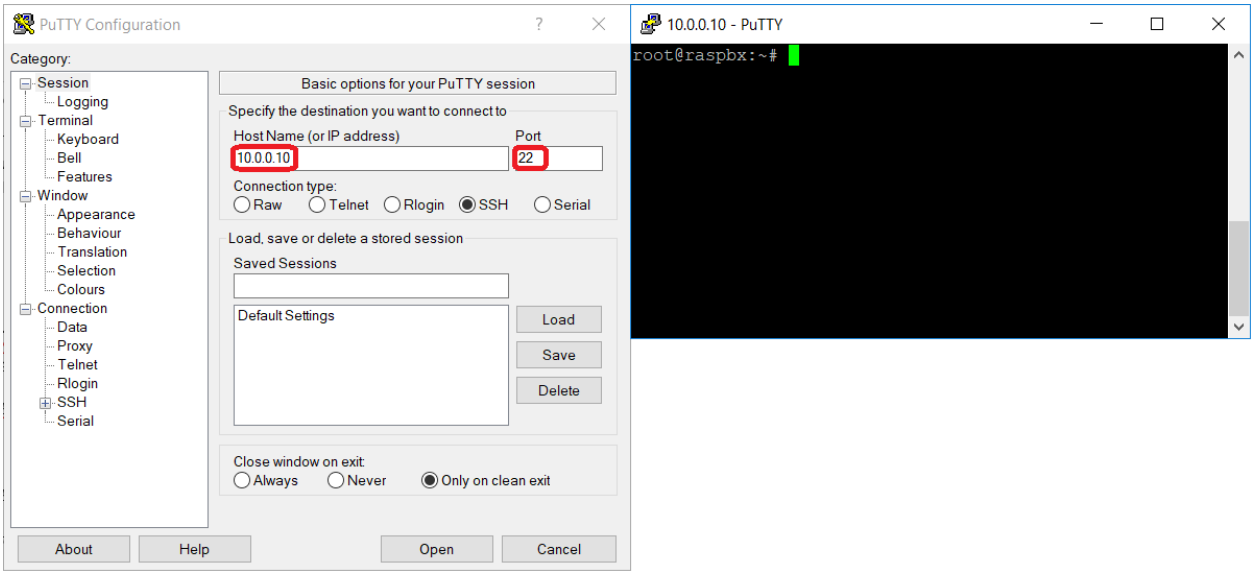


Figura 23: Captura de pantalla del software PuTTY

Al ejecutar putty, primero se debe especificar la dirección IP del RPI sabemos que es la 10.0.0.10 porque se configuro el router para que entregue esa dirección IP, el protocolo SSH por defecto trabaja en el puerto 22. Al oprimir la tecla “Enter” o clic el botón “Open” se abrirá una venta como se ven en lado derecho de la figura 23.

Para mostrar de una manera más clara las configuraciones en hecha en la consola de comando y resultado de comandos ejecutados se utilizará el siguiente formato.

```
root@raspbx:~# commando ejecutado

resultado del comando ejecutado resultado del comando ejecutado
resultado del comando ejecutado resultado del comando ejecutado
en color rojo notas y entradas de parámetro
root@raspbx:~#
```

Al acceder a la consola del RPI, se solicitará el usuario y la contraseña para acceder. Por defecto Raspbx tiene el usuario “root” y la contraseña “raspberry”, luego introducida dicha credenciales aparecerá un mensaje de bienvenida en la consola de comandos como se ve a continuación:

```
login as: root
root@10.0.0.10's password:
Linux raspbx 4.14.34-v7+ #1110 SMP Mon Apr 16 15:18:51 BST 2018 armv7l
Welcome to RasPBX - Asterisk for Raspberry Pi

RasPBX is based on Debian. The programs included with the Debian GNU/Linux
system are free software; the exact distribution terms for each program are
described in the individual files in /usr/share/doc/*/copyright.

RasPBX comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

List of RasPBX specific commands:
--------------------------------------------------------------------------------
raspbx-upgrade      Keep your system up to date with the latest add-ons and
                    security fixes
configure-timezone  Set timezone for both system and PHP
install-fax         Install HylaFAX
add-fax-extension   Add additional fax extension for use with HylaFAX
install-fail2ban    Install Fail2Ban for additional security
install-dongle      Install GSM/3G calling capability with chan_dongle
raspbx-backup       Backup your complete system to an image file

root@raspbx:~#
```

2.4.1 Cambio de contraseña del usuario “root”

Es muy importante cambiar la clave del usuario “root” que trae por defecto por otra más segura ya que con la configuración echa previamente en el *router*, la RPI está expuesta a que cualquiera pueda acceder a ella desde el internet. Esto se hace ejecutado el comando “passwd”.

```
root@raspbx:~# passwd

Enter new UNIX password: introducir contraseña nueva

Retype new UNIX password: confirmar contraseña nueva

passwd: password updated successfully

root@raspbx:~#
```

2.4.2 Configuración de zona horaria

Es conveniente que de la zona horaria de Raspbx esté correcta, ya que los registros que realice el sistema estarán debidamente ubicados en el tiempo, además el sistema puede ser programado para hacer tareas a una hora y fecha indicada. Con el comando “configure-timezone”. Al ejecutar el comando aparecerá una aplicación grafica en la consola de comando como se ven en la figura 24. Con las flechas del teclado debemos elegir la opción “America” presionar “Enter” en la siguiente ventana seleccionar la opción Santo Domingo y presionar la tecla nuevamente.

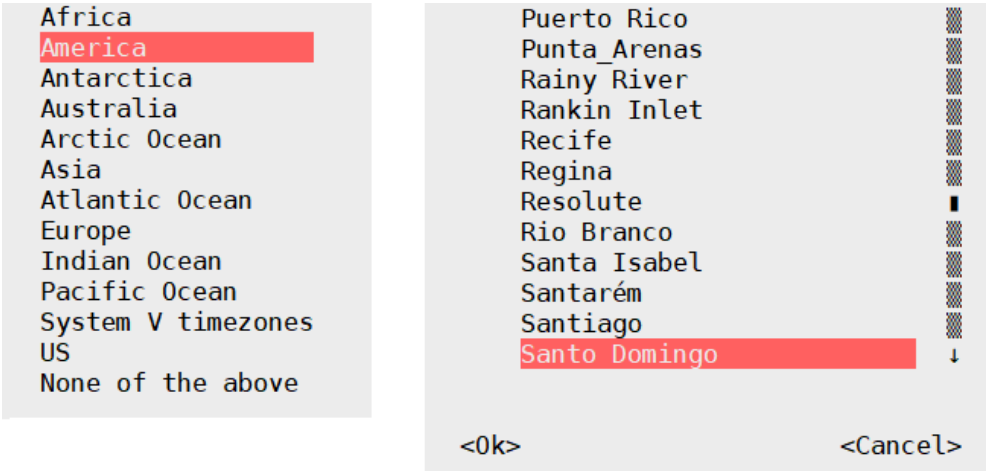


Figura 24: Captura de pantalla consola de RPI, salida del comando configure-timezone

Al finalizar la configuración horaria, la consola tendrá la siguiente salida:

```
root@raspbx:~# configure-timezone

Current default time zone: 'America/Santo_Domingo'

Local time is now:      Mon Jun 25 18:23:02 AST 2018.

Universal Time is now:  Mon Jun 25 22:23:02 UTC 2018.

Setting PHP to system timezone: America/Santo_Domingo

root@raspbx:~#
```

2.4.3 Expansión del sistema de archivo

Cuando se graba el sistema operativo en la memoria SD del RPI gran parte de la capacidad de esta que inhabilitada para su uso. Para dar reverso a esa condición se ejecuta el comando “raspi-config”; al ejecutar este comando aparecerá una aplicación gráfica con un menú de opciones como el que se ve en la figura 25. Con las flechas del teclado primero se debe seleccionar la opción “7 Advanced Options” y presionar la tecla “Enter” luego aparecerá otra ventana, ahora se debe seleccionar la opción “A1 Expand Filesystem”.

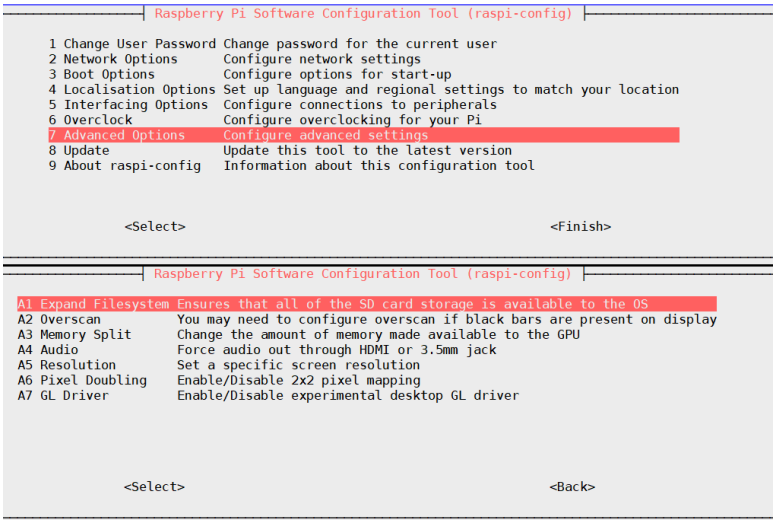


Figura 25: Captura de pantalla consola de RPI, salida del comando raspi-config

Después de realizar la expiación del sistema de archivo, la aplicación mostrará un mensaje como el que se ve a la izquierda de la figura 26, indicando que los cambios serán efectuados luego de reiniciar el sistema. Al presionar la tecla “Enter” la aplicación volverá a presentar el menú inicial, en esta ocasión se seleccionará la opción “Finish” y presionar la tecla “Enter”.

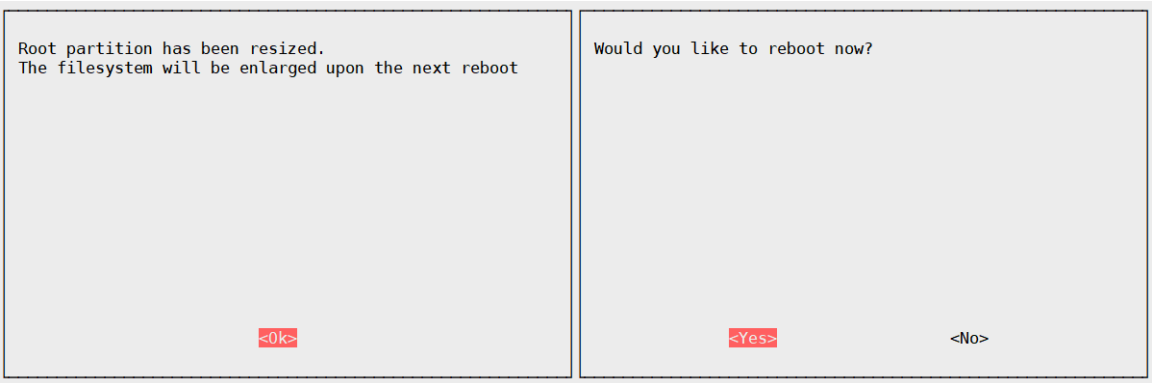


Figura 26: Captura de pantalla consola de RPI salida del comando raspi-config.

Antes de la aplicación cerrarse preguntará si se desea reiniciar el sistema ahora, seleccionar la opción “Yes” y presionar la “Enter”. La RPI se reiniciará y después de unos segundos ya estará lista para uso nueva vez.

2.5 Instalación del GSM Gateway

Para este trabajo de investigación se usará un módem modelo K3765 de la marca Huawei. Este comercialmente se vende como un módem USB para para usar servicios de internet vía GSM. Raspbx permite utilizar este dispositivo como GSM Gateway. Solo se requiere ejecutar el comando install-dongle y como resultado este configurará Asterisk para que este se asocie con el módem.

Figura 27: Imagen módem Huawei K3765, Obtenida de un anuncio online.

Cuando de ejecuta el comando install-dongle este solicita una serie para parámetros para completar correctamente la ejecución. A continuación, se muestra el proceso en la consola.

```
root@raspbx:~# install-dongle
Installing GSM VoIP gateway with chan_dongle.

Aquí solicita el teléfono móvil que le pertenece al módem

Please enter the phone number of your SIM card

Aquí solicita un teléfono móvil donde redirigir los mensajes SMS que llegan al
módem.

(defaults to +1234567890 if left blank): 8x9xxxxxxx

Send incoming SMS to email address

Aquí solicita un correo electrónico móvil donde redirigir los mensajes SMS que
llegan al módem.

(leave empty to disable SMS forwarding): ********.****@gmail.com

Forward incoming SMS to mobile phone number (via dongle0)

(leave empty to disable): 8296991504

Saving previous contents of /etc/asterisk/extensions_custom.conf to
/etc/asterisk/extensions_custom.conf.orig

Aquí pregunta si se desea instalar una interface web para enviar mensajes SMS "Y"
para sí "N" para no

Would you like to install a webpage for sending SMS with

chan_dongle? (http://raspbx/sms/) [y/N] y

si la repuesta es y, el acceso a la interface será en la dirección web
http://raspbx/sms/ y se debe especificar una contraseña para uso de la interface
para enviar SMS

Enter password for SMS page: **********

Your configuration has been saved successfully to these files:

/etc/asterisk/dongle.conf

root@raspbx:~#
```

2.6 Instalación de OpenVPN

Raspbx cuenta con un amplio repositorio de aplicaciones, con solo un comando se puede instalar cualquier aplicación que se encuentre dentro del repositorio de Raspbx. La instalación de OpenVPN se hace con la ejecución del comando apt-get install OpenVPN -y. A continuación, se muestra el resultado en consola de la ejecución del comando para instalar OpenVPN.

```
root@raspbx:~# apt-get install openvpn -y
Reading package lists... Done
The following NEW packages will be installed:
  openvpn
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/440 kB of archives.
After this operation, 1,048 kB of additional disk space will be used.
Preconfiguring packages ...
Selecting previously unselected package openvpn.
(Reading database ... 51690 files and directories currently installed.)
Preparing to unpack .../openvpn_2.4.0-6+deb9u2_armhf.deb ...
Unpacking openvpn (2.4.0-6+deb9u2) ...
Setting up openvpn (2.4.0-6+deb9u2) ...
Processing triggers for systemd (232-25+deb9u2) ...
Processing triggers for man-db (2.7.6.1-2) ...
root@raspbx:~#
```

2.7 Instalación de mail

Otra aplicación que será útil para el sistema es la aplicación mail. Esta permitirá enviar correos electrónicos con notificaciones. Se instala con el comando apt-get install mailutils -y.

```
root@raspbx:~# apt-get install mailutils -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  mailutils-mh mailutils-doc
The following NEW packages will be installed:
  mailutils
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 552 kB of archives.
After this operation, 949 kB of additional disk space will be used.
Get:1 http://mirror.us.leaseweb.net/raspbian/raspbian stretch/main armhf
mailutils armhf 1:3.1.1-1 [552 kB]
Fetched 552 kB in 14s (38.8 kB/s)
...
...
root@raspbx:~#
```

2.8 Configuración de OpenVPN

El primer paso es generar el certificado de autenticación (CA) certificados para los clientes y llaves. OpenVPN incluye una aplicación llamada Easy-rsa, esta aplicación se usa en la línea de comando y sirve para la gestión y creación de los certificados de autenticación. Está disponible para Windows y Linux. En esta ocasión se utilizó el Easy-rsa para Windows, siguiendo el siguiente procedimiento.

Todos los scripts y archivos par el uso de Easy-rsa se encuentran en el directorio C:\Program Files\OpenVPN\easy-rsa, para acceder ahí basta con solo escribir la dirección mencionada en el explorado de Windows. una vez ahí se deben realizar los siguientes pasos.

1) Editar el archivo vars.bat, este archivo contiene todas las variables a utilizar para la creación los certificados. Para editar este archivo basta con hacer clic derecho sobre el archivo y clic en la opción editar.se Abrirá algún editor de texto comúnmente Notepad. De este archivo solo se debe modificar las siguientes líneas.

```
set KEY_DIR=keys directorio donde se grabarán los certificados y llaves
set DH_KEY_SIZE=1024 tamaño en bits de las llaves
set KEY_SIZE=4096
set KEY_COUNTRY=RD     datos del propietario del certificado
set KEY_PROVINCE=SD
set KEY_CITY=SantoDomingo
set KEY_ORG=Unapec
set KEY_EMAIL=servidorrasp.alertas@gmail.com
set KEY_CN=changeme
set KEY_NAME=Unapec
set KEY_OU=Electronica
set PKCS11_MODULE_PATH=changeme
set PKCS11_PIN=1234
```

2) Una vez establecidas las variables, se debe ejecutar el script vars.bat para que Easy-rsa realice los cambios en las variables y el script clean-all.bat este creara la carpeta “keys” en el directorio donde esta Easy-rca y copiaraá otros archivos necesarios para la creación de los certificados y llaves. Para poder ejecutar los scripts se debe hacer desde la consola de Windows de la siguiente manera.

```
C:\WINDOWS\system32>cd C:\Program Files\OpenVPN\easy-rsa cambia al directorio de Easy-rca.
C:\Program Files\OpenVPN\easy-rsa>vars.bat ejecuta el script vars
C:\Program Files\OpenVPN\easy-rsa>clean-all.bat ejecuta el script clean-all
        1 file(s) copied.
        1 file(s) copied.
```

3) luego de preparar el camino con los pasos anteriores, ya es posible comenzar a generar el certificado de autoridad (CA). Este será usado para firmar los certificados de los clientes, además este archivo debe permanecer solo en manos del administrador. El ca genera con el script buil-ca.

```
C:\Program Files\OpenVPN\easy-rsa>build-ca.bat
Generating a 4096 bit RSA private key
.....................................++
writing new private key to 'keys\ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
----- solicita que le confirme las variables que ya predefinimos en var
```

```
Country Name (2 letter code) [RD]: solo se debe oprimir la tecla enter
State or Province Name (full name) [SD]:
Locality Name (eg, city) [SantoDomingo]:
Organization Name (eg, company) [Unapec]:
Organizational Unit Name (eg, section) [Electronica]:
Common Name (eg, your name or your server's hostname) [changeme]:raspbx  este parámetro es el único que debe ser diferente en todos los archivos
Name [Unapec]:
Email Address [servidorrasp.alertas@gmail.com]:
```

4) con el CA generado, ya será posible las llaves del servidor y de los clientes. Para generar la llave del servidor se ejecuta el script "buil-key-server nombre_archivo" donde el nombre_archivo es el nombre que asignar al archivo de la llave que va a generar el script.

```
C:\ProgramFiles\OpenVPN\easy-rsa>build-key-server.bat VPN_Server
Generating a 4096 bit RSA private key
...
...
Country Name (2 letter code) [RD]:
State or Province Name (full name) [SD]:
Locality Name (eg, city) [SantoDomingo]:
Organization Name (eg, company) [Unapec]:
Organizational Unit Name (eg, section) [Electronica]:
Common Name (eg, your name or your server's hostname) [changeme]:vpn_raspbx
Name [Unapec]:
Email Address [servidorrasp.alertas@gmail.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:***************
An optional company name []:
...
...
Certificate is to be certified until Jun 24 03:16:56 2028 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

5) para generar las llaves de los cliente el proceso es similar al de generar las llaves del servidor. Para cada cliente se debe de generar una llave.

```
C:\Program Files\OpenVPN\easy-rsa>build-key.bat VPN_PCJ
...
...
Country Name (2 letter code) [RD]:
State or Province Name (full name) [SD]:
Locality Name (eg, city) [SantoDomingo]:
Organization Name (eg, company) [Unapec]:
Organizational Unit Name (eg, section) [Electronica]:
Common Name (eg, your name or your server's hostname) [changeme]:vpn_pcj
Name [Unapec]:
Email Address [servidorrasp.alertas@gmail.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:MoEa0PZAB5joOcF5Kem2
An optional company name []:
...
...
Certificate is to be certified until Jun 24 03:30:23 2028 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

C:\Program Files\OpenVPN\easy-rsa>
```

6) ya por último se debe de generar el archivo Diffie–Hellmanuild-sh.bat.

```
C:\Program Files\OpenVPN\easy-rsa>build-dh.bat
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time

.........+.....................................................................
.....................+.........................................................
.....................+.........................................................
.............................+.................................................
.....................................................................+.....+
...+.+.....................................................................
.....................................................................
.+..................................

C:\Program Files\OpenVPN\easy-rsa>
```

La ejecución de este script puede demorar un largo tiempo.

Después de realizar los pasos anteriores, en el directorio keys tendremos el siguiente listado de archivos.

Nombre	***Necesitado por***	***propósito***	***privado***
ca.crt	*Todos los clientes*	*Certificado CA*	*no*
ca.key	*administrador*	*Llave CA*	*si*
Dh1024.pem	*servidor*	*Diffie Hellman*	*no*
VPN_server.crt	*servidor*	*Certificado servidor*	*no*
VPN_server.key	*servidor*	*Llave servidor*	*si*
VPN_PCJ.crt	*Laptop de prueba*	*Certificado celular*	*no*
VPN_PCJ.key	*Laptop de prueba*	*Llave lapto*	*si*
J_cel.crt	*Celular de prueba*	*Certificado celular*	*no*
Jcel.key	*Celular de prueba*	*Llave celular*	*si*

7) Ya generado todos los archivos necesarios para la VPN, entonces se debe distribuir los archivos a los *host*s correspondiente. En la laptop se debe de copiar los archivos en el directorio `C:\Program Files\OpenVPN\config\keys\`, en el celular el contenido de los archivos se copia en el archivo de configuración y en el caso de la RPI con una aplicación llamada Winscp, esta es cliente sftp (safe File Tranfer Protocol / protocolo de trasferencia de archivos seguro) que sirve para transferir archivos a través de la red.

Al ejecutar la aplicación aparecerá una ventana como se ve en la figura 28. En ella se debe especificar la dirección IP del *host* a que se conectará, sftp trabaja en el puerto 22, las credenciales. Ya después de llenar los campos mencionados sigue hacer clic en botón “Login”

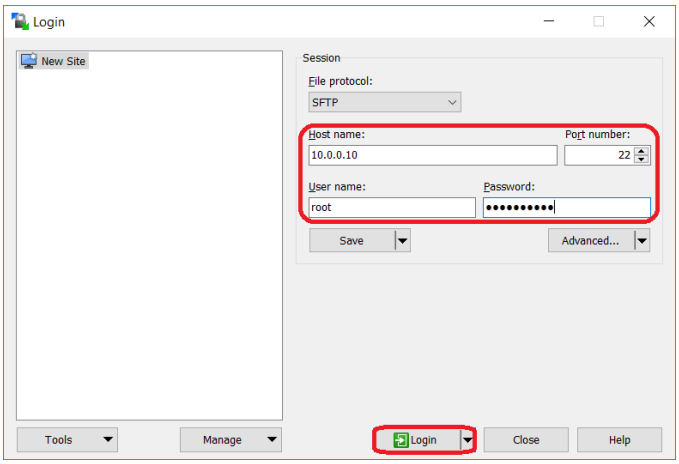


Figura: 28: Captura de pantalla software winscp, inicio de sección

Luego aparecerá una ventana como en la figura 29 del lado izquierdo se debe buscar la dirección donde está la dirección donde se generó los archivos. Del lado derecho se buscar el directorio etc/openvpn/unapec/keys. Por último, se debe de arrastrar los archivos necesarios del lado izquierdo al derecho y los archivos se transferirán.

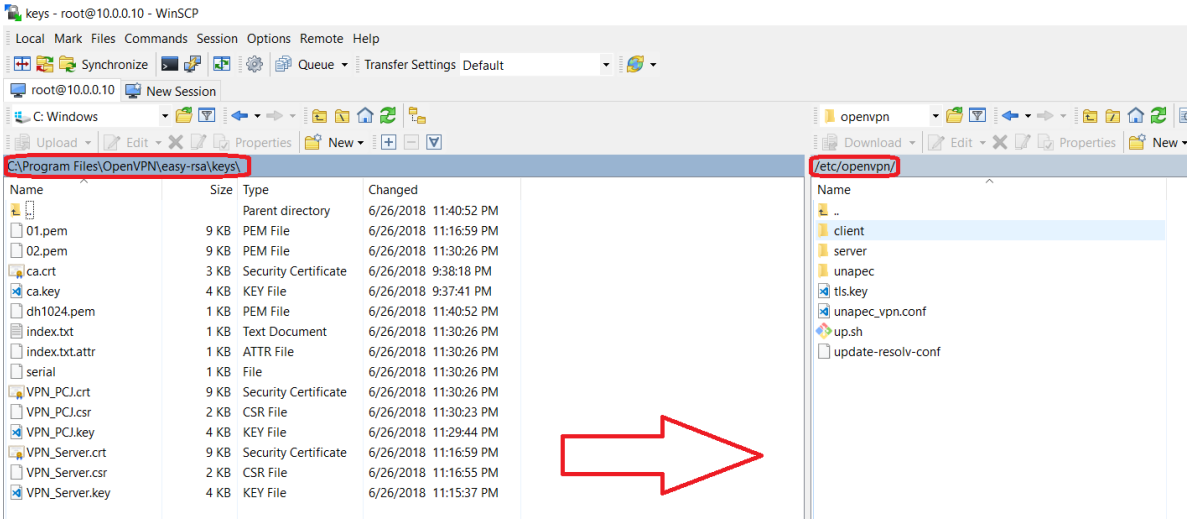


Figura 29: Captura de pantalla software winscp, traspaso de archivos

8) Ya con los certificados y llaves es su lugar ahora es tiempo de escribir los archivos de configuración.

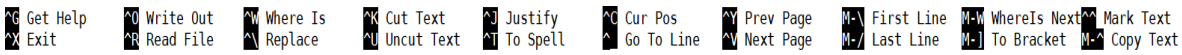
El archivo de configuración del RPI debe de estar en el directorio /etc/openvpn, este archivo configurara el RPI como un servidor VPN. Para crear un archivo de configuración se hace el siguiente procedimiento en las consolas del RPI.

```
root@raspbx:~# cd /etc/openvpn cambio de directorio

root@raspbx:/etc/openvpn# nano unapec_vpn.conf comando para abrir editor de texto nano y crear el archivo unapec_VPN.conf
```

La escritura del archivo de configuración consiste en escribir cada una de las opciones que sean necesaria para lograr la configuración deseada. El listado de opciones que tiene despoblé OpenVPN se pueden encontrar en [45] [44] [49].

```
float
proto tcp
port 1194
dev tunVPN
script-security 2
up "up.sh up"
up-restart
plugin /usr/lib/openvpn/openvpn-plugin-auth-pam.so login
push "route 10.0.0.1 255.255.255.255"
push "route 10.0.0.0 255.255.255.0"
topology subnet
server  192.168.200.0 255.255.255.0
tls-server
ca      /etc/openvpn/unapec/keys/ca.crt
cert    /etc/openvpn/unapec/keys/VPN_Server.crt
key     /etc/openvpn/unapec/keys/VPN_Server.key
dh      /etc/openvpn/unapec/keys/dh1024.pem
verb 3
log-append /var/log/openvpn.log
status /var/log/status.log 5
down "up.sh down"
down-pre
```



Donde las funciones de las opciones son la siguientes:

float: Permite que OpenVPN pueda trabajar con varios clientes a la vez.

proto: Especifica el protocolo de transporte que usara la red VPN, puede ser UDP o TCP.

port: Especifica el puerto donde trabajar la VPN.

dev: Especifica el tipo de interfaz virtual a utilizar y el nombre que tendrá.

script-security: Habilita la posibilidad que OpenVPN ejecute scripts.

up: Ejecuta un comando al momento que OpenVPN inicie. En este caso ejecuta un script para enviar una notificación por correo.

up-restart: Ejecuta un comando al momento después que OpenVPN es reiniciado. En este caso ejecuta un script para enviar una notificación por correo**.**

plugin: Sirve para agregar módulos a OpenVPN. En este caso agrega el módulo openvpn-plugin-auth-pam.so que es utilizado para que los clientes acedos con las credenciales de los usuarios registrado en el RPI como el usuario root.

push: Esta opción limita que solo el tráfico de red 10.0.0.0/255.255.255.0 sea el que pase a tres de la VPN. Así solo para el tráfico de servicios de VoIP a través de la VPN y no el tráfico que generen todos los usuarios.

topology: Configura el direccionamiento para las interfaces virtuales**.**

server: Especifica el direccionamiento a utilizar en la VPN.

tls-server: habilita el uso de certificados en la VPN**.**

ca: indica el certifica de autoridad que usara el servidor**.**

cert: Indica el certificado que usara el servidor.

key: Indica la llave que usara el servidor**.**

dh: Indica el archivo con los parámetros Diffie Hellman a utilizar.

verb: Indica el grado de sensibilidad con que se guardaran los registros de la VPN, donde 1 es el mínimo 5 en el máximo

log-append: Indica el archive donde se guardará los registros de la VPN**.**

status: Registra estados de las secciones en la VPN

down: Ejecuta un comando al momento que OpenVPN sea detenido. En este caso ejecuta un script para enviar una notificación por correo.

down-pre: Ejecuta un comando al momento que OpenVPN es reiniciado. En este caso ejecuta un script para enviar una notificación por correo.

El archivo de configuración para un cliente en una Computadora o laptop puede ser escrito en cualquier editor.

```
client
proto tcp
remote vozunapec.hopto.org
port 1194
dev tun
tls-client
auth-user-pass
tls-auth "C:\\Program Files\\OpenVPN\\config\\tls.key"
nobind
ca "C:\\Program Files\\OpenVPN\\config\\keys\\ca.crt"
cert "C:\\Program Files\\OpenVPN\\config\\keys\\VPN_PCJ.crt"
key "C:\\Program Files\\OpenVPN\\config\\keys\\VPN_PCJ.key"
```

client: Indica que se una cliente.

remote: Indica la dirección el servidor VPN a conectarse**.**

tls-client: Habilita el uso del sistema PKI.

auth-user-pass: pregunta al usuario por sus credenciales de acceso.

El archivo de configuración para un celular debe ser escrito en el siguiente formato.

```
client
proto udp
remote vozunapec.hopto.org
port 1194
dev tun
nobind

<ca>
-----BEGIN CERTIFICATE-----
Copiar contenido del archivo ca
-----END CERTIFICATE-----
</ca>

<cert>
-----BEGIN CERTIFICATE-----
Copiar contenido del archivo j_cel.crt
-----END CERTIFICATE-----
</cert>

<key>
-----BEGIN PRIVATE KEY-----
Copiar contenido del archivo j_cel.key
-----END PRIVATE KEY-----
</key>

<tls-auth>
-----BEGIN OpenVPN Static key V1-----
-----END OpenVPN Static key V1-----
</tls-auth>
```

2.9 Configuración de IPtables

Es necesario configurar el firewall de Raspbx para que nos permita acceder a los demás *host* de la LAN a través de la VPN. Para dicho objetivo se debe de modificar la línea net.ipv4.ip_forward = 0 del archivo /etc/sysctl.conf por net.ipv4.ip_forward = 1. Luego ejecutar los siguientes comandos.

```
root@raspbx:~# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.10

root@raspbx:~# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERAD
```

2.9.1 Configuración de mail

Para la configuración mail debemos especificar en el archivo en el directorio /etc/ssmtp/ssmtp.conf, que email será utilizado por el servidor para enviar las notificaciones del sistema. En la siguiente consola muestra los escritos en el archivo de configuración.

```
mailhub=smtp.gmail.com:587 (servidor de Gmail)
AuthUser=servidorrasp.alertas@gmail.com (email que utilizara el servidor)
AuthPass=******** (clave de acceso al correo)
UseTLS=YES
UseSTARTTLS=YES
```

2.10 Configuración de Asterisk

Ahora se mostrará el proceso de configuración de la central VoIP.

2.10.1 Configuración inicial

Como primer paso se debe acceder desde cualquier explorador web a la dirección IP del RPI. Cuando se intente acceder a la WEBUI aparecerá una página web como la que se ve en la figura 30. Se debe de llenar el formulario para crear una cuenta de administrador, además de especificar un email donde se recibirán notificaciones de la central.

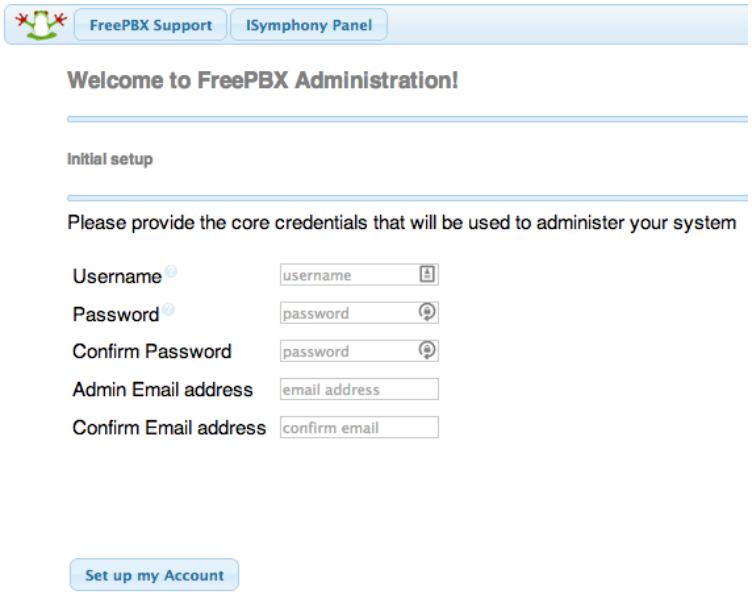


Figura 30: Captura de pantalla de WEBUI FreePBX de Asterisk

Luego de llenar el formulario de la figura 30 se debe hacer clic el botón (“Set up my Account”). Luego aparecerá una página web como a que se en la figura 31, haciendo clic en el ícono (“FreePBX Administration”) y luego introduciendo las credenciales especificadas

en el formulario anterior podremos acceder a la web de configuración de Asterisk.

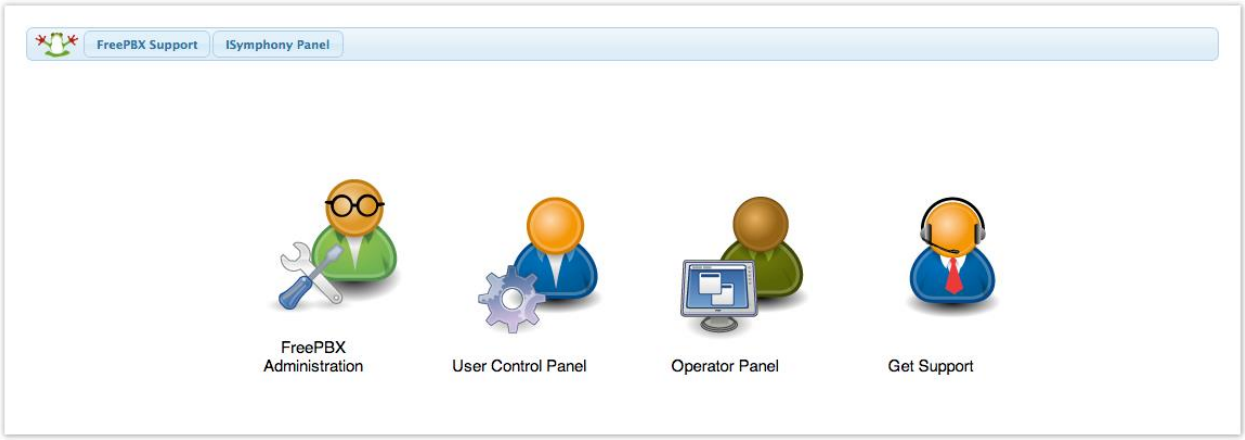


Figura 31: Captura de pantalla de WEBUI FreePBX de Asterisk, inicio de sección

Al acceder a Asterisk la primera página que aparecerá es la del *dashboard* (panel de control) en esta se podrán ver información del estado del sistema y alguna estadística del uso del sistema. En la figura 32 se muestra una vista del *dashboard*, y se desglosan sus partes.

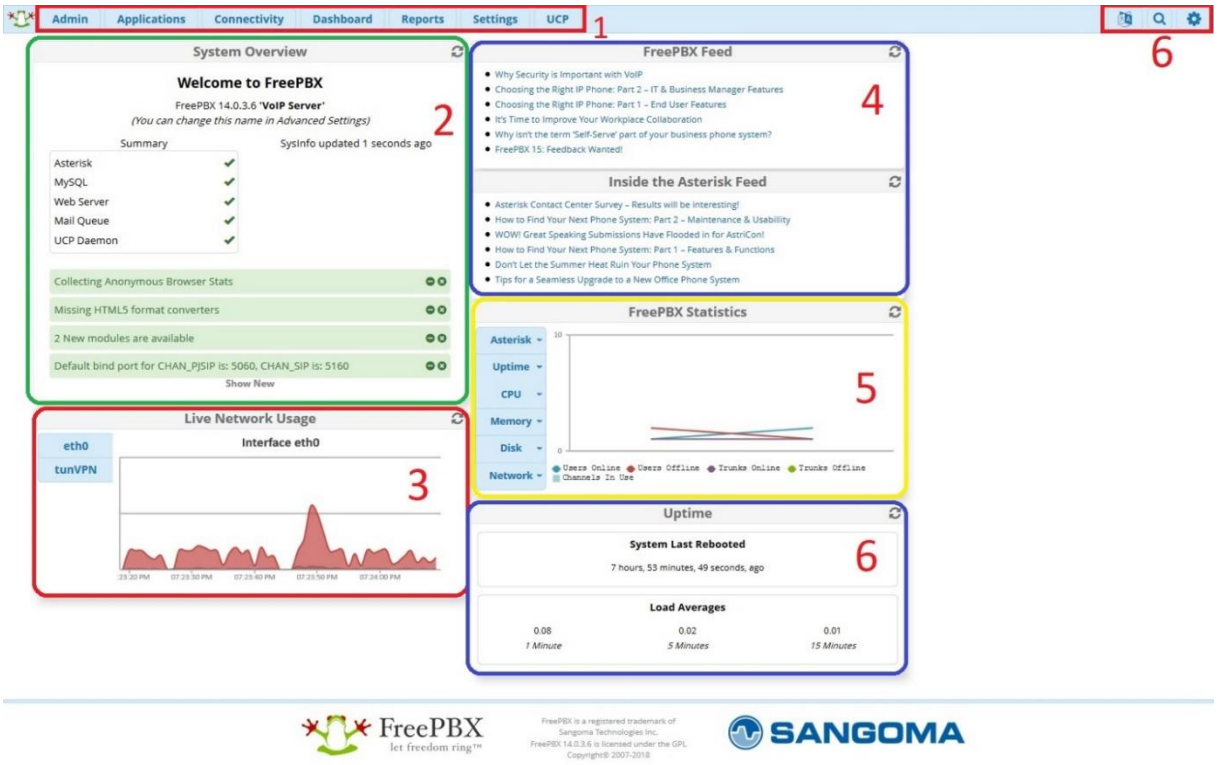


Figura 32: Captura de pantalla del dashboard de FrePBX

1) En esta parte están las pestañas para acceder a las configuraciones de Asterisk.
2) Muestra el estado de los servicios ofrecido y utilizado por Asterisk.
3) Muestra una gráfica del flujo que pasa a través de la interface de red.
4) Muestra noticias relevantes a Asterisk como actualizaciones y nuevas opciones.
5) Muestra una estadística del uso de recurso de Asterisk.
6) Muestra estadísticas de promedio de llamadas y tiempo de encendido del sistema.

2.10.2 Creación de extensiones

Las extenciones son como números telefonicos que se asiganan a los telefonos IP o *softphone* en el sistema. Se crean con el siguiente procedimiento:

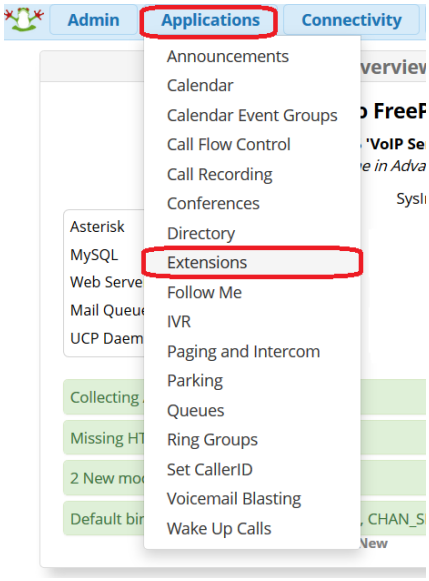


Figura 33: Captura de pantalla de FreePBX, pestaña aplicaciones

1) Clic en la pestaña de aplicaciones y luego clic en la en la opción de extensiones

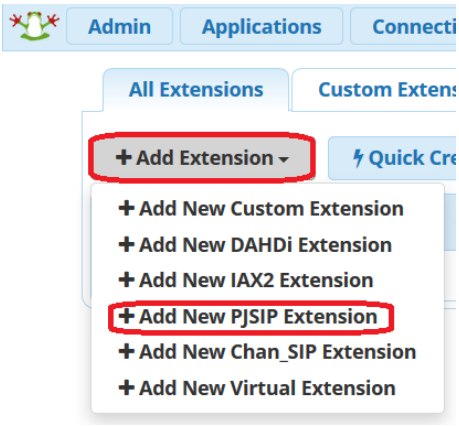


Figura 34: captura de pantalla de FreePBX, ventana extensiones

2) Después, en la ventana de extensiones, para crear una nueva extensión primero se debe hacer clic en el botón "+ Add Extension" luego en la opción "+ Add New Pjsip Extension"

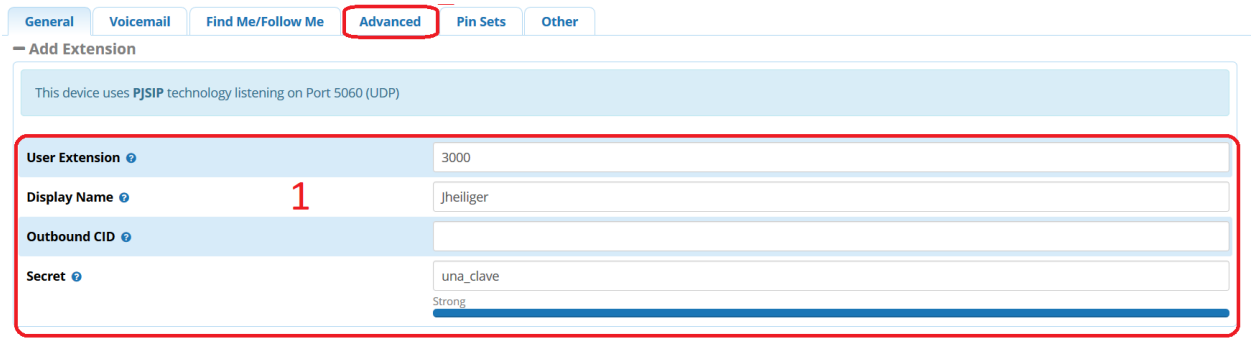


Figura 35: Captura de pantalla de FreePBX, ventana extensiones

3) Como siguiente paso se debe llenar un formulario de la figura 35, con los datos de la extensión a crear.

Donde:

- ***User Extension*:** el número que se le asignará a la extensión.
- ***Display Name*:** el nombre que aparecerá en pantalla cuando realice una llamada.
- ***Outboun CID*:** este campo no es necesario llenarlo.
- ***Secret*:** aquí podremos la clave con la que el *softphone* o teléfono IP se asociará al sistema .

4) luego debemos especificar otros parámetros, clic en la pestaña “Advanced”: en esta pestaña hay una gran cantidad de opciones que podemos usar, pero para los fines de este trabajo de investigación solo nos interesan las opciones de las figuras 36 y 37.

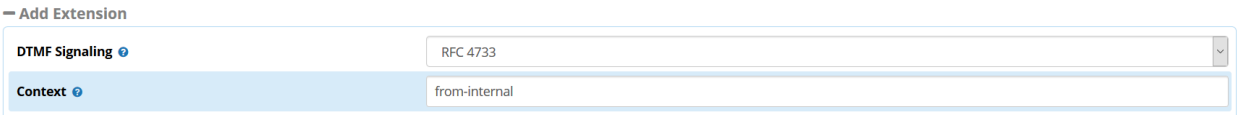


Figura 36: Captura de pantalla de FreePBX, ventana extensiones

- **En DTMF Signaling;** RFC4733 este es el estándar de señalización de las teclas de teléfono.

Figura 37: Captura de pantalla de FreePBX, ventana extensiones

En las opciones de “recording” para fines de este trabajo de investigación serán marcadas todas en “yes” porque así se graban todas las llamadas de las extensiones.

2.10.3 Configuración de colas

Las colas permiten tratar grupos de extensiones como una sola extensión. Para la configuración de las colas se hace el siguiente procedimiento.

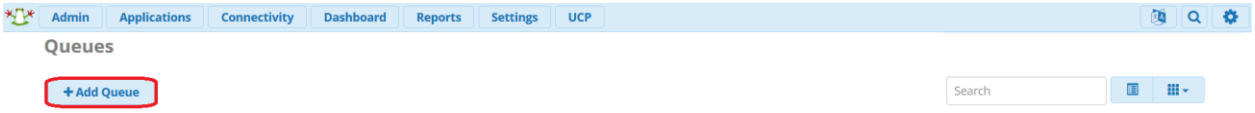


Figura 38: captura de pantalla de FreePBX, ventana de configuración de colas.

1) Clic en el botón “+Add Queue”

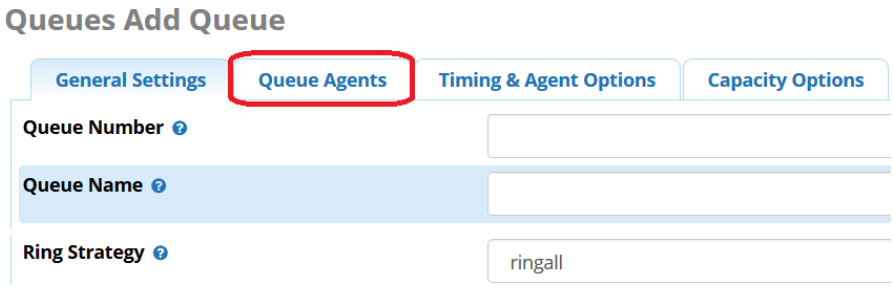


Figura 39: Captura de pantalla de FreePBX, ventana de configuración de colas.

2) Se llena el formulario de la figura 39. La opción "Queue Number" es un identificador numérico para las colas. "Queue Name" asigna un nombre. La opción "Ring strategy" indica la acción a realizar cuando se dirija una llamada a la cola, "ringall" sonarán todas las extensiones y la primera que se conteste se quedará con la llamada.

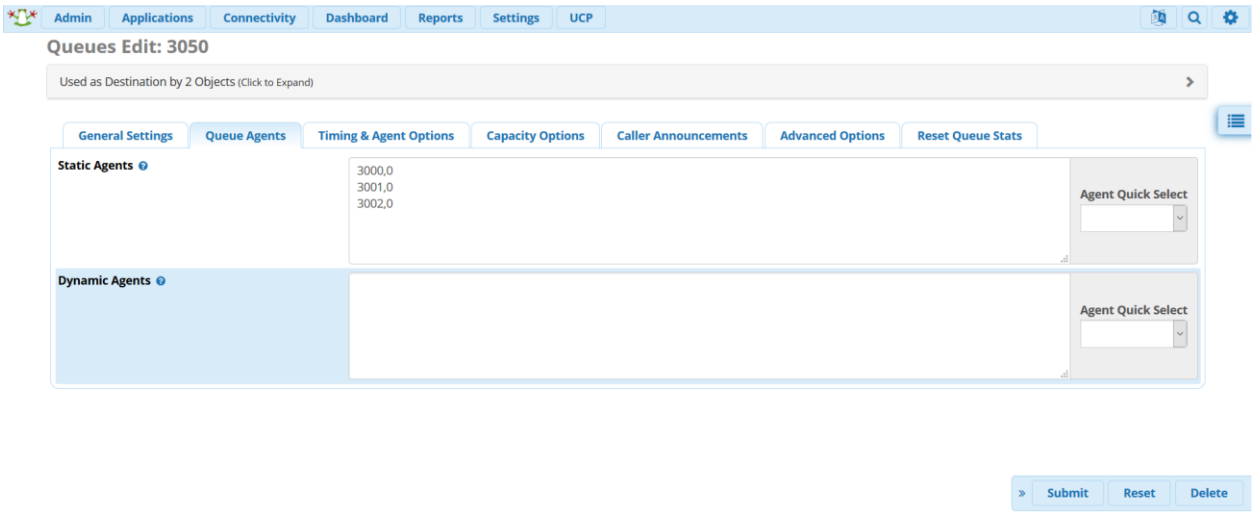


Figura 40: Captura de pantalla de FreePBX, ventana de configuración de colas.

3) Clic en la pestaña “Queue Agents”, Se agregan todas las extensiones creadas previamente en el campo Static Agents.

2.10.4 Configuración de *trunks*

Los *trunks* (troncales) son configuraciones que permiten a Asterisk enrutar llamas a través de dispositivos como el FXS-FXO gateway o el GSM gateway. Para fines de este proyecto de investigación será necesario hacer una configuración troncal para el módem k3765 y el SPA3000. Para configurar los *trunk* se siguen los siguientes pasos:

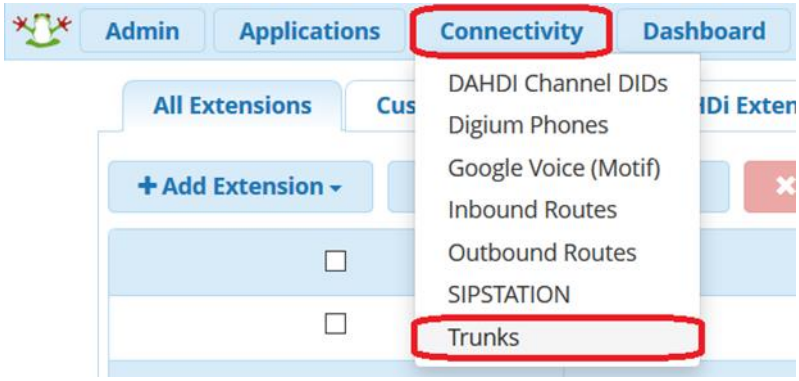


Figura 41: Captura de pantalla de FreePBX, pestaña “Connectivity”

1) Para ir la ventana de configuración de los troncales, primero hacer clic en la pestaña “Connectivity” luego clic en la opción trunks

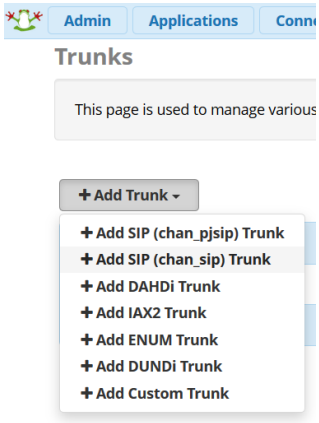


Figura 42: Captura de pantalla de FreePBX, ventana "Trunks"

2) Ya en la página de los *trunks*, haciendo botón "+Add Trunk" debemos elegir el tipo de *trunks* que se va a configurar. Para el gsm gw se debe elegir la opción "+Add Cuntom Trunk".

Figura 43: Captura de pantalla de FreePBX, ventana "Trunks"

3) Llenar el formulario que se ve en la figura 43 donde:

- ***Trunk Name*:** Asignar un nombre a la configuración del "Trunk".
- ***Outbound CallerID*:** se pone el número telefónico de la tarjeta sim del gsm sw.
- ***Máximum chaneles*:** se especifica la cantidad de llamadas simultáneas que se puede realizar en el trunk, si se deja en blanco como en este caso Asterisk no restringe el máximo de canales a utilizar.

Luego de llenar el formulario, clic en la pestaña "Custom Settings".

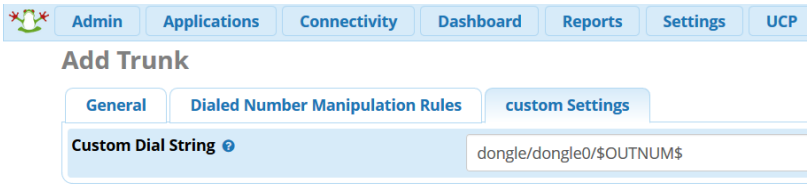


Figura 44: Captura de pantalla de FreePBX, ventana "Trunks"

4) Ya en la pestaña "Custom Settings" se le especifica la ruta al dispositivo gsm sw.
5) Después de configurado el troncal para el gsm gw, ese necesario agregar otro trocal para el pstn gw. para eso se debe volver acceder a la página de configuración de trocales y agregar un nuevo troncal esta vez con la opción sham_sip.

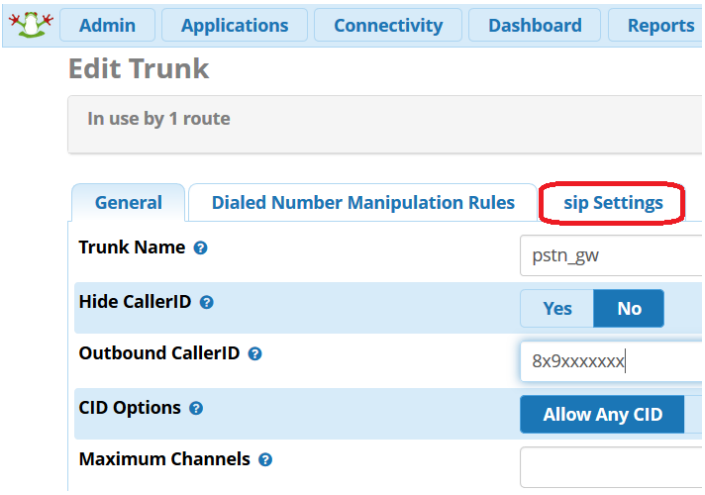


Figura 45: Captura de pantalla de FreePBX, ventana "Trunks"

6) La primera parte en igual que en la figura 43 a diferencia que aquí el (Outbound CallerID) sera igual al número telefónico donde se esté instalando el sistema. Ya hecho eso, clic en en la pestaña "sip Settings".

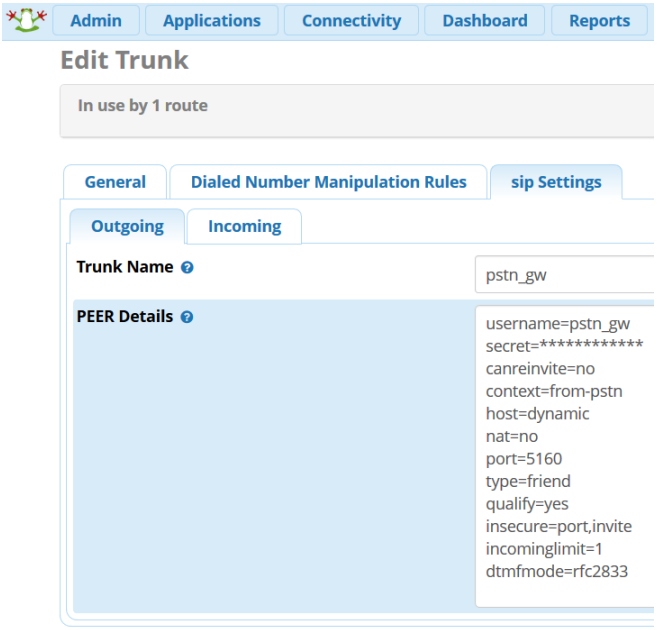


Figura 46: Captura de pantalla de FreePBX, ventana "Trunks"

7) Ya en la pestaña "sip Settings" hay dos campos para llenar: en el campo Trunk Name se asigna un nombre para la configuración y el PEER Details se coloca la configuracion manual del procolo sip para que se enlace con el SPA300 [39].

2.10.5 Rutas de entrada y salida

Ya hecha la configuración de los troncales, es necesario definir cómo se procesarán las llamadas entrantes y las llamadas salientes.

2.10.5.1 Rutas de entrada

Las rutas de entrada definen como el sistema procesará las llamadas entrantes. Se requiere crear una ruta tanto para las llamadas entrantes a través de la pstn como de las gsm. Estas se configuran siguiendo los siguientes pasos.

1) Para acceder a la página de configuración de las rutas de entrada, primero clic en la pestaña “connectivity” luego clic en la opción “Imbound Routes”.
2) Ya en la página de configuración de las rutas de entrada clic en el Boton “+Add inbound *router*).

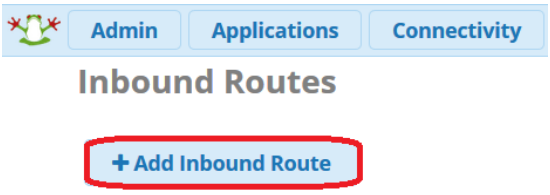


Figura 47: Captura de pantalla de FreePBX, ventana “Inbound Routes”

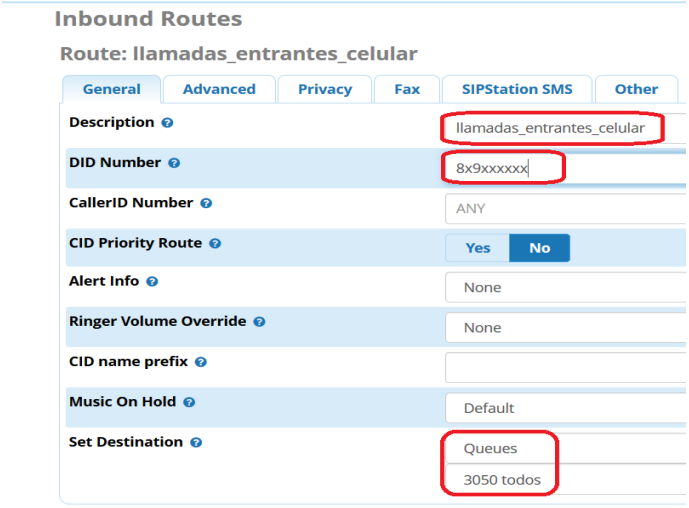


Figura 48: Captura de pantalla de FreePBX, ventana “Inbound Routes”

3) Llenar las siguientes 3 opciones del formulario de la figura 48.

- ***Description*:** se asigna un nombre para la configuracion
- ***DID Number*:** se especifica el número del truncal desde donde se va a recibir la llamada, o sea el número de la línea telefónica en el caso del ATA y el número de la tarjeta sim en caso del GSM gw.
- ***Set destination*:** se especifica dónde será dirigida la llamada. En este caso será a la cola 3050, que fue la cola creada anteriormente con todas las extensiones. Por lo tanto, cuando el sitema reciba una llamada por cualquiera de los trocales, todas las extensiones alertarán la llamada.

2.10.5.2 Rutas de salida

Las rutas de salidas se definen según el patrón de los números marcados. Las llamadas locales se realizan marcando el número de área y las llamadas a celulares o internacionales primero se marca el 1 y luego el número de área y el teléfono a llamar.

Para acceder a la página de la configuración de las rutas de salida, primero clic en la pestaña "connectivity" luego clic en la opción "Outbound Routes"

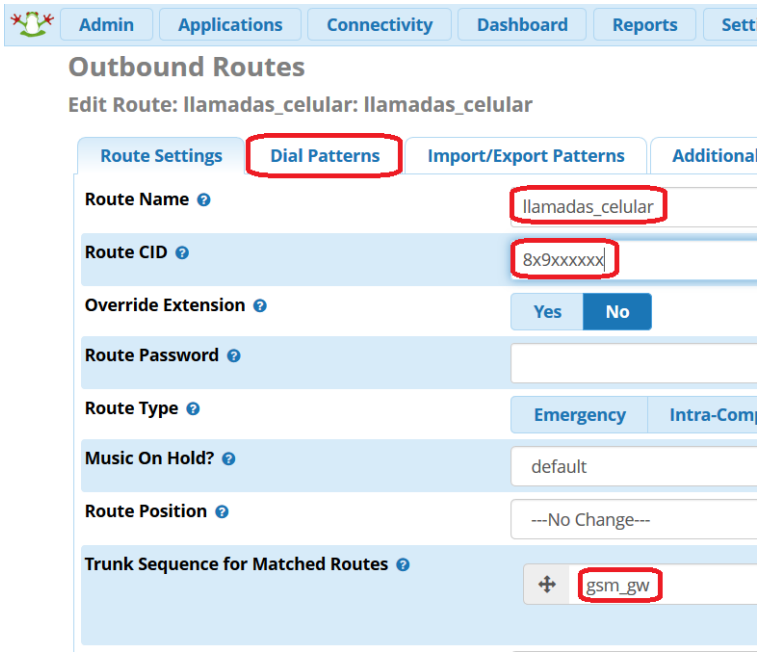


Figura 49: Captura de pantalla de FreePBX, ventana "Outbound Routes"

En formulario de la figura 49 solo se requiere llenar los siguientes campos:

- ***Route Name*:** asignar un nombre para la ruta
- ***Route CID*:** se coloca el número telefónico del *gateway* que se usará

- ***Trunk Sequence for Matched Routes*:** Se especifica el *gateway* que utilizará la ruta. Para fines de este trabajo de investigación se crearon dos rutas de salida, una para llamadas locales (utilizará el *trunk* con el SPA300) y otra para llamadas a celulares e internacionales (utilizando el *trunk* con el *gsm_gw*).

4) Llenados los campos, clic en la pestaña “dial Patterns”

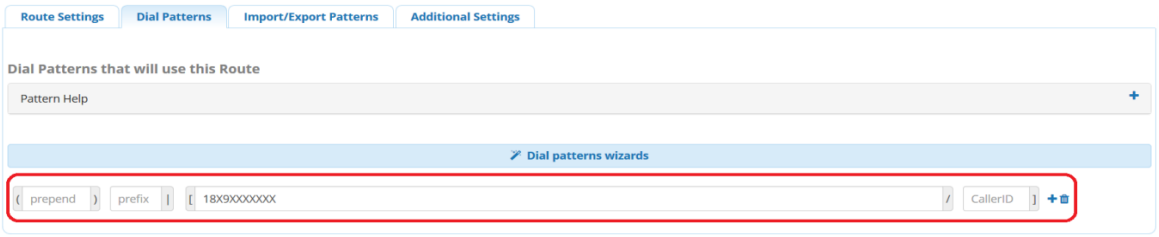


Figura 50: Captura de pantalla de FreePBX, ventana “Outbound Routes”

Ya en la pestaña “dial Patterns”, se especifica el patrón que definirá la ruta de salida a utilizar para llamadas locales. Se utilizó el patrón 8x9xxxxxxx donde Asterisk .

Este reconoce las x como cualquier número entre 0 y 9. Para llamadas a celulares e internacionales se usó el patrón 18x9xxxxxxx.

2.11 Configuración de SPA3000

La configuración del SPA3000 consta de dos partes. La configuración de una extensión, que hará posible utilizar un teléfono convencional con sistema VoIP y la configuración del trunk, para que Asterisk pueda enlazarse con el SPA3000 y enrutar llamadas a través de la PSTN.

2.11.1 Extensión ata

Para configurar la extensión en el SPA3000, primero se debe acceder a la WEBUI a través de un explorador de internet con la dirección IP 10.0.0.9 que fue la se le asignó fija a dicho dispositivo.

Figura 51: Captura de pantalla de WEBUI del SPA3000, inicio se sesión

Al acceder nos mostrará una página con varias opciones. Para acceder a la configuración de la extensión primero se debe hacer clic en el botón “Admin Login” luego al botón “advanced”.

Figura 52: Captura de pantalla de WEBUI del SPA3000, pestaña “Line 1”

Luego de acceder como usuario administrador hacer clic en la pestaña “Line 1”, una vez ahí se verá una página como en la figura 52 donde se debe de llenar los siguientes campos.

- SIP Port: puerto en trabaja el protocolo sip con las extensiones en el sistema 5060 para este caso
- Proxy: Dirección IP del servidor Asterisk.
- Display Name: el nombre que presentará al realizar una llamada
- User ID: la extensión que se configuró para el spa3000 que fue la 3002
- Password: la clave que se asignó a la extensión cuando se configuró Trunk ata

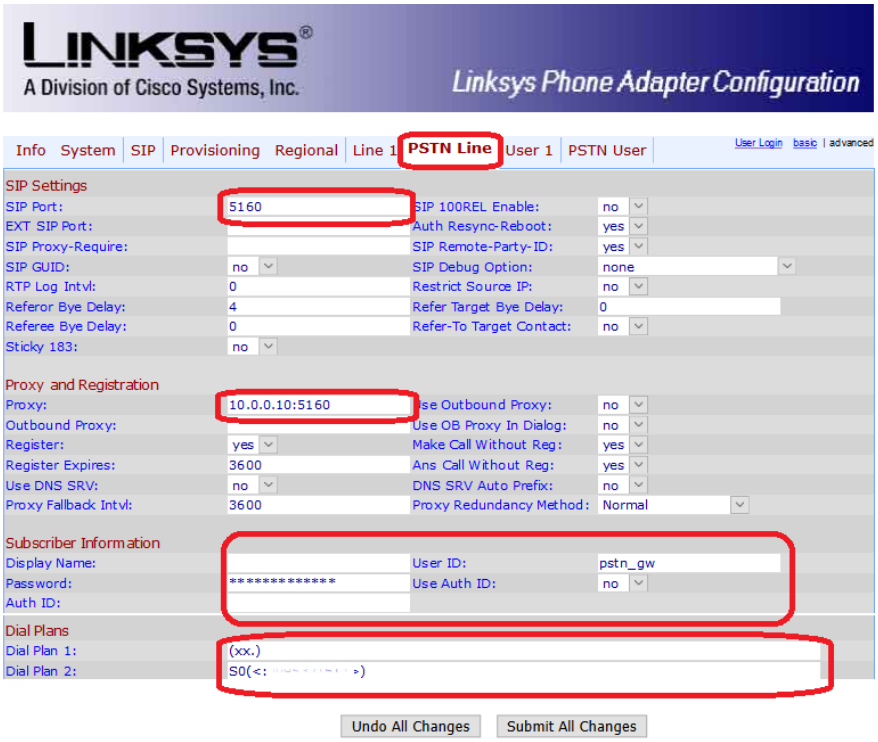


Figura 53: Captura de pantalla de WEBUI del SPA3000, pestaña "PSTN Line"

Para configurar el trunk el SPA300 hay que dirigirse a la pestaña PSTN Line. Una vez ahí se llenan los siguientes campos

- ***SIP Port***: puerto en que trabaja el protocolo sip con los troncales en el sistema, 5160 para este caso
- ***Proxy***: Dirección IP del servidor Asterisk. En este caso con él debe de usarse el formato IP:Purto
- User ID: el usuario que se especificó en la configuración del troncal pstn_gw en Asterisk
- Password: la clave que se asignó al troncal pstn_gw cuando se configuró.

- El dial plan se coloca S0(<:ruta de entrada>); así, cuando se reciba una llanada por el spa3000 este la enviará a la ruta de entrada correspondiente.

3 Puesta en marcha de un sistema VoIP

Ya realizada las configuraciones correspondientes al servidor, *router y los clientes. Ahora se explicará como poner en funcionamiento el sistema y se mostraran resultados de pruebas realizadas al sistema.*

3.1 Fotos del montaje final del proyecto

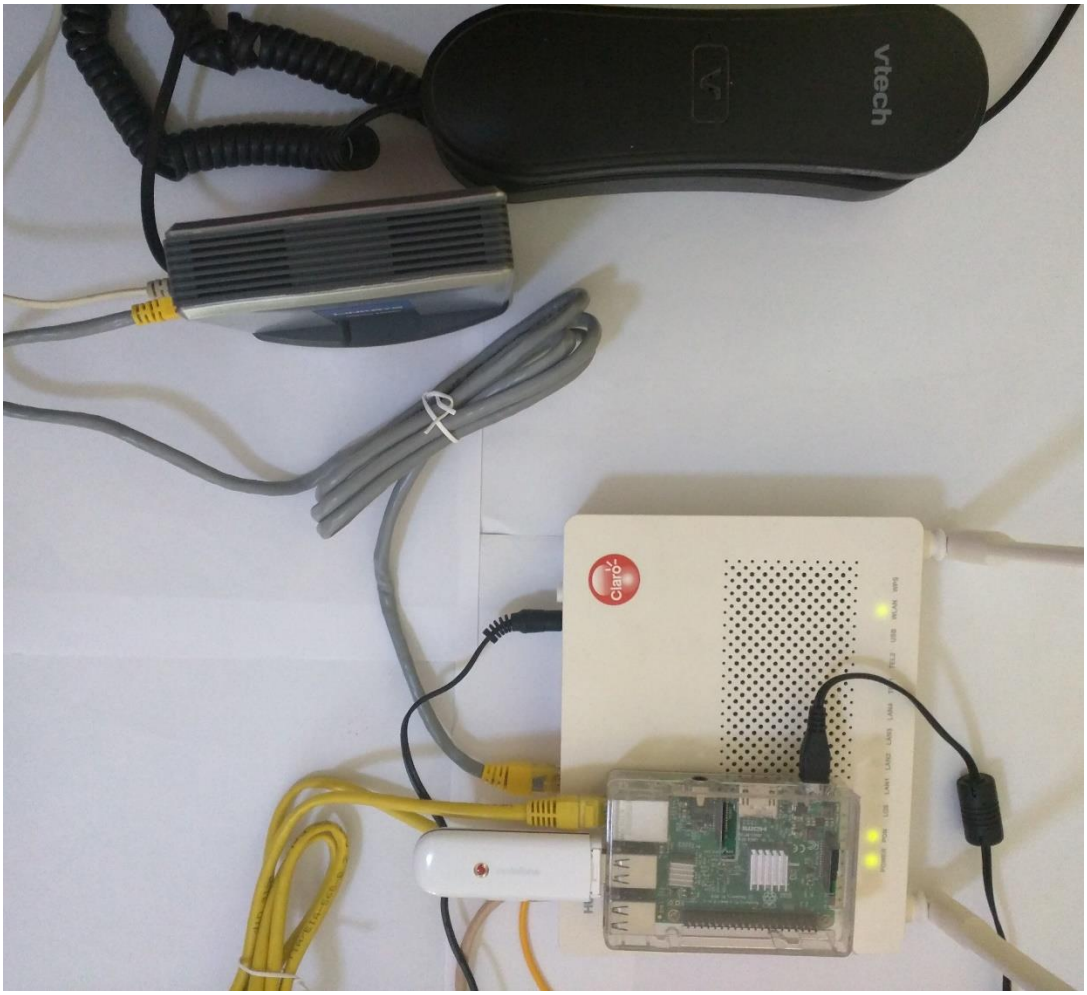


Figura 54: foto tomada del montaje final del proyecto.

Figura 55: foto tomada del montaje final del proyecto.

En las figuras 54 y 55 se muestra como quedo el sistema ya ensamblado.

3.2 Conectar clientes a la VPN

Para pruebas de este proyecto de investigación se configuraron dos clientes: uno en una computadora portátil y otro en un celular. A continuación, se mostrará cómo se conectan dicho cliente a la red VPN.

Conexión VPN de un cliente en una laptop:

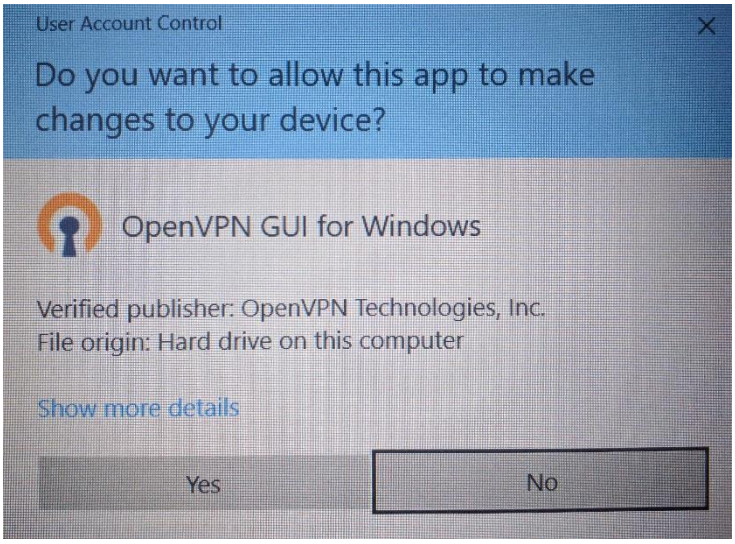


Figura 56: Captura de pantalla de la aplicación OpenVPN.

1) Al abrir la aplicación “OpenVPN GUI” para Windows aparecerá un mensaje como el que se ve en la figura 56, advirtiendo que se ejecutará como administrador. Se debe hacer clic en el botón “yes” para continuar.

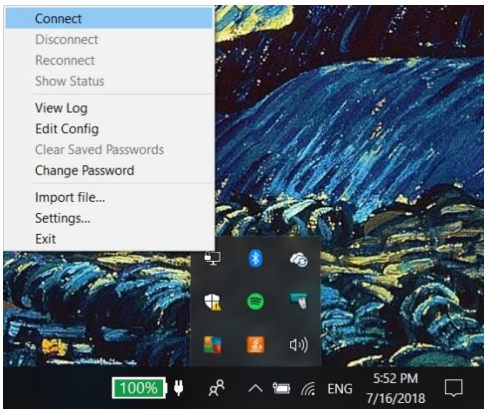


Figura 57: Captura de pantalla de la aplicación OpenVPN.

2) Una vez abierta la aplicación, clic derecho en el ícono de la aplicación que estará en la barra de trabajo de Windows. Después, clic en la opción “conectar”.

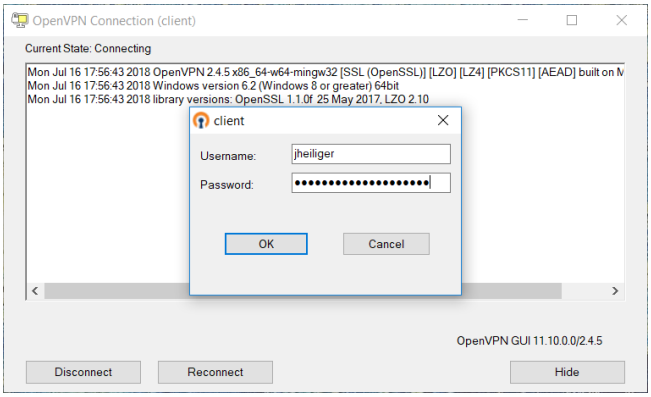


Figura 58: Captura de pantalla de la aplicación OpenVPN.

3) Luego solicitará las credenciales del usuario para acceder a la VPN. Clic en “OK”.

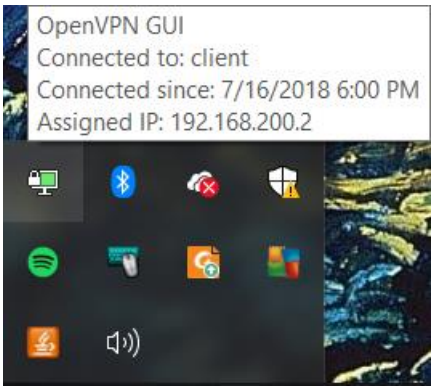


Figura 59: Captura de pantalla de la aplicación OpenVPN.

Cuando ya se establece la conexión, el ícono de OpenVPN cambia a uno como el que está en la figura 59, en la esquina superior izquierda.

Conexión de un cliente en un celular Android:

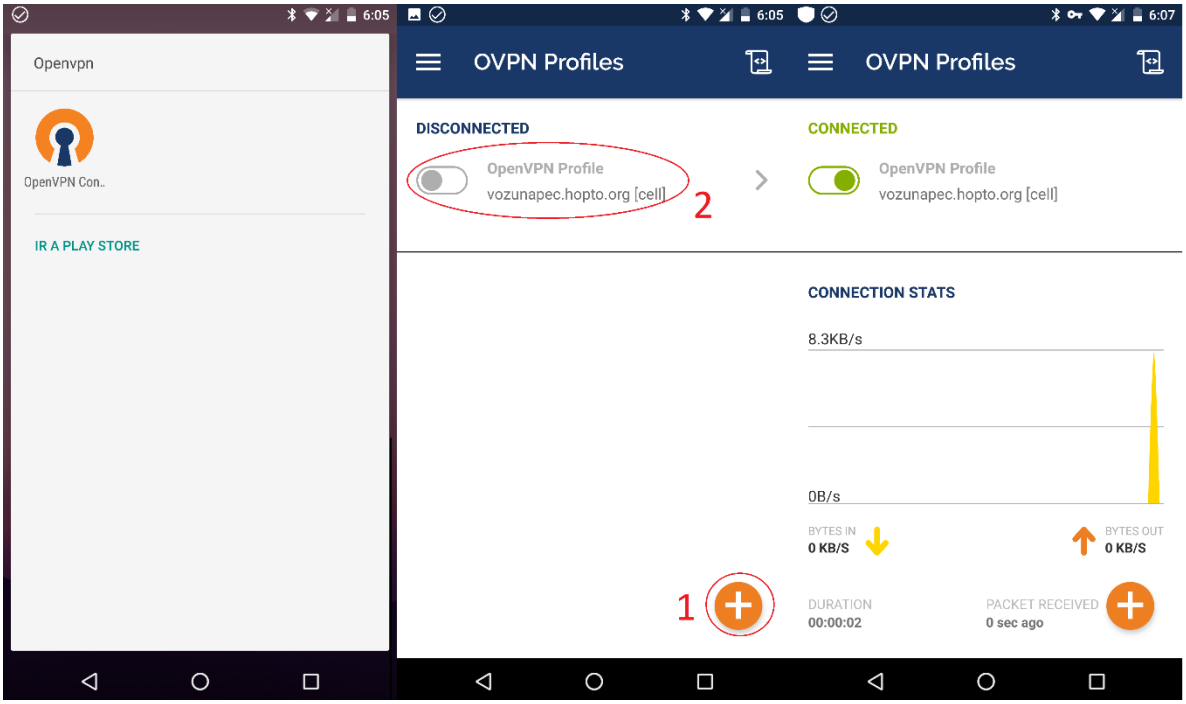


Figura 60: Captura de pantalla de la aplicación OpenVPN para celulares Android.

1) Abrir la aplicación OpenVPN en el celular.
2) Se le debe especificar dónde está ubicado el archivo de configuración que se usará para la conexión. Esto se hace usando la opción 1 de la figura 60.
3) Luego se puede acceder a la red VPN pulsando la opción 2 de la figura 60.
4) Si todo sale bien, la aplicación tomará un aspecto como se ve a la izquierda de la figura 60.

3.2.1 Pruebas de *ping* y de latencia

Las pruebas de *ping* y de latencia se hacen a la vez. La prueba de *ping* confirma si puede haber conexión entre dos *host*s y la de latencia mide el tiempo que dura un paquete para llegar de un extremo a otro. Estas pruebas se pueden hacer ejecutando el comando *ping* en cualquier sistema.

A continuación se muestra el resultado de ejecutar el comando *ping* en el celular de un cliente conectado a la VPN a través de una red GSM 4G.

```
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=145 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=141 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=40.5 ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=64 time=137 ms
64 bytes from 10.0.0.10: icmp_seq=5 ttl=64 time=133 ms
64 bytes from 10.0.0.10: icmp_seq=6 ttl=64 time=102 ms
64 bytes from 10.0.0.10: icmp_seq=7 ttl=64 time=168 ms
64 bytes from 10.0.0.10: icmp_seq=8 ttl=64 time=130 ms
64 bytes from 10.0.0.10: icmp_seq=9 ttl=64 time=150 ms
64 bytes from 10.0.0.10: icmp_seq=10 ttl=64 time=148 ms

...
...
...

--- 10.0.0.10 ping statistics ---
47 packets transmitted, 47 received, 0% packet loss, time 46078ms
rtt min/avg/max/mdev = 25.651/122.958/168.203/35.638 ms
```

Como resultado final muestra que de 47 paquetes enviados al RPI los 47 fueron enviados correctamente y hay una latencia promedio de 122.958 ms Es una latencia un poco más alta que el límite recomendado de 120 ms.

Ahora se mostrará el resultado de ejecutar el comando *ping* con el cliente VPN en una *laptop* conectada a la red wifi del campus 1 Unapec.

```
C:\Users\heili>ping -t 10.0.0.10
Pinging 10.0.0.10 with 32 bytes of data:
Reply from 10.0.0.10: bytes=32 time=12ms TTL=64
Reply from 10.0.0.10: bytes=32 time=109ms TTL=64
Reply from 10.0.0.10: bytes=32 time=14ms TTL=64
Reply from 10.0.0.10: bytes=32 time=22ms TTL=64
...
...
...
Ping statistics for 10.0.0.10:
    Packets: Sent = 40, Received = 40, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 312ms, Average = 30ms
C:\Users\heili>
```

La prueba hecha en campus 1 arrojó mejores resultados que la prueba hecha con la red GSM con una latencia promedio de 30 ms. Ese es buen escenario para el sistema VoIP.

3.3 Conectando teléfonos *softphone* al sistema

Ya hechas las pruebas de *ping* y latencia, entonces se puede iniciar el procedimiento de configuración de teléfonos de los clientes. Para fines de pruebas de este trabajo de investigación se usará el *softphone* Zoiper. Este puede ser utilizado de manera gratuita con el códec G711(U-Law).

Para configurar Zoiper en un teléfono Android se deben seguir los siguientes pasos:

1) Abrimos la aplicación Zoiper en el teléfono. Una vez abierta, esta tendrá una apariencia como se ve a la izquierda se la figura 61; entonces pinchamos en el ícono que está señalado con la flecha roja.

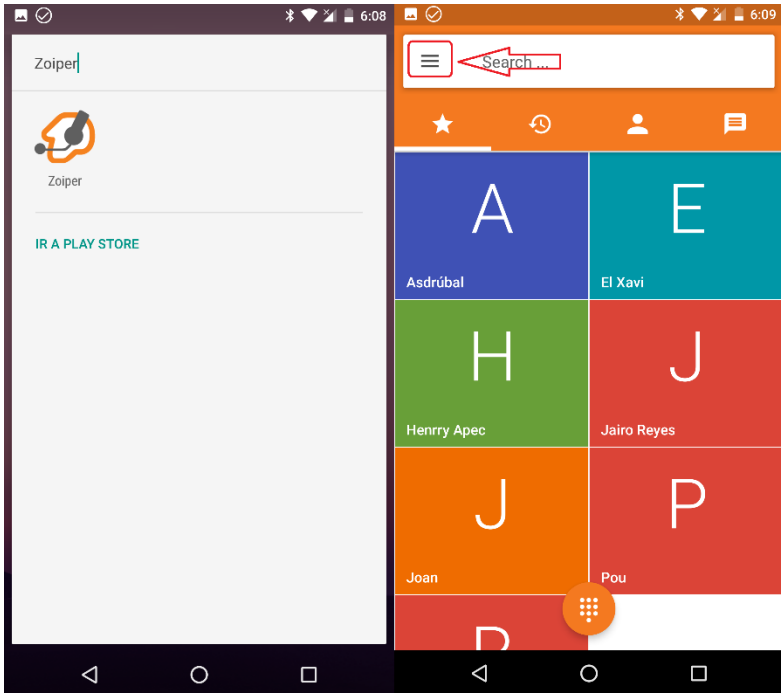


Figura 61: Captura de pantalla *softphone* Zoiper para teléfonos Android.

2) Una vez pulsado el ícono aparecerá un menú, como se ve en la parte derecha de la figura 62; pulsamos la opción “settings”.

3) Ya en “settings, aparecerá una formulario como el que se ve en el centro de la figura 62. Llenamos la sección “autenticación” con los datos de la extensión que pertenecerá al cliente. En este caso la 3001.

4) Pulsamos la tecla de retroceso del celular y si todo está correcto Zoiper indicará que la cuenta está conectada, como se ve a la derecha de la figura 62.

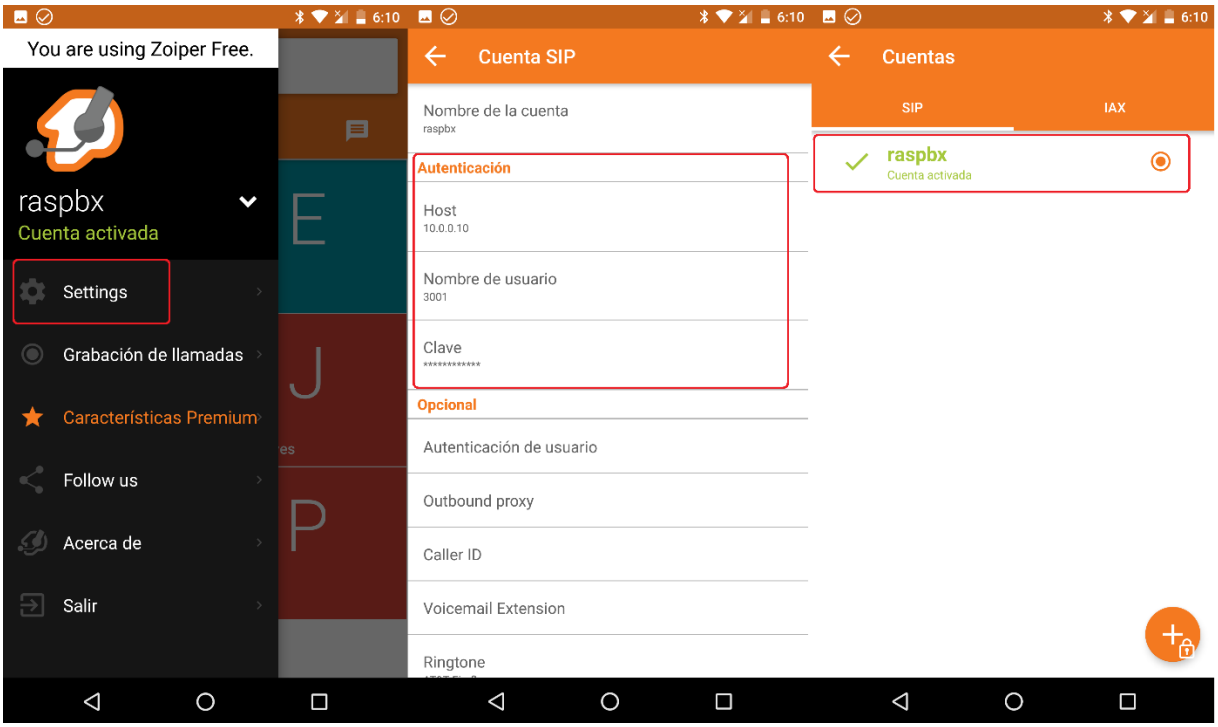


Figura 62: Captura de pantalla *softphone* Zoiper para teléfonos Android.

3.4 Pruebas de marcado y calidad de llamadas

Ya con los clientes conectados al sistema podemos pruebas de marcado y de calidad de llamada.

Las pruebas de marcado consisten en llamar a un número telefónico y si el número que recibe la llamada es el correcto, entonces la central VoIP tiene las señalizaciones DTMF debidamente configuradas.

La calidad de la llamada se puede calificar según la apreciación de las personas que están en la conversación.

Entre los archivos en el disco compacto anexos entregados con este trabajo de grado encontraremos una grabación de una llamada realizada a la universidad APEC. APEC cuenta con una central telefónica interactiva con sistema IVR, la cual cuando recibe una llamada le da indicaciones a la persona que llama para dirigir la llamada al departamento correcto. En esa llamada se hizo una prueba de marcado más relevante y se confirmó la calidad del sistema. La llamada fue realizada con el cliente *laptop* conectado al sistema con la red wifi del campus 1 de UNAPEC. También hay otras grabaciones de llamadas realizadas en distintas redes y distintos clientes.

CONCLUSIONES

Al culminar este trabajo de investigación, se ha logrado el diseño de un sistema VOIP usando software libre y hardware de bajo costo que permite realizar y recibir llamadas telefónicas ya sea a través una línea telefónica fija o una línea móvil, en un dispositivo móvil conectado a internet como una laptop, celular o tableta.

Los resultados de las pruebas hechas al sistema arrojaron muy positivos, donde podemos destacar que:

- Las pruebas de latencia en redes wifi publicas arrojaron resultados muy positivos ya que fue posible conseguir latencia de apenas 15 ms como fue en la red wifi del campus 1 UNAPEC. Un factor sorpresa fue la prueba la latencia en una red GSM 4G que se supones que son redes super rápidas. El resultado de la prueba arrojo un promedio de 122ms, lo que esta por encima del valor recomendado. Pero igual se pudo conseguir buena calidad en el sonido de las llamadas con una conexión GSM 4G .
- Las pruebas de marcado también fueron otra prueba exitosa del sistema. Lo que quiere decir que las configuraciones realizadas respeto a las señalizaciones DTMF fueron correctas.
- Las pruebas de llamadas fueron realizadas utilizando el códec G711(U-Law). La calidad del sonido en las llamadas en general fue buena. Pero cuando había otro tráfico en la red distinto al de sistema solía aparecer un ruido en la comunicación.

Sin duda un sistema como este podría ser una solución factible para reducir gastos en facturas de llamadas a largas distancia.

RECOMENDACIONES

- Implementar configuraciones de QoS(Calidad de servicio) en la red donde sea implementado el sistema VOIP, Así priorizar los paquetes o comunicación que estén destinado al servidor VOIP, de manera que si hay mucho tráfico generado por otros hosts en la red donde está el sistema no afecte la calidad de la voz por poca disponibilidad de ancho de banda.
- Instalar mecanismos de defensa contra ataques cibernéticos. Si el sistema no se protege bien un atáquenle podría acceder a él y generar grades costes de facturas en llamadas o dejar el sistema inútil.
- Hacer pruebas cómo se comporta el sistema con otros códecs de audio y otros protocolos de VOIP.
- Utilizar el modelo de Raspberry 3B+ en ves del modelo 3B. El modelo 3B+ tiene mejores cualidades de Hardware para trabajar con VOIP
- Investigar aspectos legales que pudieran impedir la implementación de un sistema como el diseñado en este trabajo, en [51]podemos ver que en algunos países es ilegal el uso de tecnología que eviten el pago de impuestos.

REFERENCIAS BIBLIOGRAFICAS

[1] L. S. Hernández, «Universidad Central “Marta Abreu” de Las Villas MININT en Camagüey,» Universidad Central “Marta Abreu” de Las Villas, Santa Clara Cuba, 2013.

[2] J. Janitor, «Efficient VoIP Solution for the Environment of the Technical University of Kosice,» Technical University of Kosice, Kosice, 2008.

[3] G. Mwansa, «VoIP system using open source software component in tertiary institutions: The case of the University of Namibia,» Gardner Mwansa, Namibia, 2007.

[4] T. Kelly, VoIP For Dummies, Indianapolis, Indiana: Wiley Publishing, 2005.

[5] W. Odom, CCENT/CCNA ICND1 100-105 Official Cert Guide, Indianapolis, IN 46240 USA: Cisco Press, 2016.

[6] R. Oppliger, «Internet and Intranet Security,» Artech House, 2001.

[7] T. Kelly, VoIP For Dummies 1st edition, Indianapolis, Indiana: Wiley Publishing, 2005.

[8] M. W. Candace Leiden, TCP/IP For Dummies, 6th Edition, Indianapolis, Indiana: Wiley Publishing, 2009.

[9] Fujitsu, «fujitsu.com,» 11 Abril 2006. [En línea]. Available: http://www.fujitsu.com/downloads/TEL/fnc/pdfservices/ethernet-prerequisite.pdf.

[10] K. W. R. James F. Kurose, «Computer Networking a Top-Down Approach,» Person, New Jersey, 2013.

[11] K. Wallace, «CompTIA Network+ N10-005 Authorized Cert Guide,» Person, Indianapolis USA, 2012.

[12] IETF, «TRANSMISSION CONTROL PROTOCOL RFC793,» 1981. [En línea]. Available: https://tools.ietf.org/html/rfc793.

[13] IETF, «User Datagram Protocol RFC768,» [En línea]. Available: https://www.rfc-editor.org/rfc/rfc768.txt.

[14] IETF, «INTERNET PROTOCOL RFC760,» 1981. [En línea]. Available: https://tools.ietf.org/html/rfc791.

[15] M. M. C. Fernando Boronat Segui, Direccionamiento e interconecxión de redes basadas en TCP/IP, España: Universidad Politecnica de Valencia. Servicio de publicacion , 2015.

[16] J. B. C. J. B. M. O. R. R. J. R. R. G. S. A. ,. F. S. C. Mª Del Carmen Romero Ternero, Redes Locales, España: Paraninfo, 2010.

[17] A. C. D. P. E. C. E. P. Z. Enzo Alegría Arias, «NAT y su relación con IPV6,» Universidad Tecnica Federico Santa Maria. , españa.

[18] IETF, «The IP Network Address Translator (NAT) rfc1631,» [En línea]. Available: https://tools.ietf.org/html/rfc1631.

[19] J. F. A. Juan, protocolos de enrutamiento, Cartegena De Indias : Universidad Tecnologica De Bolivar., 2006.

[20] J. B. Plaza, «Implantación de un sistema VOIP basado en Asterisk,» Universidad Peruana de Ciencias Aplicadas, Peru, 2009.

[21] D. I. R. C. Leslie Fernanda Monter Martínez, «uaeh.edu.mx,» 28 julio 2015. [En línea]. Available: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro27/.

[22] M. V. García, «VoIP: una puerta hacia la convergencia,» Universidad de Vigo, Galicia España, 1999.

[23] C. A. Naranjo, «Compresión De Audio En Base A La Aplicación De Técnicas De Cuantificación No Lineal Para Determinar El Efecto Sonoro De Su Reproducción En Los Seres Humanos.,» Universidad Tecnica De Ambato, Ecuador, 2010.

[24] V. C. M. L. Paola Renata Montenegro Cantos, «Análisis y evaluación para la selección de codecs de VoIP,» Cuenca, Ecuador, 2007.

[25] nefta.anaya, «http://elastixtech.com, MANUAL TEORICO CURSO ENTRENAMIENTO ELASTIX 2013,» 18 1 2013. [En línea]. Available: http://elastixtech.com/wp-content/uploads/2013/01/MANUAL-TEORICO-CURSO-ENTRENAMIENTO-ELASTIX-2013.pdf?x64472.

[26] E. C. Forero, «Estudio De La Influencia De Códecs Voip En El Consumo Energético En Smartphones Con Sistema Operativo Android,» Escuela Técnica Superior de Ingeniería Universidad de Sevilla, Sevilla, España, 2010.

[27] H. C. G. Flores, «Voz Sobre IP,» Escuela Superior De Ingeniería Mecánica y Electríca Unidad Culhuacan , D.F Mexico, 2009.

[28] A. N. Rendón, «aiu.edu,» junio 2008. [En línea]. Available: https://aiu.edu/spanish/publications/student/spanish/Comunnicacion%20de%20Systemas.html.

[29] D. Salomon, Compresión de datos la referencia completa, Springer-Verlag Londo: British Library, 2014.

[30] J. C. C. Pacheco, «Análisis Comparativo de Desempeño Entre Aodecs de Audio y Video Para Videoconferencia Sobre Infraestructura VPN,» Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2012.

[31] J. M. Ferrer, «Análisis De Herramientas De Gestión De VOIP,» Universidad De Sevilla, Sevilla, España, 2010.

[32] J. G. P. Priego, «JGTel: Una Herramienta para la Comunicación Interna de la UDLA-P,» Universidad de las Américas Puebla, Cholula, Puebla, México, 2000.

[33] IETF, «SIP: Session Initiation Protocol rfc3261,» 2002. [En línea]. Available: https://tools.ietf.org/html/rfc3261.

[34] L. R. J. D. Alban Méndez Carlos Billy, «Implementación De Un Sistema De Telefonía Ip Para La Empresa Isacnet S.A.,» EScuela Politécnica Nacional, Quito, Ecuador , 2010.

[35] N. Anaya, «elastixtech.com,» [En línea]. Available: http://elastixtech.com/sip-en-elastix/.

[36] IETF, «Session Initiation Protocol (SIP) Basic Call Flow Examples RFI3665,» 2003. [En línea]. Available: https://tools.ietf.org/html/rfc3665.

[37] IETF, « IAX: Inter-Asterisk eXchange Version 2 RFC5456,» 2010. [En línea]. Available: https://tools.ietf.org/html/rfc5456#page-87.

[38] C. S. C. Erik, «Consumo de ancho de banda en VoIP,» Universidad Nacional Pedro Ruiz Gall, Peru, 2017.

[39] L. M. J. V. M. Russell Bryant, Asterisk The Definitive Guide, EEUU: O'Reilly l, 2013.

[40] «adminso.es,» [En línea]. Available: http://www.adminso.es/index.php/Asterisk-ARQUITECTURA.

[41] A. G.́. Soria, «Plataforma domótica basada en Raspberry Pi y el protocolo MQTT,» Universidad de granada, Granada, España, 2017.

[42] C. Negus, Linux Bible, Canada: John Wiley & Sons, Inc, 2012.

[43] J. M. G. A. J. J. R. B. CARLOS ALBERTO CASTRO FLORES, «Remasterización De Un Sistema Operativo De Libre Distribución Con Aplicaciones De Carácter Pedagógico Para Ser Utilizado,» Universidad Fracisco Gavidia , Santa Ana, El Salvador , 2010.

[44] M. Feilner, Beginning OpenVPN 2.0.9, Olton, Texas: Packt Publishing Ltd, 2009.

[45] J. J. K. Eric F Crist, Mastering OpenVPN, Birmingham B3 2PB, UK: Packt Publishing Ltd, 2015.

[46] A. G. Morales, «Redes privadas virtuales,» Universidad Autónoma del Estado de Hidalgo, Estado de Hidalgo, Mexico, 2006.

[47] N. L. J. J. R. J. S. Michael Cross, Security+ Study Guide and DVD Training System, Rockland Massachusetts: Syngress Publishing, Inc, 2003.

[48] N. G. Markus Feilner, Beginning OpenVPN 2.0.9, Birmingham, Reino Unido: Packt Publishing Ltd, 2009.

[49] c. OpenVPN, «openvpn.net,» OpenVPN, 14 abril 2018. [En línea]. Available: https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage.

[50] N. Anaya, «elastixtech.com,» [En línea]. Available: http://elastixtech.com/protocolo-iax/.

GLOSARIO

Voz sobre ip (Voip): Es conjunto de técnicas y tecnologías utilizados para la comunicación de voz en redes IP.

GSM Gateway: Dispositivo que permite el enrutamiento entre una red de voz sobre IP y una red de teléfonos móviles GSM (sistema global para las comunicaciones móviles, del inglés Global System for Mobile communications).

Redes IP: Es un conjunto de quipos y dispositivos interconectados entre si para compartir datos utilizando el protocolo de internet IP.

Host: Equipo o dispositivo conectado en una red IP.

Dirección IP: Es la identidad de un host en red IP.

Dirección MAC: dirección de control acceso al medio (media access control address) es un identificador para interfaces de red, asignado por el fabricante.

DNS: sistema de nombres de dominio (Domain Name System, por sus siglas en inglés), es un sistema utilizado en redes IP para traducir nombres de dominio a una dirección IP o viceversa, con el fin de redirigir al host que consulta al dominio o dirección IP correspondiente.

DNS dinámico: es método que sirve para actualizar de manera automática información nombres de dominio en un sistema dns. Es usado en redes IP para asignar un nombre de dominio a un host con dirección variable.

VPN: Es método para conectar computadoras o dispositivos móviles a una red privada o LAN a través de internet.

Protocolo de comunicación: conjunto de reglas y estándar que determinan cómo los hosts en una red se comunican. Determinan el formato, la sincronización, la secuencia y el control de errores en la comunicación de datos.

open-vpn: Es un software de código abierto, es un entorno de trabajo diseñado para la implementación de VPNs.

Red LAN: Red de área local (local area network) es una red limitada al área de una localidad como casa, oficina, edificio o incluso un campus universitario.

Red WAN: Red de área amplia (wide area network) es una red que no está centrada en una sola localidad geográfica.

Software de código libre: son software cuyo código fuente y otros derechos son publicados bajo una licencia de código abierto. las licencias compatibles con la Open Source Definition permiten que los propietarios de los derechos de autor, permitir a los usuarios finales utilizar, hacer modificaciones y redistribuir el software, a cualquiera, para cualquier propósito (Laurent, 2008).

Linux: Sistema operativo de código libre, software que hace de interface para el usuario de una computadora entre el hardware o demás software de un sistema informático.

Asterisk: Software de código abierto creado por la empresa Digium, es un entorno de trabajo para crear aplicaciones de comunicación como Centrales telefónicas privadas, centros de llamadas, sistemas IVR, puente conferencias. (Russell Bryant, 2013).

Open-vpn: software de código libre utilizado para gestionar VPNs.

Softphone: Software que emula las funciones de un teléfono, permite a los usuarios realizar y recibir llamadas desde computadoras y teléfonos móviles.

ANEXOS

APÉNDICE A: Contenido del CD

- ***Trabajo de grado digital.***

- Documento versión Word 2013.

- Documento versión PDF 2013.

Video: "Propuesta de aplicación de un sistema de voz sobre IP utilizando Raspberrry pi, Asterisk, open-vpn y dns dinámico en campus II, UNAPEC**"**. En este video se muestra la prueba de funcionamiento del sistema.