



UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

ESTUDIO DE UNA APLICACIÓN VOZ SOBRE IP (VoIP) UTILIZANDO SERVIDORES DE BAJA CAPACIDAD DE PROCESO DE INFORMACIÓN

Alumno: Rafael Mesa Hidalgo

Tutor: Prof. D. Joaquín Cañada Bago

Dpto.: Ingeniería de Telecomunicación

Junio, 2016



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Ingeniería de Telecomunicación

Don Joaquín Cañada Bago, tutor del Proyecto Fin de Carrera titulado:
Estudio de una aplicación Voz sobre IP (VoIP) utilizando servidores de baja
capacidad de proceso de información, que presenta Rafael Mesa Hidalgo,
autoriza su presentación para defensa y evaluación en la Escuela Politécnica
Superior de Jaén.

Jaén, Junio de 2016

El alumno:

Rafael Mesa Hidalgo

Los tutores:

Joaquín Cañada Bago

AGRADECIMIENTOS

A mis padres, por todos los valores que me han inculcado desde pequeño, por su gran esfuerzo en darme siempre la mejor educación y motivarme para conseguir todos los objetivos que me he propuesto.

A mis hermanos y amigos de toda la vida, porque siempre han estado cuando más los necesitaba.

A Martyna, por el apoyo y cariño que me ha dado siempre, por su inmensa paciencia durante todo este tiempo y por toda la motivación para realizar este proyecto.

A mis compañeros de clase en especial a Ana, Javi, Pablo y Alberto por haber sido siempre un grupo en los malos y buenos momentos.

A la universidad de Jaén, por haberme cedido la mayoría de material del proyecto.

Finalmente me gustaría agradecerle a mi tutor del proyecto Joaquín Cañada todo el tiempo y esfuerzo que me ha dedicado durante la realización de este trabajo.

Índice

1. Introducción.....	10
1.1. Motivación.....	10
1.2. Objetivos.....	11
1.3. Estructura del documento	12
2. Antecedentes	13
2.1. Teléfono.....	15
2.2. Video.....	15
2.3. Internet RDSI	16
2.4. Telefonía VoIP	17
2.5. Videoconferencia	20
2.6. Ventajas de utilizar Voz sobre IP.....	22
3. Materiales y Métodos	23
3.1. Infraestructura de red	23
3.1.1. Red LAN	23
3.1.2. VPN	25
3.1.3. VLAN	26
3.1.4. QoS.....	28
3.2. Equipos terminales.....	30
3.2.1. Teléfonos VoIP.....	30
3.2.2. Gateway FXS.....	32
3.2.3. Gateway FXO	33
3.2.4. Gateway GSM.....	35
3.3. Protocolos de señalización VoIP	36
3.3.1. SIP	36
3.3.2. IAX.....	38
3.3.3. Diferencias entre SIP e IAX.....	39
3.4. Protocolos de transmisión	40
3.4.1. RTP.....	40
3.4.2. RTCP	40
3.5. Códecs.....	40
3.5.1. Códecs de voz	42
3.5.2. Códecs de video	44
3.6. Equipos de baja capacidad de procesamiento	46
3.7. Servidores VoIP (Software).....	51
3.7.1. Asterisk	51

3.7.2.	FreePBX	53
3.8.	Herramientas de análisis	57
3.8.1.	WireShark	57
4.	Resultados	58
4.1.	Configuración y puesta en marcha de RasPBX en Raspberry Pi.....	59
4.2.	Configuración y puesta en marcha de uElastix en Raspberry Pi	60
4.3.	Instalación de equipos en laboratorio	70
4.4.	Pruebas en laboratorio	72
4.5.	Conexión entre diferentes sedes mediante una troncal IAX	76
4.6.	Configuración de llamadas externas mediante troncal SIP	80
4.7.	Configuración de llamadas externas mediante troncal GSM	85
4.8.	Configuración de llamadas externas mediante troncal FXO	87
4.9.	Comparación con soluciones comerciales.....	92
5.	Estudio de viabilidad	93
5.1.	Planificación temporal	93
5.1.1.	Tareas.....	93
5.2.	Estimación de costes	96
5.2.1.	Presupuesto.....	96
6.	Conclusiones.....	100
7.	Líneas de Futuro	102
	Bibliografía	103

Índice de Ilustraciones

Ilustración 1: Troncal SIP	20
Ilustración 2: Router	24
Ilustración 3: Switch	24
Ilustración 4: Esquema VPN.....	26
Ilustración 5: Esquema VLAN.....	27
Ilustración 6: Fases de servicios diferenciados.....	29
Ilustración 7: Teléfono VoIP	31
Ilustración 8: Teléfono VoIP 2.....	32
Ilustración 9: SoftPhone Xlite.....	32
Ilustración 10: Gateway FXS	33
Ilustración 11: Esquema de Red Gateway FXO	35
Ilustración 12: Modem GSM	36
Ilustración 13: Señalización SIP	37
Ilustración 14: Características del códec H.264.....	46
Ilustración 15: Logo Raspberry Pi.....	47
Ilustración 16: Eben Upton	47
Ilustración 17: Raspberry Pi 1 modelo A.....	48
Ilustración 18: Raspberry Pi 1 modelo B.....	49
Ilustración 19: Raspberry Pi 2 Modelo B.....	49
Ilustración 20: Asterisk Logo	51
Ilustración 21: Mark Spencer	51
Ilustración 22: FreePBX Logo.....	53
Ilustración 23: RasPBX.....	55
Ilustración 24: Elastix Logo.....	56
Ilustración 25: Elastix	56
Ilustración 26: WireShark	57
Ilustración 27: Instalación RasPBX.....	59
Ilustración 28: Instalación uElastix 1.....	61
Ilustración 29: Instalación uElastix 2.....	61
Ilustración 30: Instalación uElastix 3.....	62
Ilustración 31: Instalación uElastix 4.....	63
Ilustración 32: Instalación uElastix 5.....	64
Ilustración 33: Instalación uElastix 6.....	66
Ilustración 34: Instalación uElastix 7.....	69
Ilustración 35: Esquema de red, distribución de equipos	70
Ilustración 36: Instalación en laboratorio 1.....	71
Ilustración 37: Instalación en laboratorio 2.....	72
Ilustración 38: Llamadas simultaneas en uElastix.....	73
Ilustración 39: Análisis WireShark	76
Ilustración 40: Esquema de red, Troncal IAX.....	76
Ilustración 41: Configuración Troncal IAX 1.....	78
Ilustración 42: Configuración Troncal IAX 2.....	79
Ilustración 43: Configuración Troncal SIP 1.....	81
Ilustración 44: Configuración Troncal SIP 2.....	81
Ilustración 45: Configuración Troncal SIP 3.....	82
Ilustración 46: Configuración Troncal SIP 4.....	83

Ilustración 47: Configuración Troncal SIP 5.....	83
Ilustración 48: Configuración Troncal SIP 6.....	83
Ilustración 49: Configuración Troncal SIP 7.....	84
Ilustración 50: Configuración Troncal SIP 8.....	84
Ilustración 51: Configuración Troncal SIP 9.....	85
Ilustración 52: Configuración Troncal GSM 1	85
Ilustración 53: Configuración Troncal GSM 2	86
Ilustración 54: Configuración Troncal GSM 3	86
Ilustración 55: Configuración Troncal GSM 4	87
Ilustración 56: Configuración Troncal FXO 1	88
Ilustración 57: Configuración Troncal FXO 2	88
Ilustración 58: Configuración Troncal FXO 3	89
Ilustración 59: Configuración Troncal FXO 4	90
Ilustración 60: Configuración Troncal FXO 5	90
Ilustración 61: Configuración Troncal FXO 6	91
Ilustración 62: Configuración Troncal FXO 7	91
Ilustración 63: Configuración Troncal FXO 8	92
Ilustración 64: Elastix MicroUCS	92
Ilustración 65: Placa ASIRI.....	93
Ilustración 66: Diagrama de Gantt	95

Acrónimos

ADSL. Asymmetric Digital Subscriber Line

FTP. File Transfer Protocol

FTTH. Fiber To The Home

FXO. Foreign eXchange Office

FXS. Foreign eXchange Subscriber

GPL. General Public License

GSM. Global System for Mobile communications

IAX. Inter-Asterisk eXchange protocol

IETF. Internet Engineering Task Force

IP. Internet Protocol

IVR. Interactive Voice Response

LAN. Local Area Networks

NAT. Network Address Translation

OSI. Open System Interconnection

PCM. Pulse Code Modulation

QoS. Quality of Service

RDSI. Red Digital de Servicios Integrados

RSVP. Resource Reservation Protocol

RTB. Red Telefónica Básica

RTCP. Real-time Control Protocol

RTP. Real-time Transport Protocol

SIP. Session Initiation Protocol

SSL. Secure Sockets Layer

TCP. Transmission Control Protocol

UDP. User Datagram Protocol

VLAN. Virtual LAN

VoIP. Voice over IP

VPN. Virtual Private Network

1. Introducción

En los últimos años se ha producido un notable incremento en el uso de tecnologías de Voz sobre IP (VoIP), que engloba todas aquellas tecnologías que están involucradas en la transmisión de voz entre dos o más participantes a través de una red de conmutación de paquetes IP.

Este incremento está relacionado con el avance en las tecnologías de comunicación y la mejora de las infraestructuras de telecomunicaciones, lo que ha repercutido en un acceso a Internet más rápido y barato para los usuarios y ha permitido la difusión de las tecnologías de VoIP.

Actualmente los proveedores de servicios de Internet ofrecen conjuntamente servicios VoIP que compiten con la telefonía convencional, puesto que VoIP proporciona más versatilidad que la telefonía tradicional, dando la posibilidad de integrar la transmisión de voz y datos en una sola red.

1.1. Motivación

Uno de los motivos más importantes que me han llevado a desarrollar este proyecto ha sido la estrecha relación que guarda con tecnologías de telecomunicaciones, el área de la informática con el que más identificado me siento.

La tecnología VoIP es el futuro de las comunicaciones telefónicas tanto fijas como móviles, teniendo en cuenta que la sociedad actual vive en un mundo conectado y que el uso de esta tecnología está en auge, considero muy útil la realización de este proyecto.

Para el desarrollo de este proyecto se ha utilizado como servidor principal una placa hardware de propósito general y de bajo rendimiento llamada Raspberry Pi la cual ha permitido integrar el denominado IoT (Internet de la cosas) que tan demandado está actualmente.

El año pasado cursé una asignatura muy relacionada con aspectos técnicos de este proyecto, fue entonces cuando descubrí el interés tan grande que tenía hacia la tecnología VoIP y los sistemas de transmisión de audio y video en streaming.

Para finalizar este apartado quería también indicar que durante el último trimestre del año 2015 estuve realizando prácticas de empresa Erasmus en Polonia en las cuales mi tarea fue la modernización del sistema telefónico existente con la implantación de un nuevo sistema de comunicación VoIP, fue allí donde aprendí realmente la configuración de estos sistemas ya que me encontraba en un escenario real, actualmente la empresa cuenta con 50 terminales VoIP en funcionamiento.

1.2. Objetivos

Como objetivos a alcanzar en este proyecto encontramos:

- Verificar la posibilidad de instalar un servidor de VoIP Asterisk en sistemas de bajas prestaciones como Raspberry Pi (versión 1 modelo B y versión 2)
- Verificar la compatibilidad con aplicaciones y protocolos estándar como RTP, RTCP y SIP.
- Instalación de Teléfonos VoIP Cisco disponibles en el departamento de informática.
- Instalación y comparación de los distintos softphones disponibles tanto para plataformas Linux / Windows y Android.
- Verificar la posibilidad de realizar llamadas externas a través de una troncal SIP o de una línea analógica utilizando un gateway ATA.
- Verificar la posibilidad de realizar un enlace entre dos servidores VoIP a través de una conexión IAX.
- Cuantificar el número máximo de usuarios registrados.
- Cuantificar el número máximo de conversaciones simultáneas que soportan dichos sistemas.
- Estudiar la posibilidad de videoconferencias.
- Configurar troncales IAX entre diferentes sedes donde hay instaladas una Raspberry Pi con el servidor Asterisk.
- Análisis de los protocolos SIP, RTP, RTCP, UDP y TCP a través de la aplicación WireShark.

1.3. Estructura del documento

El proyecto ha sido estructurado en siete capítulos, partiendo de una vista más general, hasta llegar a un punto de vista más concreto y específico.

Para ello se ha tratado de describir la evolución que ha tenido la telefonía desde su invención hasta la situación actual.

Los siete capítulos anteriormente mencionados han sido ordenados con el criterio de explicar conceptos teóricos en los primeros capítulos y llevarlos a la práctica en los capítulos siguientes.

- **Capítulo 1 – Introducción**

En el primer capítulo se mostrarán los motivos por los cuales se realiza este proyecto, enumerando los objetivos a alcanzar durante su desarrollo.

- **Capítulo 2 – Antecedentes**

En este segundo tema se tomará como referencia las tecnologías utilizadas hasta la fecha antes de la aparición de telefonía VoIP. Se hará una comparativa entre las ventajas e inconvenientes de esta tecnología.

- **Capítulo 3 – Materiales y Métodos**

Este es uno de los capítulos más extenso e importantes del proyecto ya que en el se explica todas las tecnologías utilizadas en lo que se refiere a hardware y software. Se trata de explicar con detalle todas las características de manera teórica para posteriormente llevarlas a la práctica en capítulos siguientes.

- **Capítulo 4 – Resultados**

En este capítulo se describen cómo se ha configurado todos los elementos previamente descritos y las diferentes opciones que se han tomado acerca de ofrecer un mejor servicio a menor coste, se muestran todos los resultados de las pruebas realizadas en laboratorio y los pasos a seguir para su puesta en marcha.

- **Capítulo 5 – Estudio de viabilidad**

Uno de los factores más importantes antes de comenzar un proyecto es conocer si es viable en cuestiones económicas y de tiempo, puesto que vamos a desarrollar una solución que podría ser implantada en el ámbito empresarial. En este capítulo se realiza una planificación temporal y un estudio de costes para saber si la ejecución de este proyecto es viable.

- **Capítulo 6 – Conclusiones**

Finalmente se analiza la información obtenida durante todo el proyecto llegando a unas conclusiones finales donde se plantean las diferentes soluciones o opciones a tomar en función del tipo de usuario final.

- **Capítulo 7 – Líneas de Futuro**

En el séptimo y último capítulo se describen las posibles opciones de futuro que podría tomar este proyecto, trabajos que se basan en proporcionar a este sistema soluciones frente a posibles problemas asociados al corte en los distintos servicios.

Durante la redacción del proyecto se ha tratado de documentar de manera clara, explícita y haciendo uso de esquemas de red, todos los aspectos técnicos relacionados con la interconexión de los dispositivos, tratando de aclarar con las ilustraciones las dificultades que puedan suponer la configuración de cada elemento.

2. Antecedentes

En este capítulo describiremos los sistemas de comunicación existentes hasta la llegada de la telefonía VoIP y haremos una comparación entre ellas.

Desde sus orígenes, el ser humano siempre ha tenido la necesidad de comunicarse con otros seres de su misma especie, con el fin de compartir cualquier tipo de información entre ellos.

Comenzando por señales de humo, continuando con palomas mensajeras y llegando hasta sistemas más actuales y relacionados con este proyecto, podríamos

decir que la comunicación entre personas es vital e imprescindible para vivir en un mundo conectado.

Antes de que existiera el sistema telefónico tal y como lo conocemos hoy en día, los únicos sistemas de comunicación eran el correo postal y el telégrafo, los cuales tenían una serie de desventajas frente al teléfono.

El sistema postal comenzó hace siglos, hoy en día se sigue utilizando ya que es una manera muy económica y práctica de enviar gran cantidad de información a cualquier parte del mundo, una de sus mayores limitaciones es el tiempo, puesto que las cartas son transportadas físicamente desde el emisor hasta el receptor por carretera, si la información que se desea enviar no es urgente, el sistema postal es una opción muy acertada, sin contar con los problemas de seguridad que conlleva ya que las cartas son gestionadas por personas que pueden vulnerar la seguridad de las mismas.

Posteriormente apareció el telégrafo que fue el antecesor al teléfono, este dispositivo comenzó a instalarse en la mayoría de las oficinas postales de todas las ciudades y los usuarios debían dirigirse hasta allí para enviar un telegrama.

El telégrafo utilizaba el lenguaje Morse para transmitir códigos que son interpretados por otro telégrafo en destino. Este sistema era notablemente más rápido que el sistema postal pero su limitación era que los telégrafos se cobraban por palabras, es decir que los mensajes debían ser cortos y concisos.

Hoy en día el mundo no es consciente de la revolución en aspectos de comunicación que surgió con la aparición del teléfono. Si comparamos el estado actual con el de hace cien años la diferencia es abismal, ya que la inmensa mayoría de los hogares cuentan con un teléfono con el que pueden realizar llamadas a cualquier parte del mundo con solo marcar unos dígitos en el teclado, hace cien años esto era impensable puesto que toda la información tardaba semanas e incluso meses en llegar a destino.

2.1. Teléfono

En 1876 fue registrada por **Alexander Graham Bell** la primera patente sobre el teléfono, no entraremos en debates sobre quién fue el descubridor de tal invento ya que no es relevante en este proyecto.

Podríamos decir que desde sus comienzos, el teléfono como tal, poco o nada ha cambiado en relación a su funcionamiento interno ya que sigue contando en general con los mismos elementos desde su creación.

Un teléfono es uno de los dispositivos más simples que hay en los hogares, en su interior encontramos:

- Un **interruptor** que es utilizado para conectar y desconectar el teléfono de la línea telefónica.
- Un **altavoz** empleado para reproducir la voz que es transmitida a través de la línea.
- Un **micrófono** que recoge la voz del emisor para posteriormente ser enviada a través de la línea al receptor.
- Un **teclado numérico** empleado para marcar el número de teléfono del destinatario.

Sea cual sea la marca y el modelo del teléfono, todos cuentan como mínimo con estos elementos. Sin embargo el funcionamiento interno es asombroso ya que es capaz de convertir la voz humana en impulsos eléctricos y viceversa.

Cuando el emisor comienza a hablar en origen el teléfono recoge la voz y la digitaliza a una frecuencia de muestreo de 8000 muestras por segundo, cada muestra se codifica en 8 bits, es decir que son necesarios solo 64 Kbit/s para transmitir voz sin compresión por la línea telefónica, en capítulos posteriores trataremos sobre este apartado con más detalle, en el capítulo de códecs de audio.

2.2. Video

Desde sus comienzos, el video era generado por cámaras de video y gestionado de manera local, es decir que no era transmitido a través cables u ondas. Posteriormente apareció la televisión y con ello la retransmisión de video, pero sólo

en un sentido puesto que los receptores no interaccionaban con el emisor, es decir que los telespectadores sólo consumían la información.

Con el desarrollo y la evolución de Internet fue posible la transmisión de contenido de todo tipo, como por ejemplo audio y video pero a modo de fichero para ser reproducido en destino ya que la red no soportaba streaming.

Finalmente en la década de los ochenta se consiguió aumentar el ancho de banda de las conexiones de Internet y asegurar un flujo de datos continuo, fue entonces cuando los sistemas de videoconferencias tuvieron cabida en Internet.

2.3. Internet RDSI

RDSI significa Red Digital de Servicios Integrados y fue la evolución de la RTB (Red Telefónica Básica) ya que la RDSI contaba con muchas ventajas como la digitalización en las comunicaciones. RDSI fue la tecnología de acceso intermedia entre RTB y ADSL, ya que las líneas analógicas existentes hasta el momento no permitían conectarse a internet a la vez que se mantenía una llamada telefónica. RDSI facilitaba conexiones digitales extremo a extremo. [1]

La tecnología de acceso RDSI utilizaba dos tipos de canales.

- Canal **B**: utilizado para transportar voz o datos a una velocidad de 64 Kbit/s.
- Canal **D**: utilizado para la señalización y control de las comunicaciones a una velocidad entre 16 y 64 Kbit/s.

Las compañías telefónicas ofrecían dos tipos de conexiones RDSI, una destinada al uso particular y otra para empresas.

- **Acceso Básico (BRI)**

Se trataba de dos canales de tipo B y uno de tipo D a 16 Kbit/s que ofrecían la posibilidad de mantener dos llamadas telefónicas simultáneamente. La conexión desde la central hasta el domicilio del cliente RDSI se realiza a través de un par de hilos de cobre (igual que el RTB)

- **Acceso Primario (PRI)**

E1: Esta modalidad proporcionaba 30 canales tipo B y uno de tipo D a 64 Kbit/s. Este tipo de conexiones era utilizado por empresas para mantener hasta 30 conversaciones telefónicas simultáneas, conexión muy utilizada para las centralitas digitales privadas, este tipo de acceso primario se utilizaba principalmente en Europa y Australia.

T1: Esta modalidad ofrecía 23 canales de tipo B y uno de tipo D, mayoritariamente utilizado en Estados Unidos, Canadá y Japón.

RDSI fue diseñada para transportar voz y datos sobre un único medio, con la idea de reutilizar la infraestructura existente ya que existía un despliegue enorme de pares de hilos de cobre.

Posteriormente para el uso particular surgió la tecnología de acceso ADSL (Asymmetric Digital Subscriber Line) la cual permitía a los usuarios conectarse a internet con un ancho de banda superior al de los antiguos modem y además poder mantener una llamada telefónica simultáneamente, hoy en día muchos hogares siguen utilizando conexiones ADSL.

En el sector empresarial todavía quedan empresas con conexiones RDSI (PRI) ya que siguen utilizando centralitas telefónicas antiguas, la evolución de las conexiones RDSI para las empresas es la fibra óptica, llamada FTTH, por la que se envía la voz y los datos a velocidades muy superiores que RDSI y ADSL.

2.4. Telefonía VoIP

La telefonía VoIP (Voice over IP) se basa prácticamente en el concepto de telefonía tradicional con la peculiaridad de que en este caso la voz en vez de ser transmitida de manera analógica se envía paquetizada a través de internet utilizando la misma infraestructura que los equipos de red, es decir, los teléfonos VoIP son considerados como un equipo más de la red ya que está conectado a través del cable Ethernet al switch como el resto de equipos.

La telefonía VoIP es el futuro de las comunicaciones de voz puesto que es mucho más barata y de mayor calidad que la telefonía convencional.

A continuación vamos a comparar las diferencias entre telefonía VoIP y telefonía convencional.

	Voz sobre IP	Convencional
Alta calidad en llamadas	Sí	Sí
Necesita electricidad	Sí	No
Necesita conexión a Internet	Sí	No
Depende del cableado	No	Si
Cada línea necesita un canal	No	Sí
Alta inmediata	Sí	No
Ofrece movilidad geográfica	Sí	No
Numeración basada en localización física	No	Sí
Compatible con diferentes dispositivos	Sí	No
Altas cuotas mensuales	No	Sí
Llamadas a bajo precio por minuto	Sí	No
Llamadas internacionales económicas	Sí	No

Por estas y otra razones debemos poner especial interés en este tipo de comunicación ya que como hemos comprobado en la tabla anterior la telefonía VoIP ofrece bastantes ventajas frente a la convencional.

Un servicio asociado a las llamadas VoIP muy utilizado por las empresas con centralitas VOIP son las **troncales SIP**. Se trata de una conexión a través de

Internet entre las empresas proveedoras de telefonía VoIP y las empresas de los clientes, en la que se establecen una serie de parámetros de configuración para poder enrutar llamadas a través de esta troncal.

Entre las características principales de troncales SIP encontramos:

- **Número de canales:** por cada canal se puede transmitir una llamada telefónica.
- **Números geográficos:** Las empresas proveedoras de VoIP proporcionan números geográficos de todo el país y el cliente elige que prefijo desea tener en función a la ubicación de la empresa. Es posible realizar la migración de números de telefonía convencional a VoIP a través de estas empresas.
- **Códec de audio y video utilizado:** se establece un acuerdo entre proveedor y cliente sobre las especificaciones de las llamadas para posteriormente configurar la centralita VoIP.

Todas las llamadas generadas por los teléfonos VoIP de una empresa son gestionadas por una centralita PBX VoIP que las enruta a través de una troncal SIP con una operadora de telefonía IP, es decir que la llamada viaja a través de internet desde el origen hasta la operadora, que se encarga de redirigirla hacia telefonía fija, móvil o destinos internacionales.

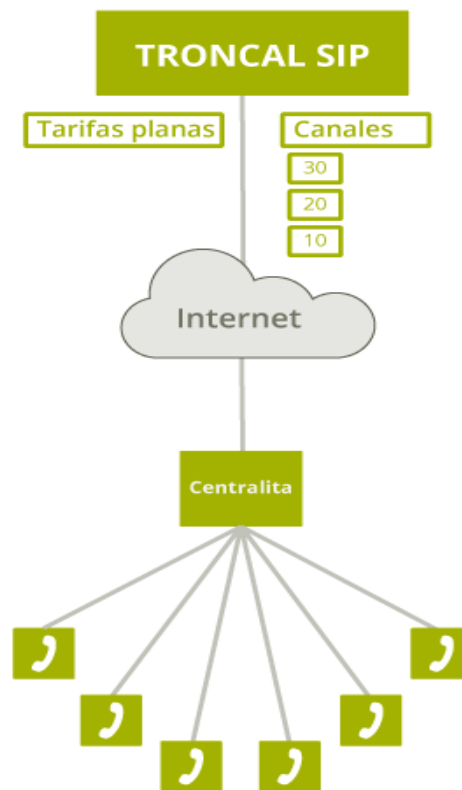


Ilustración 1: Troncal SIP

Hoy en día es posible recibir una llamada local en cualquier parte del mundo, ya que estas viajan a través de Internet, es decir, una empresa contrata una troncal SIP con otra empresa proveedora de telefonía VoIP y estas pueden estar ubicadas en países distintos, a modo de ejemplo, podríamos contratar una troncal SIP en España y nuestra empresa estar en Polonia, no habría ningún problema en que las llamadas generadas en España, hacia nuestros número españoles, sean recibidas desde Polonia ya que nuestra troncal SIP estaría registrada allí.

2.5. Videoconferencia

Un sistema de videoconferencia es básicamente el mismo concepto de una llamada telefónica pero añadiendo video, este servicio requiere un ancho de banda bastante exigente, en función de la calidad de video que queramos transmitir. Para poder realizar una videoconferencia es necesario transmitir audio y video en tiempo real desde el emisor al receptor y viceversa.

Las videoconferencias empezaron a tener éxito a mediados de los ochenta para su uso en el sector empresarial, ya que los equipos existentes en aquella época eran muy costosos.

Poco a poco fueron desarrollándose numerosas herramientas de videoconferencias para ser aplicadas en diferentes áreas como telemedicina o educación a distancia, pero su costo seguía siendo desorbitado para poder ser asumido en el ámbito doméstico. Con la mejora y el abaratamiento de los equipos informáticos orientados al público, cada vez eran más los hogares que contaban con un ordenador con conexión a Internet con el que podían consultar todo tipo de contenido como correo electrónico, prensa y demás.

En 2003 fue creado Skype, la primera aplicación de videoconferencias gratuita y orientada a los usuarios particulares, fue entonces cuando los sistemas de videoconferencias tomaron vital importancia, era necesario equipar a los ordenadores con altavoces y webcam para poder realizar una videoconferencia.

En aquellos años la calidad de las videoconferencias no era nada comparable con los sistemas disponibles hoy en día, el ancho de banda existente y la poca resolución de video de las webcam, ofrecían una calidad de videoconferencia limitada pero más que suficiente para los usuarios domésticos.

Hoy en día se siguen utilizando cada vez más las videoconferencias ya que las personas cada vez están más distanciadas de sus lugares de origen y las empresas están más distribuidas por todo el mundo como compañías multinacionales. Para este tipo de público las videoconferencias son una herramienta clave ya que el usuario doméstico puede ver y escuchar a sus familiares a miles de kilómetros y las empresas pueden realizar reuniones entre diferentes sedes sin la necesidad de viajar a una de ellas.

En la actualidad es muy fácil realizar una videoconferencia gracias al avance de la tecnología y a la notable mejora de los equipos, cualquier Smartphone de gama media del mercado tiene cámara de video capaz de gestionar una videoconferencia a través de una aplicación como Skype.

Para conseguir calidad en las videoconferencias, es necesario ordenadores con capacidad suficiente para gestionar audio y video de alta definición, sin embargo ya existen en el mercado teléfonos VoIP de gama alta que cuentan con pantalla a

color, altavoces y webcam, para poder mantener videoconferencias como si de un ordenador se tratase.

2.6. Ventajas de utilizar Voz sobre IP

- **Ahorro**

El coste de las llamadas a través de VoIP es muy inferior al de las llamadas convencionales, especialmente para llamadas internacionales ya que estas viajan por Internet desde el país de origen hasta el país de destino y allí son convertidas de nuevo en llamadas locales o nacionales.

- **Movilidad**

Basándonos en la misma idea es posible utilizar un número español en cualquier parte del mundo ya que las llamadas viajan por Internet.

- **Funcionalidad**

La voz sobre IP ofrece multitud de funcionalidades relacionadas con desvíos de llamadas, buzón de voz, discriminación horaria y demás parámetros que en la telefonía convencional eran muy limitados y costosos ya que son gestionados desde las compañías telefónicas.

- **Multiplataforma**

Para realizar llamadas VoIP no es necesario disponer de un terminal hardware, simplemente basta con instalar una aplicación softphone en nuestro ordenador o Smartphone y registrarnos con una extensión en la nuestra centralita VoIP. Un caso práctico sería la utilización de telefonía sobre IP en un Call Center, ya que simplemente con la instalación de una aplicación Softphone en el equipo de cada trabajador bastaría para enviar y recibir llamadas a todos los clientes.

3. Materiales y Métodos

En este capítulo se tratará de explicar todo el material utilizado para realzar el proyecto, así como las distintas distribuciones de sistemas operativos empleados.

En primer lugar trataremos de describir todos los elementos de hardware utilizados, posteriormente describiremos toda la parte software y características de configuración.

La inmensa mayoría de los equipos utilizados en este proyecto han sido proporcionados por la Universidad de Jaén a través del departamento de ingeniería telemática de Linares y material disponible en los laboratorios de informática.

3.1. Infraestructura de red

3.1.1. Red LAN

Las redes de área local, generalmente llamadas LAN (Local Area Networks) son redes privadas que operan dentro de un edificio, como una casa, una oficina o una empresa, las redes LAN son utilizadas para conectar ordenadores, impresoras, servidores y todo tipo de equipos con el fin de compartir recursos.

En nuestro caso vamos a utilizar una red LAN para la interconexión de los diferentes elementos de nuestra red , como routers, Switches y teléfonos VoIP entre otros, ya que esta nos proporciona una calidad de comunicación necesaria que otras redes como redes inalámbricas no nos proporcionan. Una de las razones por las cuales la telefonía VoIP se basa en las redes LAN es para poder reutilizar infraestructura ya existente en las empresas, posteriormente analizaremos los terminales de VoIP.

A continuación describiremos todos los dispositivos utilizados para la interconexión de equipos.

- **Router**

Un router es un dispositivo de interconexión de redes informáticas que permite enrutar paquetes entre diferentes tipos de redes, es decir, es el nexo de unión entre

las redes públicas como Internet y redes privadas como redes LAN. Los routers trabajan en las capas dos y tres de la arquitectura OSI.

Para la interconexión de nuestra red local con una red WAN hemos utilizado un router profesional de la marca Cisco, modelo VPN RV120 con capacidad de gestionar VPN(Redes Privadas Virtuales) que describiremos posteriormente.



Ilustración 2: Router

Una funcionalidad a destacar de este router es que cuenta con la función de capturar el tráfico de paquetes para su posterior descarga y análisis, esta opción nos ayuda a comprobar todos los protocolos utilizados por nuestro servidor VOIP.

- **Switch**

Un switch es también un dispositivo de interconexión de redes informáticas pero dentro de una red LAN, es decir que solo trabaja en la capa dos de la arquitectura OSI.

El switch utilizado posee dos funcionalidades muy útiles para nuestro proyecto, las posibilidades de configurar VLANs (LAN Virtuales), esta característica será descrita a continuación. En concreto el switch es el modelo SG300-10P de la marca Cisco.



Ilustración 3: Switch

Otro de los motivos importantes por los cuales hemos elegido este switch es por estar dotado de POE (Power Over Ethernet), es significa que es capaz de

alimentar diferentes dispositivos de red a través del cable Ethernet, de esta manera simplificamos enormemente la instalación de los terminales ya que no sería necesario la doble conexión de adaptador de corriente y conexión de red.

3.1.2. VPN

Una VPN (Virtual Private Network) es una red virtual creada principalmente entre dos routers para transmitir datos de forma segura ya que los router se encuentran geográficamente separados y por lo tanto utilizan redes públicas para su interconexión, en este proyecto será puesto en práctica en la realización de la interconexión entre dos sedes donde configuraremos una VPN e instalaremos un servidor de llamadas VoIP en cada una de las sedes.

Para el funcionamiento de una VPN se utilizan aspectos fundamentales como son la tecnología de túneles, la encriptación y autenticación. [2]

- **Túneles**

Se utiliza la palabra túnel para simbolizar que los datos viajan cifrados desde el origen hasta el destino y son totalmente ilegibles por cualquier equipo fuera del túnel. Las VPN utilizan varios protocolos para asegurar los datos.

- **Encriptación**

En la fase de encriptación se utiliza el protocolo IPSec que trabaja sobre la capa 3 y está encargado de gestionar la privacidad e integridad de los datos enviados.

- **Autenticación**

Para llevar a cabo la fase de autenticación de los datos, en origen se calcula una función Hash (Secure Hash Algorithm) que será comprobada nuevamente en destino para asegurar que los datos no han sido alterados, si la función hash calculada en destino es distinta a la de origen, el mensaje será desechado ya que no cumpliría los estándares de seguridad exigidos por las redes VPN.

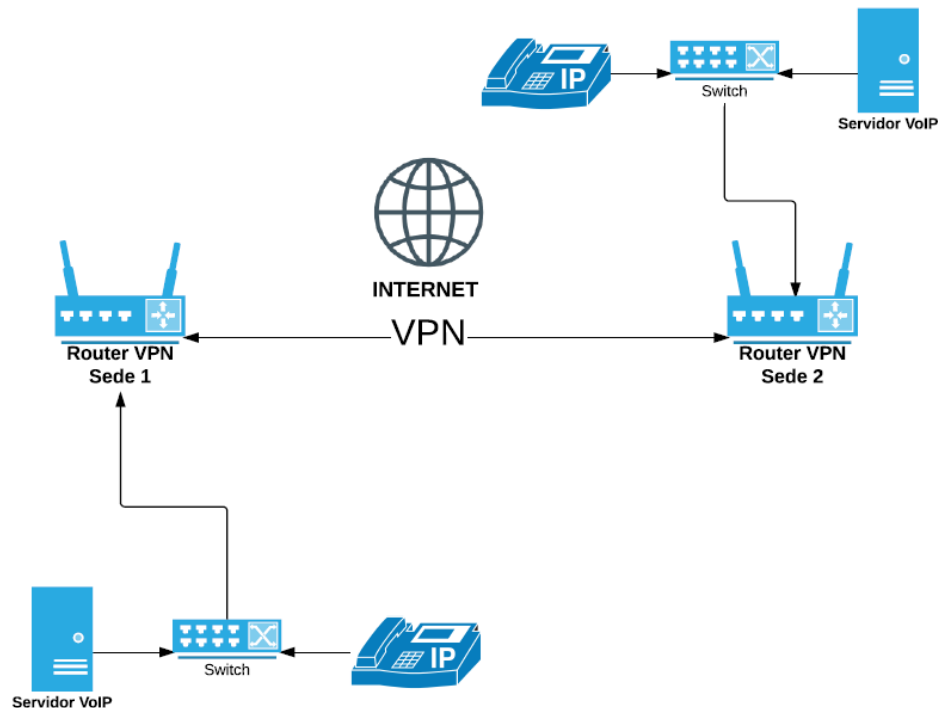


Ilustración 4: Esquema VPN

En nuestro proyecto hemos estudiado solo las VPN punto a punto entre dos router, también es posible configurar un equipo de red remoto como puede ser un ordenador conectado a un router VPN, para ello es necesario instalar un software proporcionado por el fabricante del router que asegure la conexión entre ambos equipos.

3.1.3. VLAN

Las VLAN (Virtual LAN) son redes de área local virtuales desarrolladas con el fin de crear distintas redes locales utilizando la misma infraestructura física. La mayoría de los switches de carácter profesional cuentan con esta función la cual mediante software es capaz de crear redes virtuales.

Para una empresa el uso de VLAN es esencial por motivos de seguridad ya que permite crear diferentes redes LAN de equipos conectados al mismo switch, sin que los miembros de una LAN puedan tener acceso a equipos de otra. De esta manera es posible crear una red para los directivos de la empresa y otra distinta para los empleados sin que se filtre información entre ellas.

Para configurar en un switch una VLAN se adjudican una serie de etiquetas en los puertos ethernet que pertenecen a la VLAN, posteriormente los equipos conectados a esos puertos deben de ser configurados para utilizar dicha red virtual.

En nuestro proyecto vamos a utilizar las VLANs con el fin de proporcionar a la telefonía IP un nivel más de seguridad dentro de la red, ya que todo el tráfico de estos equipos será transmitido a través de la VLAN. La mayoría de los equipos VoIP cuentan con un servidor web para ser configurados y gestionados, sin el uso de las VLAN cualquier equipo de la red tendría acceso a este servicio.

Otro de los motivos de peso para utilizar VLANs es para evitar inundar la red con tráfico de audio y video en este caso, al utilizar las redes virtuales se proporcionan una serie de características como ancho de banda, tratamiento de puertos y protocolos compatibles solo con los equipos que se encuentren dentro la VLAN.

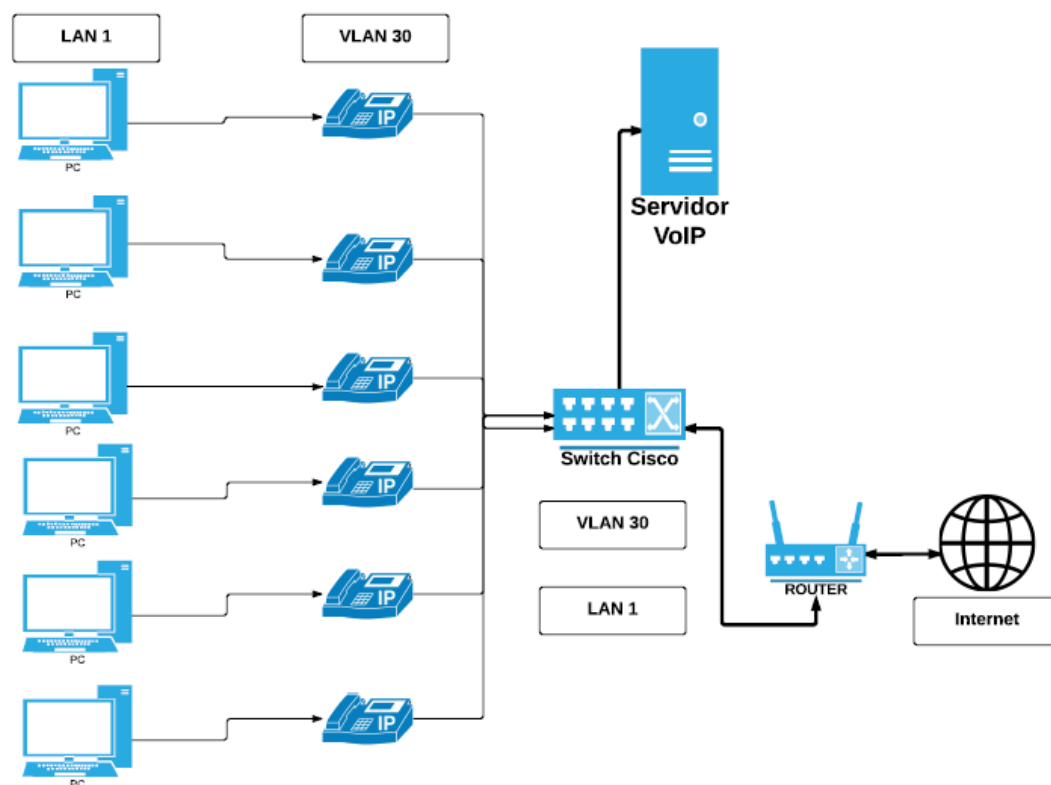


Ilustración 5: Esquema VLAN

3.1.4. QoS

Con la aparición de nuevas aplicaciones que funcionaban sobre las redes IP, fue necesaria la creación de tecnologías como QoS, estas aplicaciones generaban gran cantidad de tráfico multimedia como VoIP, el cual debía ser tratado de forma diferente al resto de tráfico.

Las siglas QoS (Quality of Service) significan en inglés calidad de servicio y se trata de priorizar un tipo de tráfico de datos con respecto a otros, en función de las necesidades que estos tengan. Existen cuatro parámetros principales que determinan a la QoS: [3]

- Ancho de banda
- Retardo (ms)
- Variación del retardo (jitter)
- Pérdida de paquetes

Podemos diferenciar entre tráfico elástico y no elástico, es decir, el elástico puede ajustarse al ancho de banda, retardo, pérdida de paquetes, etcétera como por ejemplo es el caso de servidores FTP, servicios web o correo electrónico, sin embargo el tráfico no elástico tiene requisitos de ancho de banda, jitter y retardo, como ocurre en aplicaciones multimedia.

Los routers y switches empresariales cuentan con QoS ya que es necesario priorizar tráfico no elástico frente a tráfico elástico. Existen dos maneras de configurar la QoS en un router:

- **Servicios Integrados (IntServ)**

Los servicios integrados se basan en el protocolo RSVP (Resource Reservation Protocol) de reserva de recursos, es decir, se reserva una cantidad de ancho de banda de forma dinámica para la transmisión de flujos multimedia con el fin de evitar congestión del resto de la red y asegurar un ancho de banda constante para ese flujo en concreto. Cuando el sistema de reserva de recurso se escala a grandes redes como Internet surgen problemas con la gestión de flujos, señalización,

estados y demás problemas que le restan utilizad a los servicios integrados y potencia el desarrollo de alternativas como servicios diferenciados.

- **Servicios Diferenciados (DiffServ)**

Los servicios diferenciados se basan en el marcado de mensajes para su posterior clasificación en los equipos de comunicación como routers y switches, ya que en función del tipo de marca que estos mensajes lleven serán tratados con distintas prioridades por cada nodo de la red. No es necesario reservar ancho de banda ya que los mensajes se clasifican en distintas categorías en función del tipo de datos que contengan.

Para poder utilizar QoS con servicios diferenciados en Internet es necesario que se pacte con la empresa proveedora de Internet un trato especial a esos mensajes marcados, ya que de lo contrario será tratado como el resto de mensajes.

Las marcar de los paquetes se coloca en un campo reservado para DiffServ dentro de la cabecera de los datagramas, llamado campo DS. Una vez que el mensaje marcado llega al router pasa por las siguientes fases:

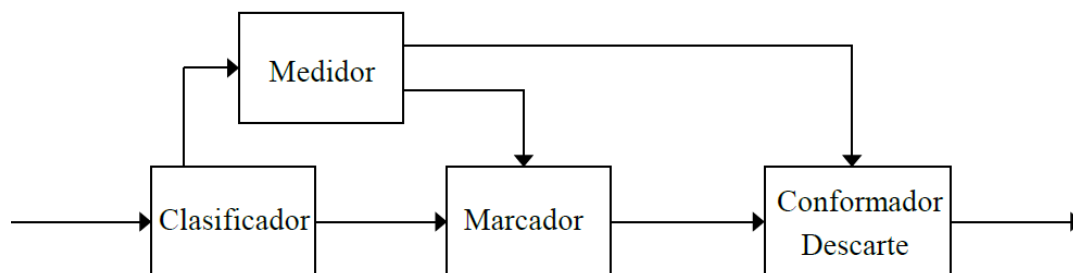


Ilustración 6: Fases de servicios diferenciados

- **Clasificador:** se encarga de separar los mensajes en diferentes clases basándose en el campo DS de la cabecera del mensaje. En función de la prioridad que tenga el mensaje se coloca en el campo DS tres tipos de marcas:
 - **Expedito (tráfico no elástico):** proporciona garantías de caudal, tasa de pedidas, retardo y jitter, esta marca es

utilizada en mensajes que contengan contenido multimedia como llamadas VoIP, videoconferencias o video en streaming.

- **Asegurado** : concede un trato preferente pero sin garantías, existen doce subclases dentro de la marca asegurado
 - **Best Effort (tráfico elástico)**: no concede ninguna garantía, esta marca es utilizada por mensajes que contengan contenido menos prioritario como correos electrónicos, contenido web o FTP.
- **Medidor**: un dato a tener muy en cuenta al usar calidad de servicio es que el tráfico prioritario debe de ser minoritario, por lo tanto la función del medidor es comprobar que cada clase no excede el nivel máximo para esa clase.
 - **Marcador**: la tarea del marcador es volver a marcar los mensajes a un nivel inferior y por lo tanto menos prioritario cuando los mensajes exceden el nivel de esa clase.
 - **Conformador**: el conformador se encarga de enrutar todos los mensajes con una serie reglas de prioridad, es decir, por cada cinco turnos seguidos que sean concedidos a los mensajes marcados como expeditos, se concederá dos turnos a los mensajes asegurados y un turno para los best effort, siguiendo reglas de teoría de colas como Round Robin o Round Robin con pesos.

3.2. Equipos terminales

3.2.1. Teléfonos VoIP

Los teléfono VoIP son terminales hardware físicamente iguales a los teléfonos convencionales analógicos pero con la peculiaridad de poder trabajar en redes LAN ya que posee uno o varios puertos de red rj45 mediante los cuales se conectan al switch como si se tratara de otro equipo de la red, como lo hace un ordenador o una impresora. La calidad de audio de estos terminales es superior a la de los teléfonos convencionales y además cuentan con funcionalidades extra como teclas

programables, desvíos de llamadas o visualizar información adicional en una pantalla digital que llevan incorporada.

Internamente los teléfonos Voip cuenta con un microprocesador capaz de codificar la voz recogida por el micrófono en datos legibles para el resto de equipos de red, el trato recibido por los teléfonos dentro de una LAN es el mismo que el de un ordenador ya que necesita los mismos parámetros de red para operar.

Para la realizar las pruebas en laboratorio hemos utilizamos un modelo de teléfono VoIP de la marca Cisco, disponible como material de prácticas en los laboratorios de informática.

El modelo en concreto era el **Linksys Cisco SPA 512G** que contaba con la posibilidad de configurar una línea SIP manualmente o a través de un propio servidor web, además de poder alimentarlo a través del cable de red con POE "Power over Ethernet". [4]



Ilustración 7: Teléfono VoIP

Este teléfono cuenta con dos puertos de red Gigabit para poder conectar en el segundo puerto un ordenador y así poder utilizar la misma infraestructura de red tanto para el teléfono como para el ordenador como ya vimos en esquemas anteriores.

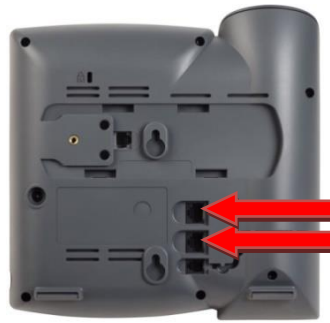


Ilustración 8: Teléfono VoIP 2

Otra de las razones por las cuales hemos utilizado estos terminales ha sido para poder trabajar con un códec de audio compatible con nuestro servidor de llamadas.

- **SoftPhone**

Hoy en día ya no es realmente necesario comprar un terminal hardware para realizar llamadas a través de internet, existen multitud de aplicaciones multiplataforma gratuitas, desarrolladas para cumplir las mismas funciones que un teléfono VoIP. En un ordenador portátil o un smartphone actual podemos instalar estas aplicaciones softphones para poder recibir llamadas y videoconferencias.

Entre las aplicaciones softphone más conocidas encontramos Xlite y Zoiper.

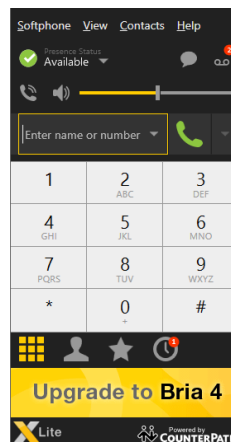


Ilustración 9: SoftPhone Xlite

3.2.2. Gateway FXS

En primer lugar un puerto FXS (Foreign eXchange Subscriber) es el encargado de dar conexión telefónica a teléfonos analógicos a través de un puerto rj11.

Podríamos encontrar este tipo de puertos FXS en las centralitas antiguas analógicas en las que cada teléfono tenía que conectarse para poder recibir llamadas.

En este caso nos centramos en un dispositivo que se encarga de convertir un teléfono analógico en un terminal VoIP con el fin de minimizar gastos y poder adaptar fácilmente una instalación telefónica antigua a VoIP, podríamos decir que un teléfono analógico más un Gateway FXS da como resultado un teléfono VoIP aunque con limitaciones en las funcionalidades y calidad de audio ya que no debemos de olvidar que seguimos utilizando un terminal analógico. [5]



Ilustración 10: Gateway FXS

Para la realización de este proyecto el servicio de informática de la Universidad de Jaén me ha proporcionado un Gateway ATA (Analog Telephone Adaptor) FXS de la marca **GrandStream, modelo HT 502** con el fin de realizar todo tipo de pruebas de calidad y configuración. Este dispositivo cuenta con dos puertos rj11 FXS para conectar dos terminales analógicos y dos puertos rj45 para tener conexión de red con el servidor VoIP.

3.2.3. Gateway FXO

Una interfaz FXO(Foreign eXchange Office) se trata de puerto rj11 por el cual nos conectamos a la compañía proveedora de teléfono e internet (en el caso de conexiones ADSL), es decir es la conexión con la RTB(Red Telefónica Básica).

En este proyecto utilizaremos una Gateway FXO para enrutar llamadas VoIP a una línea analógica con el fin de dar conexión alternativa a troncales SIP y así evitar cortes de servicio telefónico. Utilizaremos este Gateway como troncal de respaldo.

El equipo elegido ha sido un Gateway FXO de la marca GrandStream modelo HT503, dotado con puertos idénticos al modelo HT 502 ya visto, a excepción de uno de los puertos FXS que ha sido sustituido por un puerto FXO y así proporcionar las mismas funcionalidades del terminal anterior más la conexión FXO para la línea analógica.

Utilizar este dispositivo como troncal principal carece de sentido ya que estaríamos convirtiendo una llamada VoIP en analógica de peor calidad, pero para un uso doméstico es una opción muy acertada ya que la inmensa mayoría de domicilios cuentan con conexión analógica y planes de llamadas a líneas fijas y móviles nacionales gratuitas.

Es por ello que la adquisición de este terminal toma importancia, ya que a través de él podemos enrutar llamadas desde una extensión VoIP a nuestra línea analógica y viceversa. Por lo tanto podemos aprovechar el servicio de llamadas gratuitas que nos proporciona nuestro proveedor de Internet y teléfono, esta práctica es conocida como “hop on/ hop off”.

Con este sistema podemos realizar llamadas desde el extranjero utilizando el número local de nuestra empresa o domicilio, como podemos observar en el siguiente esquema de red, tenemos un terminal remoto conectado desde Internet.

Abriendo una serie de puertos en nuestro router y configurando nuestro servidor para que permita el registro desde Internet, podemos registrar una extensión remota para enviar y recibir llamadas como si estuviéramos ubicados en nuestra propia empresa.

Este servicio ofrece gran variedad de posibilidades ya que en ningún momento perderíamos una llamada entrante, puesto que podemos redirigir las llamadas que entren por la línea analógica a una extensión VoIP registrada desde Internet.

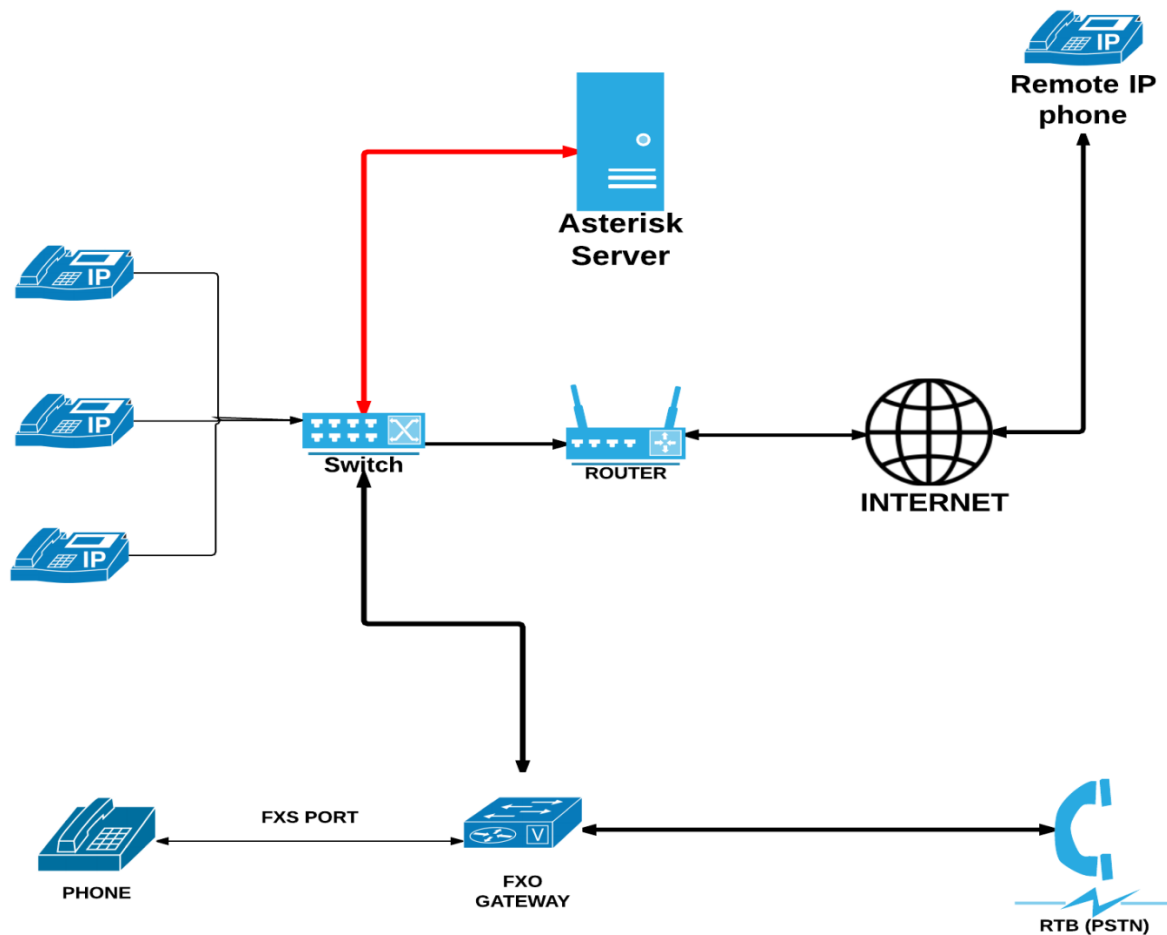


Ilustración 11: Esquema de Red Gateway FXO

3.2.4. Gateway GSM

Un terminal Gateway GSM es prácticamente el mismo concepto que Gateway FXO con la peculiaridad que utiliza la cobertura GSM para enrutar las llamadas, en lugar de un puerto FXO a través de la red telefónica básica como hacía el Gateway anterior.

Para realizar enviar y recibir llamadas a través del Gateway GSM es necesario insertar una tarjeta SIM dentro de este y configurar una serie de parámetros en el servidor para que las llamadas sean enrutadas a través de este dispositivo, esta configuración será descrita en capítulos posteriores.

El modelo común de Gateway GSM suele ser un dispositivo de forma similar a los ya vistos anteriormente pero en este proyecto, hemos utilizado una alternativa muy útil y económica. Haciendo uso de uno de los puertos USB con los que cuenta

la Raspberry Pi, hemos optado por la utilización de un Modem 3G/GMS de la marca **Huawei modedo K3520**.



Ilustración 12: Modem GSM

Este tipo de dispositivos normalmente son proporcionados por las compañías proveedoras de internet móvil para poder disfrutar de conexión de internet móvil (3G) en zonas donde las conexiones de banda ancha como ADSL o FTTx no están disponibles. Estos dispositivos están desarrollados para trabajar con cobertura 3G pero este modelo en concreto también trabaja con cobertura estándar GSM desde los 900 hasta 1800 MHz.

En su interior alberga una ranura para una tarjeta SIM, la cual lleva asociada un número de teléfono móvil, este será utilizado en nuestro servidor VoIP. La utilización de este dispositivo ha sido todo un acierto ya que lo utilizaremos para enrutar llamadas VoIP a GSM y viceversa, de esta forma dotaremos a las centralita de mucha más versatilidad.

3.3. Protocolos de señalización VoIP

3.3.1. SIP

El protocolo SIP (Session Initiation Protocol) fue desarrollado en 1999 para facilitar el control y la señalización en sistemas de telefonía IP. Este protocolo está basado en texto, permite crear, modificar y finalizar sesiones multimedia con uno o varios participantes, entre sus mayores ventajas cabe destacar la simplicidad y consistencia. [6]

Antes de la aparición de este protocolo, los sistemas VoIP utilizaban otros como H.323 o SCCP pero la mayoría eran protocolos privados, de poca capacidad de adaptación en diferentes plataformas y con dificultades a la hora de encriptar la comunicación, como consecuencia de estas incompatibilidades se optó por la estandarización del protocolo SIP. Hoy en día todas las plataformas VoIP y hardware de telefonía son totalmente compatibles a SIP, incluso las de carácter privado como las centralitas CISCO.

Entre las características SIP encontramos:

- Localización de usuarios.
- Negociación de parámetros.
- Disponibilidad del usuario.
- Establecimiento y mantenimiento de sesión.

Es necesario destacar que este protocolo **no** se utiliza para la transmisión de datos como audio o video, sino que gestiona los registros de las diferentes extensiones en el servidor VoIP, de tal manera que la extensión previamente configurada en el servidor es dada de alta para poder recibir llamadas o videollamadas.

SIP es utilizado para establecer una primera comunicación con el servidor VoIP a través del puerto 5060 en TCP o UDP, posteriormente los datos se transmiten entre emisor y receptor a través del protocolo RTP(Real Time Protocol) que veremos más adelante.

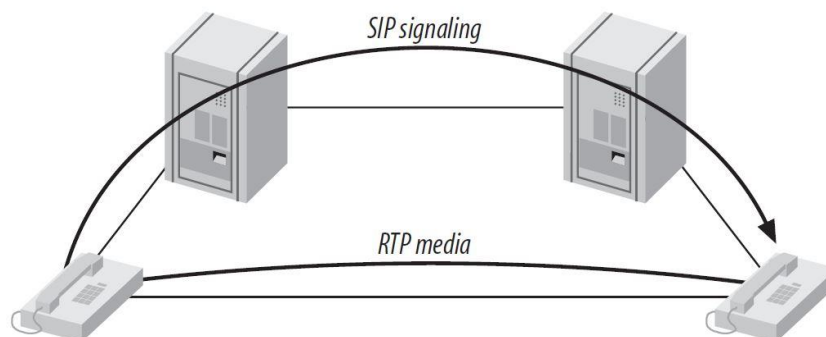


Ilustración 13: Señalización SIP

3.3.2. IAX

El protocolo IAX (Inter-Asterisk eXchange protocol) como su nombre indica fue creado para mejorar la interconexión entre dos centrales Asterisk. Fue desarrollado por Mark Spencer, del que trataremos más adelante. La estructura básica de IAX se fundamenta en la multiplicación de la señalización y del flujo de datos sobre un simple puerto UDP entre dos sistemas

Las características más relevantes de este protocolo son:

- Es un protocolo de código abierto compatible con la mayoría de centralitas VoIP y permite manejar una gran cantidad de códecs.
- Utiliza un único puerto (4569) en UDP para la señalización y comunicación entre puntos finales.
- Es casi transparente a los cortafuegos ya que la voz la transmite in-band y no interfiere a NAT.
- Prioriza el tráfico de voz frente al tráfico de datos haciéndolo imprescindible en conexiones lentas.

El uso de este protocolo es altamente recomendable cuando enviamos gran número de llamadas entre dos centralitas VoIP, ya que IAX unifica múltiples sesiones en un solo flujo de datos y así el consumo de ancho de banda disminuye notablemente.

Un escenario idóneo para utilizar este protocolo sería en la comunicación entre dos sedes de la misma empresa, ya que todas las llamadas entre ellas transcurrirían a través de una troncal IAX.

Utilizando IAX la información correspondiente a las llamadas se envía en un datagrama de manera que se consigue disminuir la latencia y el ancho de banda.

El protocolo IAX ha sido mejorado y renombrado como IAX2 , este encapsula todos los tráficos en un datagrama y con una única cabecera.

3.3.3. Diferencias entre SIP e IAX

- **Ancho de banda.**

IAX utiliza un menor ancho de banda que SIP ya que los mensajes son codificados de forma binaria mientras que en SIP son mensajes en formato texto. Asimismo, IAX intenta reducir al máximo la información de las cabeceras de los mensajes reduciendo también el ancho de banda.

- **NAT**

En IAX la señalización y los datos viajan conjuntamente con lo cual se evitan los problemas de NAT que frecuentemente aparecen en SIP. En SIP la señalización y los datos viajan de manera separada y por eso aparecen problemas de NAT en el flujo de audio cuando este flujo debe superar los routers y firewalls. SIP suele necesitar un servidor STUN para estos problemas.

- **Estandarización y uso**

SIP es un protocolo estandarizado por la IETF hace bastante tiempo y que es ampliamente implementado por todos los fabricantes de equipos y software. IAX está aún siendo estandarizado y es por ello que no se encuentra todavía en muchos dispositivos existentes en el mercado.

- **Utilización de puertos**

IAX utiliza un solo puerto (4569) para mandar la información de señalización y los datos de todas sus llamadas. Para ello utiliza un mecanismo de multiplexión o “trunking”. SIP, sin embargo utiliza un puerto (5060) para señalización y 2 puertos RTP por cada conexión de audio (como mínimo 3 puertos). Por ejemplo para 100 llamadas simultáneas con SIP se usarían 200 puertos (RTP) más el puerto 5060 de señalización. IAX utilizaría sólo un puerto para todo (4569). [7]

- **Flujo de audio al utilizar un servidor**

En SIP si utilizamos un servidor la señalización de control pasa siempre por el servidor pero la información de audio (flujo RTP) puede viajar extremo a extremo sin

tener que pasar necesariamente por el servidor SIP. En IAX al viajar la señalización y los datos de forma conjunta todo el tráfico de audio debe pasar obligatoriamente por el servidor IAX. Esto produce un aumento en el uso del ancho de banda que deben soportar los servidores IAX sobre todo cuando hay muchas llamadas simultáneas.

3.4. Protocolos de transmisión

3.4.1. RTP

El protocolo RTP (Real Time Protocol) es el encargado de transportar tanto el audio como el video en tiempo real. Utiliza UDP como protocolo de transporte ya que el uso de TCP y su control de flujo darían lugar a un retardo elevado durante la comunicación a causa de las retransmisiones. [8]

Este protocolo lleva a cabo un control de secuencias en la reproducción de paquetes que envía utilizando marcas de tiempo, chequeo del jitter y otros parámetros que aseguran que los datos se van a reproducir en destino de manera sincronizada.

3.4.2. RTCP

El protocolo RTCP (Real Time Control Protocol) es el encargado de monitorizar el flujo de paquetes RTP, obteniendo estadísticas sobre el jitter, latencia, pérdida de paquetes y demás parámetros de control relacionados con la calidad de servicio.

RTCP está muy asociado a RTP ya que controla la calidad de todos los datos que se transmiten a través de RTP, es decir que antes de que comience transmisión de datos entre dos terminales, este protocolo se encarga de gestionar parámetros como el códec de audio o video que puede manejar ambos terminales.

3.5. Códecs

La palabra códec es la abreviatura de codificador-decodificador, cumple la función de digitalizar información analógica de tal manera que esta es codificada en origen y descodificada en destino donde es interpretada.

Los códecs juegan un papel muy importante en VoIP ya que si recordamos lo mencionado en capítulos anteriores dejamos de enviar voz analógica a través las infraestructuras de comunicación para enviar datos digitalmente y estos se pueden comprimir y descomprimir sin problema.

Con la utilización de códec existe un equilibrio entre eficiencia y calidad ya que se utilizan diferentes tipos de códecs que veremos a continuación, entre los que encontramos códecs con y sin compresión y también con y sin pérdida, la compresión de los datos nos implica pérdida de calidad pero si es necesario gran capacidad de computo para gestionarlo. [9]

Este proceso de conversión analógico digital o modulación por impulsos codificados (PCM) se realiza mediante tres pasos:

1º) Muestreo:

Consiste en tomar valores instantáneos de una señal analógica, a intervalos de tiempo iguales.

2º) Cuantificación:

Es el proceso en el cual se asignan valores discretos a las amplitudes de las muestras obtenidas en el proceso de muestreo.

3º) Codificación:

Es el proceso mediante el cual se representa una muestra cuantificada, mediante una sucesión de 1 y 0, es decir, mediante un número binario.

Una vez que los datos codificados llegan a destino se utiliza un proceso de decodificación mediante el cual se reconstruyen las muestras.

A continuación describiremos los distintos tipos códecs de audio y video utilizados en este proyecto.

3.5.1. Códecs de voz

En este apartado describiremos brevemente los diferentes códecs de audio utilizados en este proyecto y haremos una pequeña comparación entre ellos.

Actualmente en el sector VoIP se utilizan multitud de códecs de audio, cada uno de ellos con características que los diferencian como ancho de banda necesario para transmitirlo, compresión o no de la información y demás. En este proyecto se han elegido tres códecs en concreto por ser los más utilizados e importantes, ya que muchos de los que encontramos en el mercado son solo utilizados en sistemas VoIP embebidos como es el caso del códec utilizado por Cisco en sus centralitas (Call Manager) y que hacen incompatible la utilización de otro hardware que no sea Cisco.

- **G.711**

Este códec es bien conocido puesto que se utiliza en la red telefónica básica por sus características de simpleza y facilidad de ser procesado en diferentes tipos de hardware.

Existen dos variantes dentro de este códec, ley A (a-law) mayoritariamente utilizado en Europa y ley μ (u-law) utilizada en América y Asia.

Las características principales de este códec son que utiliza modulación por codificación de impulsos (PCM), un ancho de palabra de 8 bits y una tasa de 8000 muestras por segundo, realizando los cálculos decimos que en un segundo se transmiten 64000 bits, es decir, ocupa un ancho de banda de 64 Kbps y añadiéndole cabeceras llega hasta los 82-83 Kbps.

Este códec funciona sin compresión de datos, por lo tanto el uso que hace de la CPU es casi nulo ya que no requiere la tarea de comprimir y descomprimir el audio, este será el códec principalmente utilizado en este proyecto puesto que contamos con un hardware de bajo rendimiento, otro de los motivos es que este G.711 está libre de patentes y puede ser utilizado de manera gratuita sin la necesidad de comprar licencias de uso.

- **G.729**

Este códec fue diseñado para su uso en instalaciones con conexiones a Internet lentas o con un ancho de banda limitado como por pueden ser ADSL.

Las características principales de este códec son que utiliza un ancho de banda inferior a G.711 pero con un mayor consumo de procesador ya que este si utiliza compresión.

El ancho de banda necesario para transmitir voz en G.729 es tan solo de 8 Kbps es por tanto que en conexiones lentas como ADSL (10Mbps bajada-1Mbps subida) puede ser una solución idónea para no tener problemas al realizar llamadas, pero en contraposición necesitamos instalar hardware más potente en ambos extremos para el procesamiento de este códec.

Otra de las características más relevantes de este códec es que no transmite el silencio a diferencia de G.711 que si lo hace. Por último cabe destacar para la utilización de este códec para un uso comercial es necesario el pago de licencias de uso.

- **GSM**

Este códec es muy conocido por los usuarios ya que es el utilizado en la telefonía móvil desde hace más de dos décadas pero con la aparición de la telefonía VoIP dejó de utilizarse puesto que no ofrecía una calidad de audio comparable con el resto de códec descritos anteriormente.

La razón por la cual citamos este códec es porque posteriormente utilizaremos la cobertura GSM para enrutar llamadas a través de un modem 3g/GSM.

Las características más importantes de este códec es la gran expansión que ha tenido desde su creación y la alta compatibilidad con multitud de productos como teléfonos móviles, Smartphone, tablets o PDA's, es por ello que hace muy versátil su uso. Como información técnica, este códec utiliza un ancho de banda de 13 Kbps sin compresión de voz y por tanto no es necesario un gran procesador para gestionarlo.

Diferencias entre los distintos tipos de códecs

A continuación analizaremos una tabla donde se mostrarán las diferencias entre algunos tipos de códecs.

Códec	Bit rate (Kb/s)	Compresión	Licencia de uso	Observaciones
G.711	64 Kb/s	No	No	Variantes a-law y μ -law
G.729	8 Kb/s	Si	Si	Preferible para conexiones lentas
GSM	13 Kb/s	No	No	Poca calidad de audio

3.5.2. Códecs de video

Al igual que en los códecs de audio, los de video también son codificados en origen y decodificados en destino pero con bastantes diferencias ya que la gestión de video es bastante más pesada en cuanto a procesamiento de la información que la de audio.

La inmensa mayoría de los códecs de video viajan comprimidos aunque se pierda cierta calidad, de lo contrario sería casi imposible gestionar con agilidad una red orientada a videoconferencias, el objetivo es lograr transmitir la máxima calidad de video ocupando el mínimo ancho de banda posible. [10]

El uso principal de los servidores VoIP es la realización de llamadas o conferencias, no videoconferencias, pero cada vez los terminales que utilizan los usuarios finales son más sofisticados y cuentan con pantallas a color, cámaras de video de alta calidad, micrófonos y altavoces incorporados, por lo tanto desde la parte técnica debemos proporcionarles este servicio.

A continuación describiremos los principales códecs de video utilizados por las centralitas VoIP, son los estandarizados por la ITU (Union Internacional de las Telecomunicaciones), estándar H.26x.

- **H.263**

Es el primer códec de video comprimido, desarrollado en 1996 con el propósito de dar servicio a sistemas multimedia basados en IP y mensajes multimedia (MMS). Este códec podía ser transmitido a través de la línea telefónica ya que sólo requería de 64 Kbps. Las resoluciones de video que podía gestionar H.263 eran de 640×480.

- **H.264**

Es el códec sucesor a H.263, desarrollado en 2003 por MPEG (Moving Picture Experts Group), contaba con notables mejoras en muchos aspectos como la calidad de resolución, la alta capacidad de compresión, compensación del movimiento. El códec H.264 utiliza la mitad de bit rate con misma calidad de video que H.263, fue diseñado en un principio para ofrecer servicios de televisión por cable y streaming.

Este códec necesita un ancho de banda comprendido entre 1.5 y 3,2 Mbit/s para ser transmitido en función a la resolución que utilice. Es compatible con resoluciones de alta calidad como 1280 x 720 y 1920 x 1080.

Otras de las ventajas es este códec es que solo transmite los elementos que estén en movimiento, de esta manera disminuye bastante la información a enviar.



Ilustración 14: Características del códec H.264

Hoy en día la utilización de este códec está en auge ya que es posible mantener una videoconferencia desde el propio ordenador portátil o desde un Smartphone. Plataformas como por ejemplo Skype permiten realizar videoconferencias gracias a la utilización de códec de video como este, aunque Skype, al igual que Cisco utilice su propio códec privado.

3.6. Equipos de baja capacidad de procesamiento

La razón por la cual hemos optado por este tipo de placas es para comprobar cuál es el máximo rendimiento que son capaces de ofrecer, beneficiándonos de su bajo consumo eléctrico y su reducido coste en el mercado.

Raspberry Pi

Actualmente y gracias a la importancia tan grande que tiene los proyectos sobre IoT(Internet of Things) “El Internet de las cosas” es difícil encontrar una persona relacionada con la tecnología que desconozca la existencia de tales placas.

En el mercado actual es fácil encontrar muchos tipos de placas diseñadas para sistemas embebidos como puede ser lavadoras, hornos, televisores, máquinas de café y demás tipos de electrodomésticos, pero estas placas están desarrolladas expresamente para gestionar un aparato electrónico en concreto, incluso el famoso Arduino, a excepción de la versión YUM, se encuentra también en el área de las placas electrónicas las cuales realizan perfectamente tareas concretas de domótica y automatismos.

Sin embargo para este proyecto es necesario un nivel de cómputo mayor ya que la placa elegida es capaz de trabajar en la capa de aplicación, y gestionar una versión adaptada del sistema operativo Linux.

A continuación describiremosmos la historia de la placa Raspberry Pi.

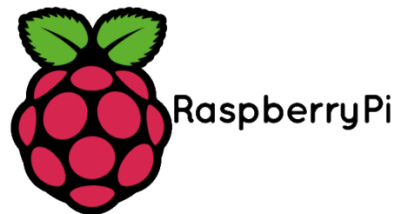


Ilustración 15: Logo Raspberry Pi

Raspberry Pi nace como fruto de un proyecto desarrollado por la Universidad de Cambridge en el año 2006 con la idea de fomentar la ciencia de la computación para niños de edades tempranas. [11]

Posteriormente en el año 2009 fue fundada la fundación Raspberry Pi con el fin de crear ordenadores portables y muy baratos que permitieran a los niños utilizarlos sin ningún tipo de peligro. Con ello se quería conseguir que desde pequeños se involucraran en las nuevas tecnologías y aprendieran a programar lenguajes de programación como Python o C++ entre otros.



Ilustración 16: Eben Upton

Fue entonces cuando **Eben Upton**, un antiguo trabajador de la empresa de microprocesadores Broadcom, desarrolló la arquitectura del software y todo el hardware de la Raspberry Pi. Una de las características más innovadora de esta placa fue la introducción de un microprocesador de arquitectura ARM, el cual era

compatible con los sistemas operativos Linux, Debian (Raspbian), RISC OS y Arch Linux adaptados a Raspberry Pi.

Desde la creación de la primera placa de Raspberry Pi en el año 2011 han sido desarrolladas diferentes versiones mejoradas, en este proyecto nos centraremos en dos modelos en concreto ya que son las placas utilizadas en laboratorio. [12]

Entre los modelos más importantes de Raspberry Pi encontramos:

- En Agosto del año 2013 presentaron el modelo A (rev. 1) con un procesador ARM de 700 MHz ,256 MB de RAM, un puerto USB y no contaba con puerto de red.



Ilustración 17: Raspberry Pi 1 modelo A

- En Febrero del 2014 sacaron a la venta el modelo B (rev. 1) con el mismo procesador de 700 MHz pero con 512 MB de memoria RAM y que ya contaba con un puerto de red y dos puertos USB que dotaba a la placa de un gran número de funcionalidades de comunicación aunque el consumo energético necesitaba mejoras ya que era de 3.5 W, (700 mA).



Ilustración 18: Raspberry Pi 1 modelo B

- Durante el mismo año 2014 hicieron mejoras con el consumo energético y presentaron el modelo B+ (rev.1) idéntico al anterior con la diferencia de que se consumía solo 3.0 W, (600 mA). Esta es una de las placas utilizadas en el proyecto.
- Finalmente en Febrero del año 2015 presentaron el modelo B (rev. 2) que ya contaba con un procesador ARM Cortex A7 totalmente nuevo y de 900MHz, 1GB de memoria RAM y 4 puertos USB 2.0.



Ilustración 19: Raspberry Pi 2 Modelo B

- Recientemente han presentado el último modelo, llamado Raspberry Pi 3 con Bluetooth y Wifi integrado en la placa, no entraremos en más detalles ya que no será utilizado en la realización de este proyecto.

A continuación mostraremos una tabla indicando las diferencias entre las dos placas utilizadas en este proyecto. [13]

	Raspberry Pi 1 B+	Raspberry Pi 2 B
CPU	700 MHz ARM 11	900 MHz ARM Cortex A7

GPU	Broadcom VideoCore IV	Broadcom VideoCore IV
RAM	512 MB	1 GB
Puertos USB	2 puertos USB 2.0	4 puertos USB 2.0
Tipo de socket	SD	Micro SD
Puerto Ethernet	Si	Si
Consumo energético	600 mA, (3.0 W)	800 mA, (4.0 W)

Además de los componentes anteriormente nombrados las dos placas que utilizamos en el proyecto cuentan con un puerto HDMI para la conexión de una pantalla para poder gestionarlas de forma gráfica, en nuestro caso no utilizaremos este conector ya que toda la gestión la realizaremos por medio de un servidor web a través del puerto de red.

En el socket para las tarjetas de memoria hemos introducido una tarjeta SD y Micro SD respectivamente, ya que esta es la componente que utiliza la Raspberry como almacenamiento para el sistema operativo y demás ficheros de configuración necesarios.

Hemos utilizado dos tarjetas de memoria de la marca Kingston de 8 GB y de clase 10 para poder transmitir información a 10 MB/s, en nuestro caso estas son las sustitutas del disco duro en un ordenador convencional y no queríamos encontrar limitaciones en procesos de lectura y escritura.

3.7. Servidores VoIP (Software)

3.7.1. Asterisk

Asterisk es una aplicación para controlar y gestionar comunicaciones de cualquier tipo, ya sean analógicas o digitales, mediante todo tipo de protocolos de transmisión de audio y video.



Ilustración 20: Asterisk Logo

Es una aplicación OpenSource basada en licencia GPL y por lo tanto disfruta de las ventajas de contar con un equipo de desarrolladores que dan soporte de manera altruista a la comunidad de usuarios. Asterisk es utilizado por profesionales para desarrollar sistemas de comunicación de gran calidad, seguridad y versatilidad.

Asterisk fue creada en 1999 por **Mark Spencer**, un ingeniero informático estadounidense, desarrollada con el fin de donarla a la comunidad para poder hacerle frente a otras soluciones con software privado. Mark Spencer fue el fundador de la empresa **Digium**, especializada en el desarrollo y fabricación de hardware y software de comunicaciones y telefonía. La empresa vende todo tipo de centralitas hardware con el software de Asterisk integrado.



Ilustración 21: Mark Spencer

Poco a poco, esta aplicación se ha convertido en la evolución de las tradicionales centralitas analógicas y digitales permitiendo también integración con la tecnología más actual VoIP. Asterisk se convierte así en el mejor, más completo, avanzado y económico sistema de comunicaciones existente en la actualidad. [14]

Otra de las características a destacar es la gran compatibilidad que tiene con sistemas operativos Linux y protocolos como SIP, H323 e IAX. Otra de las ventajas más interesantes es que permite gestionar comunicaciones tradicionales (analógicas, digitales y móviles) con comunicaciones IP mediante el uso de protocolos estándar de Voip.

Anteriormente solo existían centralitas analógicas, las cuales necesitaban ser configuradas por personal altamente cualificado ya que requerían de muchos conocimientos en esta área.

Posteriormente aparecieron las primeras centralitas VoIP con software privativo y a precios desorbitados, que solamente eran compatibles con equipos de las mismas marcas ya que utilizaban códecs de audio propios, como es el caso de las centralitas VoIP CISCO.

Gracias al desarrollo de Asterisk hoy en día es posible la intercomunicación de voz y video entre equipos a un costo muy reducido o incluso de manera gratuita.

Asterisk no es solo un programa de gestión de llamadas y videollamadas, en las primeras versiones contaba con menos funcionalidades pero hoy en día es posible configurar infinidad de funciones como:

- Buzón de voz.
- Conferencias.
- Grupos de timbrado.
- Condiciones de tiempo (vacaciones, discriminación horaria).
- Colas de llamadas.
- Prioridad de llamadas.
- Llamadas aparcadas.
- IVR (Interactive Voice Response) teleoperadora virtual.

- Anuncios.
- Listas negras (exclusión de determinados números).
- Reenvío de llamadas.
- Configuración por lotes de extensiones
- Grabación de llamadas

La única limitación que encontramos en Asterisk es que su configuración se realiza a través de línea de comandos en el terminal de Linux, modificando una serie de archivos necesarios para tareas como creación de extensiones, creación de troncales, rutas entrantes y salientes y demás. Podríamos decir que la gestión de Asterisk a través del terminal sería Asterisk puro aunque existen bastantes plantillas para la automatización de estas tareas, sin embargo existen diferentes soluciones disponibles con un interfaz gráfico amigable y una gestión a través de un servidor web integrado en la distribución.

A continuación pasaremos a describir soluciones de PBX VoIP con interfaces gráficas compatibles con nuestra Raspberry Pi 1 y 2.

3.7.2. FreePBX

FreePBX es una interfaz web que facilita la interoperabilidad del usuario con el sistema Voip Asterisk, de esta manera disminuye notablemente la complejidad de uso, es una solución idónea para usuarios y administradores que no estén muy familiarizados con la programación de Asterisk puro, este software corre también bajo licencia GPL. [16]



Ilustración 22: FreePBX Logo

Todas las configuraciones que realiza el usuario a través de la interfaz web tienen efecto sobre los ficheros de Asterisk anteriormente mencionados, es posible realizar todas las funciones que tiene Asterisk pero de manera gráfica.

Entre las características más llamativas incluidas en FreePBX podemos encontrar funciones instaladas por defecto que no encontramos en Asterisk puro, como por ejemplo un servicio de actualización automática de los repositorios oficiales de FreePBX , backups, un servicio de seguridad llamado Fail2Ban que actúa como firewall a posibles intrusos en esta PBX o el servicio de mail-to-fax o fax-to-mail entre otros. Es por esto que hacer de Freepbx un producto mucho más versátil que el simple Asterisk, lo que sí es cierto que las empresas que solo demanden un sistema telefónico limpio sin programas adicionales con Asterisk tienen la solución, ya que también es necesario destacar que cuantos más servicios tengamos en conectados a Internet más riesgo corremos de ser atacados.

A continuación veremos una distribución de FreePBX compatible con Raspberry Pi 1 y 2.

- **RasPBX**

RasPBX es una distribución de Linux, en concreto Debian Jessie .adaptada a Raspberry Pi y llamada Raspbian y que adicionalmente viene con una versión de FreePBX (Asterisk 11 y FreePBX 13) adaptada a procesadores ARM como el de la Raspberry, en resumen podríamos decir que RasPBX es un producto magnífico por la infinidad de configuraciones que se pueden realizar y sin olvidar que estamos tratando con un hardware muy limitado.

Otro aspecto a destacar de esta distribución es que es compatible con la configuración de un modem 3G/GSM para su uso como troncal a la hora de enrutar llamadas.

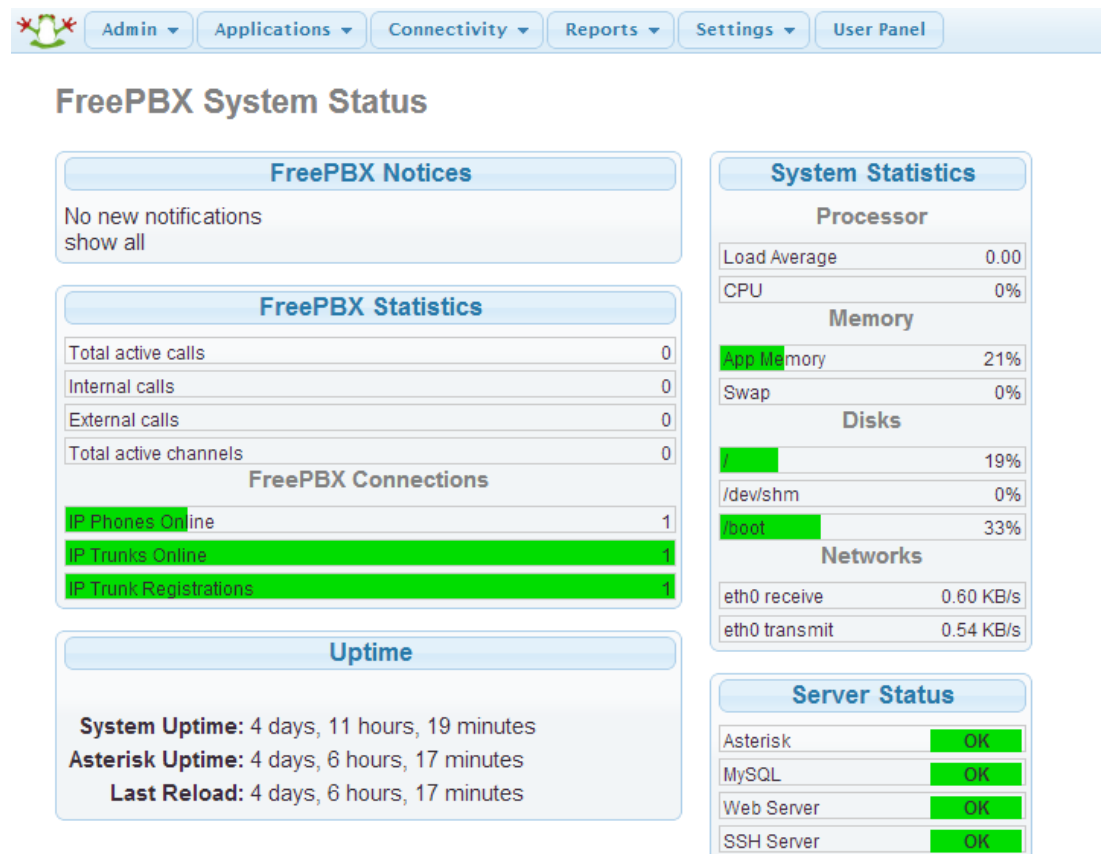


Ilustración 23: RasPBX

A la finalización de este proyecto se añadirá un anexo en el que se explica cómo instalar esta distribución en la Raspberry Pi.

- **Elastix**

Elastix es un software de código abierto desarrollado en Ecuador para proporcionar comunicaciones unificadas, es el mismo concepto de FreePBX pero ofreciendo muchos más servicios como:

- Correo electrónico.
- Videoconferencias.
- Mensajería instantánea.
- Gestor de Faxes.
- Gran variedad de Addons (servicios adicionales instalables).
- Interfaz de detección de hardware de telefonía.
- Provisionamiento de teléfonos a través de la interfaz web.

- FOP (Flash Operator Panel) muy útil e intuitivo para monitorizar llamadas.
- Y por supuesto servidor de llamadas Asterisk.



Ilustración 24: Elastix Logo

Elastix es una solución muy completa para dar servicio a lo que sus creadores llaman comunicaciones unificadas, de tal modo que en un mismo equipo disponemos de todos los servicios necesarios para nuestra empresa, como es lógico esta política tiene sus seguidores que piensan que es la solución idónea y por supuesto otras personas que no están a favor ya que no es recomendable tener todos los servicios de una empresa en la misma máquina ya que si por desgracia esta falla todos los servicios como correo electrónico, servidor de llamadas y demás, dejarían de funcionar, también por motivos de seguridad. [17]

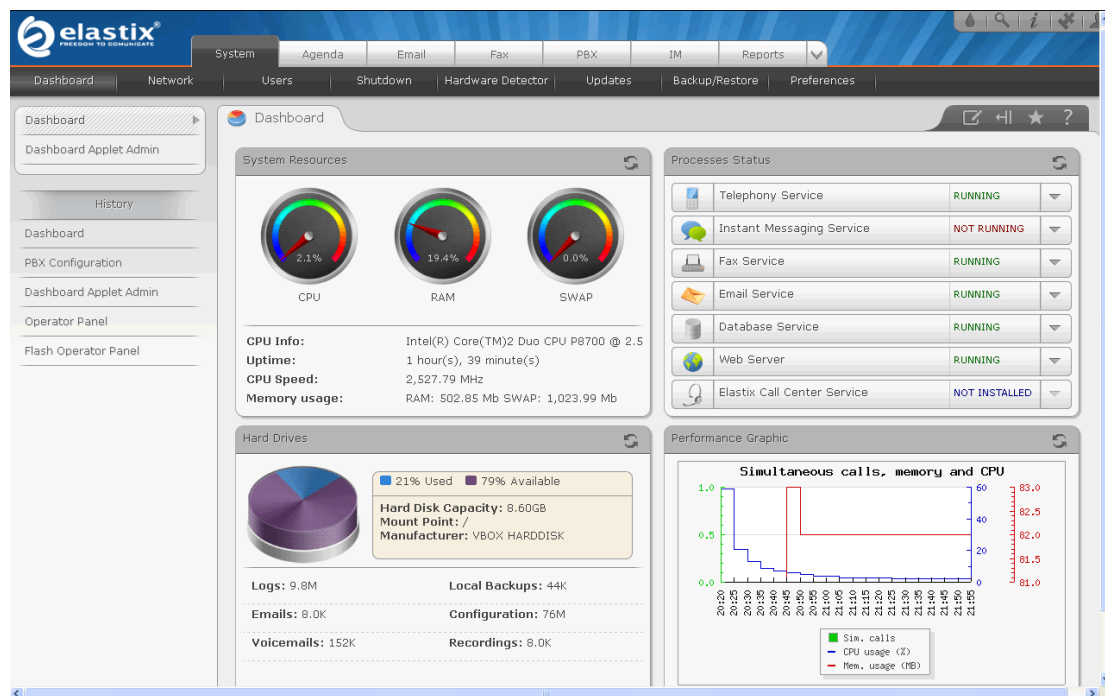


Ilustración 25: Elastix

A continuación veremos una distribución de Elastix compatible hasta la fecha solo con la versión 1 de Raspberry Pi.

- **µElastix**

µElastix fue desarrollada en 2014 por el equipo de Elastix con el lema de ofrecer grandes soluciones a pequeños dispositivos. En este proyecto vamos a analizar cuantas llamadas simultáneas puede gestionar la Raspberry Pi con esta versión de Elastix.

3.8. Herramientas de análisis

3.8.1. Wireshark

Wireshark (tiburón de los cables) anteriormente conocido como **Ethereal**, es un programa analizador de protocolos de red utilizado para multitud de funciones relacionadas con el tráfico entre equipos.

Este programa cuenta con un apartado muy útil para nuestro proyecto ya que es capaz de analizar protocolos anteriormente mencionados como SIP, IAX, RTP, RTCP, UDP, TCP y demás funciones relacionadas con la telefonía voz sobre IP.

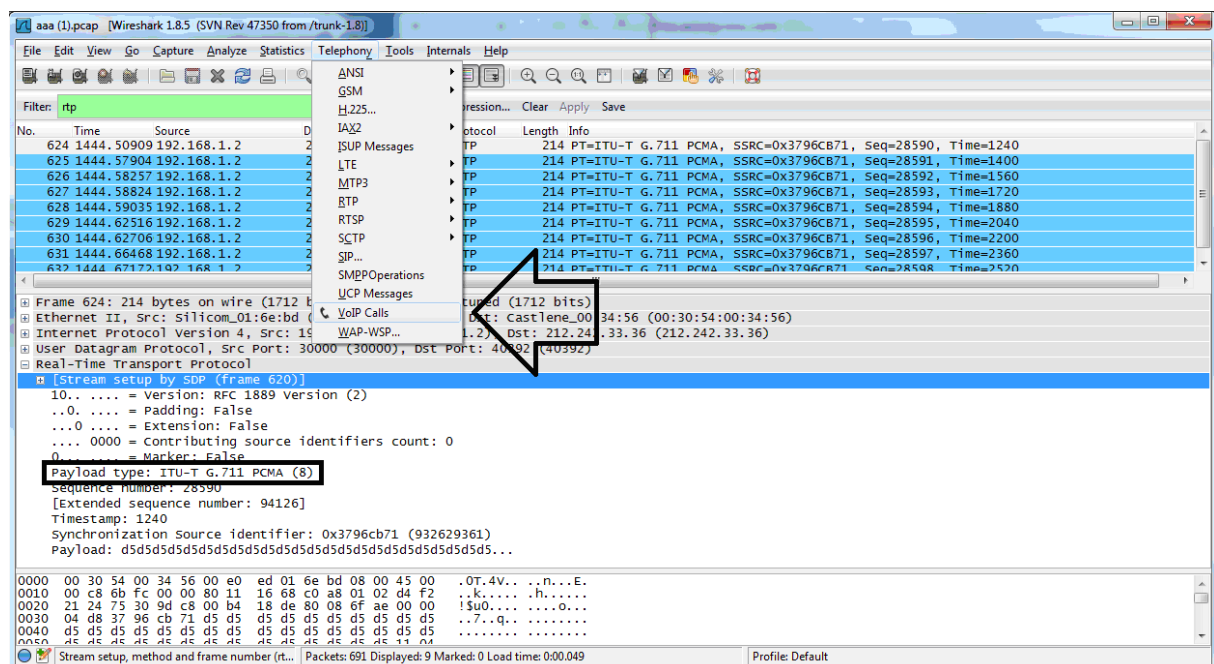


Ilustración 26: Wireshark

Gracias a WireShark es posible detectar muchos problemas asociados a retardos, paquetes incompletos, tasa de transferencia de mensajes y demás. En la pestaña Telephony podemos analizar el origen y destino de las llamadas realizadas por las diferentes extensiones. [18]

Un aspecto muy importante para el correcto funcionamiento de WireShark es que debe de estar instalado en una máquina intermedia entre el servidor VoIP y los terminales, para analizar todo el tráfico que se transmita a través de ellos.

En nuestro caso contamos con un router de gama media/alta con capacidad de capturar todo el tráfico que lo atraviere y posteriormente generar un archivo con formato legible por WireShark, es decir , una vez que hallamos realizado las pruebas pertinentes descargaremos el archivo.cap del router y lo analizaremos desde nuestro equipo de manera local.

En el interior de estos archivos encontramos información referente al tipo de códec utilizado en cada llamada, tamaño de los paquetes enviados entre terminales y servidor, registros SIP (REGISTER, INVITE, ACK y BYE), duración de la llamada e incluso es posible decodificar la llamada establecida entre dos extensiones para su posterior escucha.

Otra de las opciones que hemos utilizado para poder analizar el tráfico VoIP fue configurar el switch con la función de mirroring (reflejar tráfico), en uno de los puertos del switch conectamos nuestro servidor Voip y en otro nuestro ordenador, de tal manera que todo el tráfico que atraviere el puerto del servidor nos lo reenviara a nuestro ordenador para su análisis en línea desde WireShark, de esta manera no sobrecargamos el router/switch ya que sólo clona tráfico y no lo captura.

4. Resultados

A continuación describimos todas las pruebas realizadas en laboratorio correspondientes a la instalación y configuración de los servidores VoIP en Raspberry Pi.

Para poder realizar comparativas entre las diferentes placas y servidores software hemos instalado la versión de RasPBX en Raspberry Pi 2 ver. B y uElastix

en Raspberry Pi 1 ver. B+ de esta manera podemos comparar los ventajas e inconvenientes de cada versión y de cada placa. Otras de la razones ha sido para comprobar la compatibilidad entre ambas.

4.1. Configuración y puesta en marcha de RasPBX en Raspberry Pi

En primer lugar quería destacar que la versión FreePBX para Raspberry Pi llamada RasPBX es compatible con todas las versiones de Raspberry Pi.

FreePBX posee una comunidad muy activa que constantemente proporcionan mejoras a las versiones existentes, en este proyecto vamos a utilizar la versión estable más reciente de RasPBX (raspbx-06-03-2016.zip) de principios de Marzo de este año. A diferencia de Uelastix , la instalación de RasPBX es prácticamente automática.

Primero debemos descargar la distribución de la página oficial de FreePBX y descomprimirla, donde obtendremos la imagen

raspbx-06-03-2016.img.

Posteriormente formatearemos la tarjeta SD de al menos 8Gb con la aplicación **SDFormatter V4.0.**

A continuación utilizaremos la aplicación **Win32 Disk Manager**, donde nos pide que indiquemos la ubicación de la imagen a instalar en la SD y la letra que Windows le ha asignado a la tarjeta de memoria.

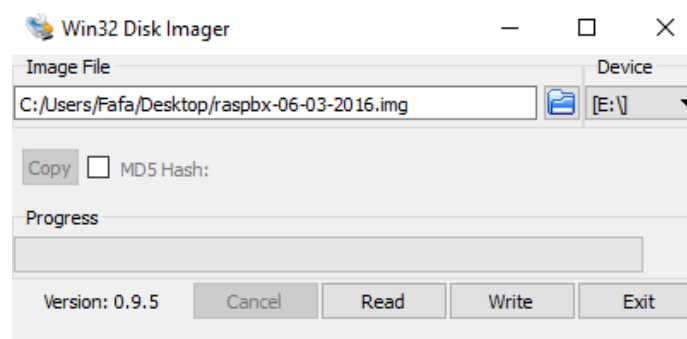


Ilustración 27: Instalación RasPBX

Una vez que el proceso haya llegado a su fin sólo nos queda expulsar la tarjeta SD e insertarla en nuestras Raspberry Pi. Al iniciar por primera vez RasPBX el

servidor está configurado en DHCP, es decir que debemos comprobar desde la página de configuración el router la dirección IP que este le ha asignado.

El usuario y contraseña de acceso por defecto es **admin**.

Un paso muy recomendable a seguir para actualizar los paquetes de RaspBX es acceder a través de SSH con el usuario **root** y contraseña **raspberry**, posteriormente teclear:

#raspbx-upgrade

El servidor comprobará en los directorios de FreePBX las versiones de todos los paquetes de la distribución y si existen nuevas actualizaciones, nos dará la opción de descargarlos e instalarlos.

4.2. Configuración y puesta en marcha de uElastix en Raspberry Pi

En este apartado explicaremos los pasos a seguir para la instalación del servidor de Voz sobre IP Uelastix en una Raspberry PI Ver. 1.

En primer lugar debemos de disponer de una tarjeta SD preferiblemente categoría 10 de al menos 8Gb.

También es necesario realizar toda la instalación desde Linux, en este caso he elegido la versión 15.10 de Ubuntu.

A continuación necesitamos descargar la imagen comprimida de Uelastix (elastix-arm-2014-01-30 .tar) de la web de Elastix, sólo es compatible con Raspberry Pi Ver. 1

Posteriormente descomprimimos la imagen y obtenemos en su interior otros tres archivos comprimidos:

- BOOT.tar
- usr.tar
- root.tar

A continuación introducimos la tarjeta SD en el lector de nuestro equipo y nos dirigimos a la herramienta “Discos” perteneciente a las herramientas de Ubuntu.

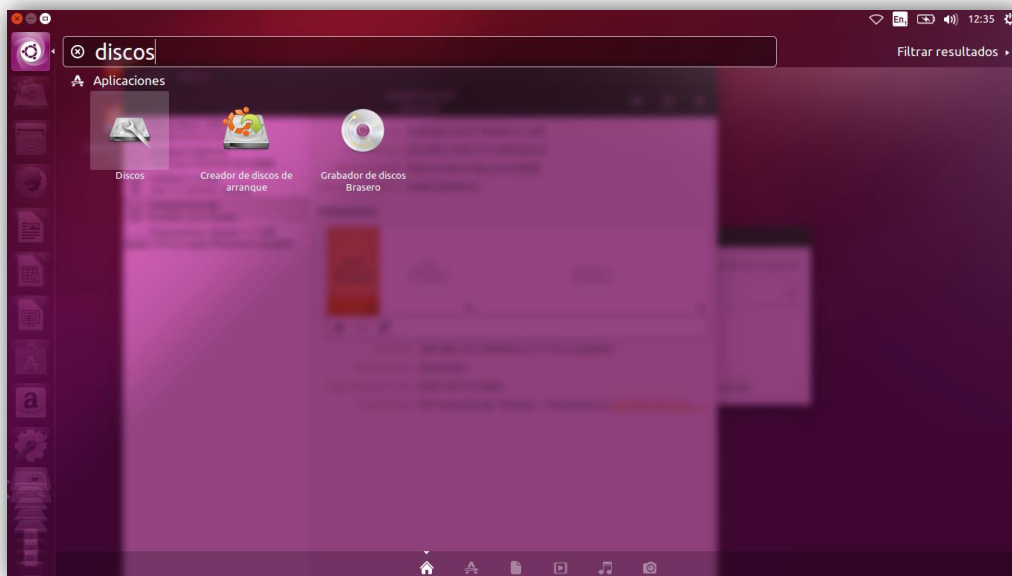


Ilustración 28: Instalación uElastix 1

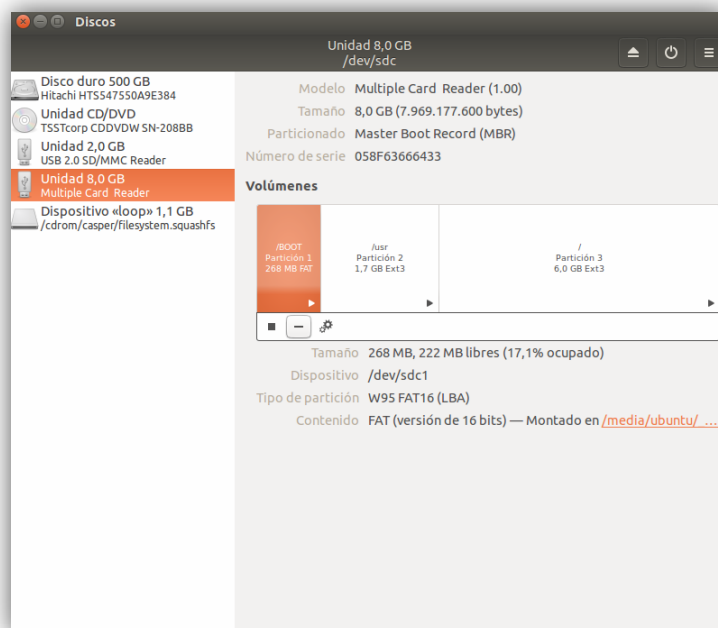


Ilustración 29: Instalación uElastix 2

En ella localizamos nuestra tarjeta y eliminamos todas las particiones existentes, dejándola completamente limpia y sin particionar.

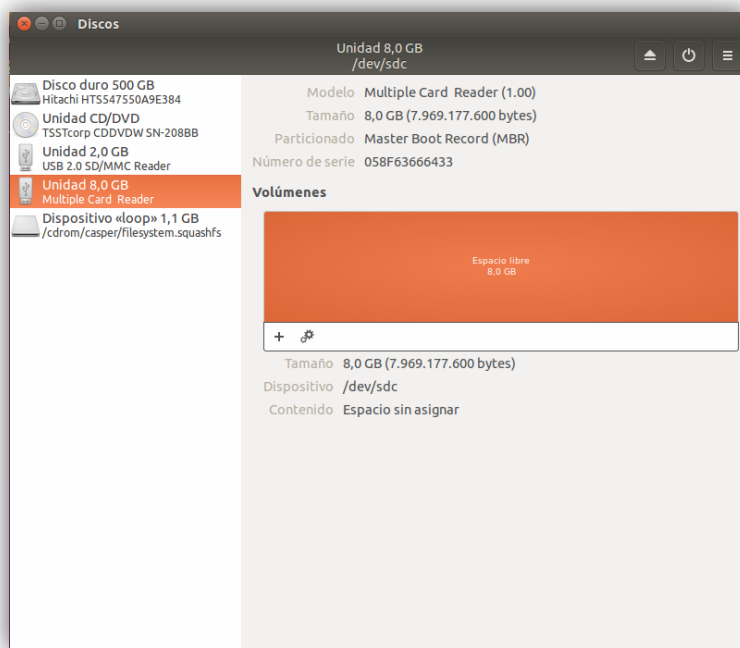


Ilustración 30: Instalación uElastix 3

Al instalar Uelastix en nuestra tarjeta SD estamos instalando una distribución de Linux orientada a procesadores ARM, es decir, necesitamos crear nuevas particiones de la tarjeta para instalar nuestra imagen.

Para instalar Linux necesitaremos dos particiones, una partición pequeña de 300Mb, del tipo **SWAP**, que es utilizada para intercambio de archivos, una segunda de 2Gb llamada **nativa** que será de tipo EXT3 y adicionalmente nosotros vamos a crear una tercera también de tipo EXT3 que utilizaremos para el almacenamiento de archivos como mensajes de voz, grabaciones del sistema, IVR, etc.

A continuación utilizaremos el terminal de Ubuntu para crear las particiones, sistemas de archivos y descomprimir cada archivo en su correspondiente partición.

Para obtener el nombre que Linux le ha asignado a nuestra tarjeta SD ejecutamos:

```
#sudo fdisk -l
```

```

ubuntu@ubuntu: ~
Disk /dev/loop0: 1,1 GiB, 1130688512 bytes, 2208376 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sda: 465,8 GiB, 500107862016 bytes, 976773168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xaa9693fe

Disposit.  Inicio  Start  Final  Sectores  Size  Id  Tipo
/dev/sda1  *      2048   206847  204800    100M   7   HPFS/NTFS/exFAT
/dev/sda2             206848 662198271 661991424 315,7G   7   HPFS/NTFS/exFAT
/dev/sda3        662198272 976769023 314570752   150G   7   HPFS/NTFS/exFAT

Disk /dev/sdb: 1,9 GiB, 1977614336 bytes, 3862528 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x0ceae12e

Disposit.  Inicio  Start  Final  Sectores  Size  Id  Tipo
/dev/sdb1  *      2048   3862527 3860480    1,9G   b   W95 FAT32

Disk /dev/sdc: 7,4 GiB, 7969177600 bytes, 15564800 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x7f42caaf
ubuntu@ubuntu:~$

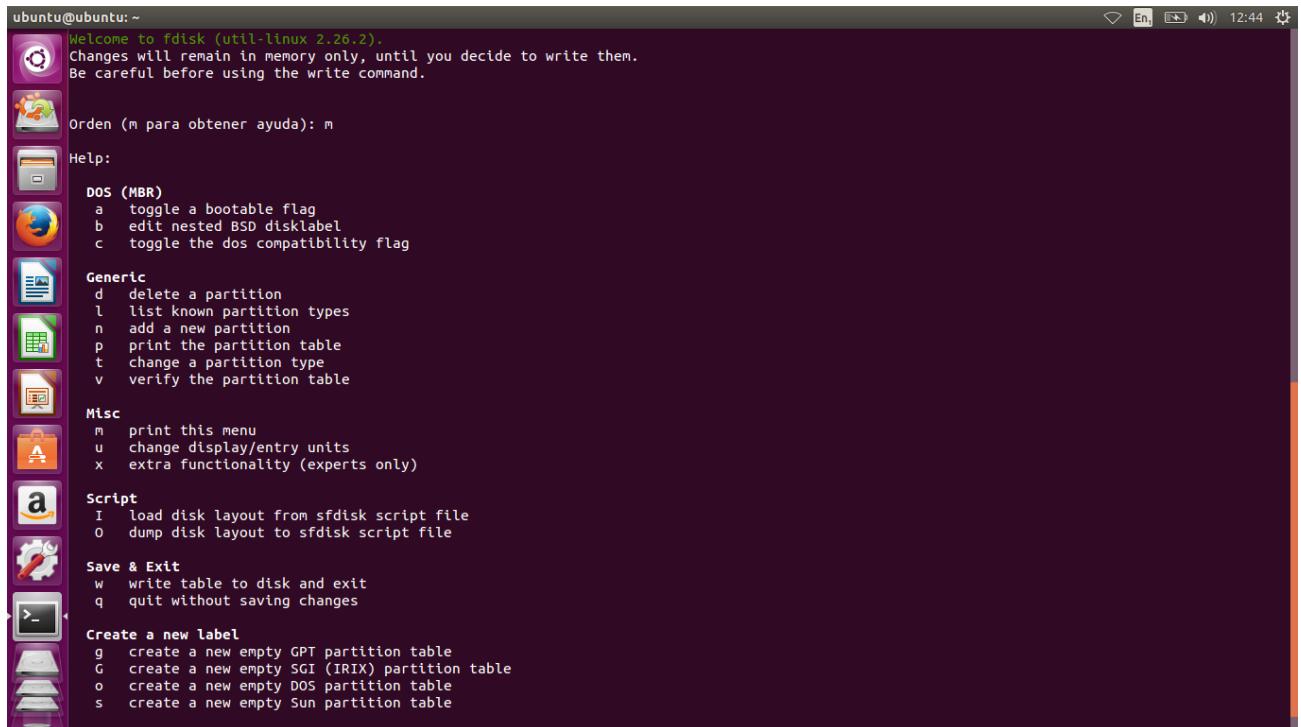
```

Ilustración 31: Instalación uElastix 4

Encontramos una unidad de almacenamiento con 7,4 Gb a la cual el sistema le ha asignado el nombre de **sdc**.

A continuación ejecutamos:

#sudo fdisk /dev/sdc

**Ilustración 32: Instalación uElastix 5**

Para crear nuevas particiones en nuestra tarjeta SD, como ya hemos tratado anteriormente, creamos 3 particiones de tipo primario:

1. Partición de tipo FAT16 de 300Mb.
2. Partición de tipo EXT3 de 2000Mb.
3. Partición de tipo EXT3 con el resto de espacio de la tarjeta.

Instrucciones dentro de la herramienta fdisk:

Primera partición,

- Nueva partición : n
- Tipo de partición (primaria): p
- Número de partición (1-4): 1
- Sector de inicio (defecto): ENTER
- Sector Final: +300M

Segunda partición,

- Nueva partición : n
- Tipo de partición (primaria): p
- Número de partición (1-4): 2
- Sector de inicio (defecto): ENTER
- Sector Final: +2048M

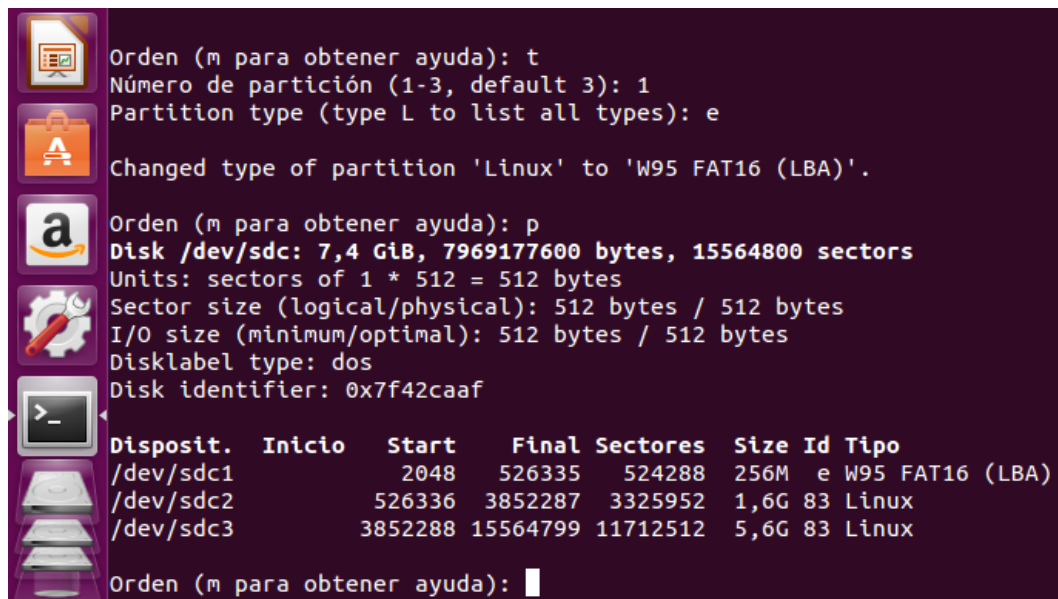
Tercera partición,

- Nueva partición : n
- Tipo de partición (primaria): p
- Número de partición (1-4): 3
- Sector de inicio (defecto): ENTER
- Sector Final (defecto): ENTER

Por defecto todas las particiones son creadas en EXT3 y necesitamos cambiar la primera a FAT16, para ello pulsamos:

- Cambiar tipo de partición: t
- Número de partición: 1
- Tipo de partición a la que desea cambiar: e

Finalmente comprobamos que todas las particiones han sido creadas con el tamaño y tipo deseado, para ello pulsamos **p**.



```

Orden (m para obtener ayuda): t
Número de partición (1-3, default 3): 1
Partition type (type L to list all types): e
Changed type of partition 'Linux' to 'W95 FAT16 (LBA)'.

Orden (m para obtener ayuda): p
Disk /dev/sdc: 7,4 GiB, 7969177600 bytes, 15564800 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x7f42caaf

Disposit.  Inicio  Start  Final Sectores  Size Id Tipo
/dev/sdc1      2048   526335   524288    256M  e  W95 FAT16 (LBA)
/dev/sdc2     526336  3852287  3325952    1,6G  83  Linux
/dev/sdc3     3852288 15564799 11712512    5,6G  83  Linux

Orden (m para obtener ayuda):

```

Ilustración 33: Instalación uElastix 6

El último paso sería escribir la tabla de particiones y salir, pulsamos **w**.

A continuación creamos los sistemas de archivos para cada partición, de tal manera que le asignamos a cada una su respectivo nombre y sistema de archivos.

Ejecutamos:

```
#sudo mkfs.vfat -n /BOOT /dev/sdc1
```

```
#sudo mkfs.ext3 -L /usr /dev/sdc2
```

```
#sudo mkfs.ext3 -L / /dev/sdc3
```

Como es lógico hemos creado cada partición con el mismo nombre del archivo que debemos copiar y descomprimir en su interior.

Ahora cambiamos de ubicación a la carpeta donde tengamos los tres archivos descomprimidos, en mi caso la carpeta Descargas.

```
#cd Descargas
```

Comprobamos que se encuentran los archivos anteriormente descomprimidos en su interior.

ls

- **BOOT.tar.gz**
- **usr.tar.gz**
- **root.tar.gz**

A continuación viene uno de los pasos más importantes de toda la instalación.

Primera partición /BOOT,

- Abrimos la partición /BOOT desde el gestor de archivos.
- Montamos la partición BOOT.

#sudo mount /dev/sdc1 /mnt/

- Copiamos y descomprimos el archivo **BOOT.tar.gz**
(sin utilizar privilegios de SuperUsuario).

#tar -C /mnt/ -xzf BOOT.tar.gz

- Desmontamos la partición BOOT.

#sudo umount /dev/sdc1

Segunda partición /usr,

- Abrimos la partición /usr desde el gestor de archivos.
- Montamos la partición **/usr**

#sudo mount /dev/sdc2 /mnt/

- Copiamos y descomprimos el archivo **usr .tar.gz**

#sudo tar -C /mnt/ -xzf usr.tar.gz

- Desmontamos la partición.

#sudo umount /dev/sdc2

Tercera partición /.

- Abrimos la partición **/** desde el gestor de archivos.
- Montamos la partición **/**.

#sudo mount /dev/sdc3 /mnt/

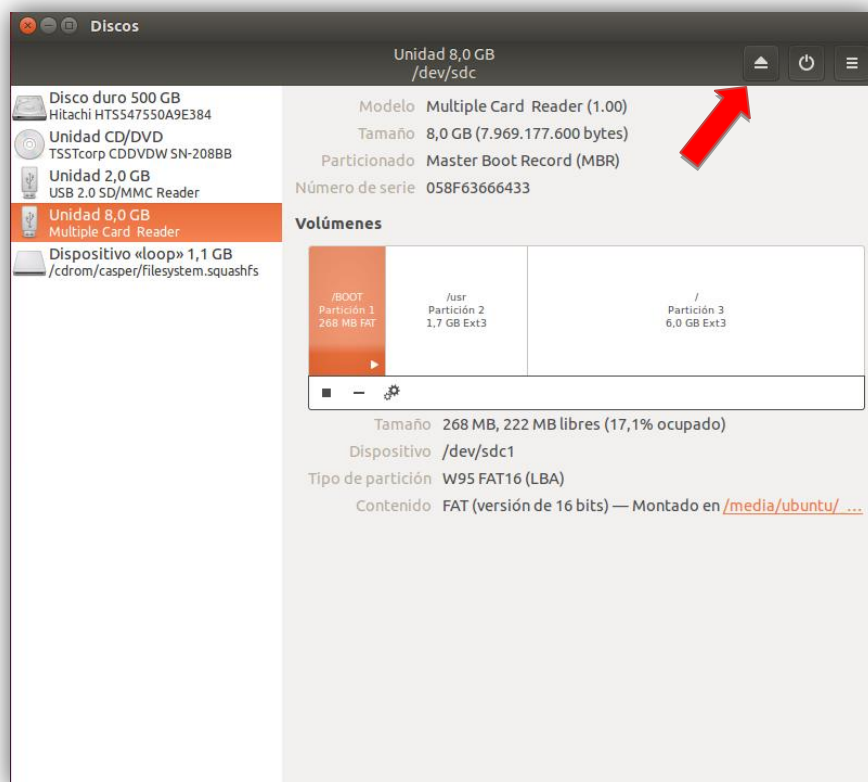
- Copiamos y descomprimos el archivo **root.tar.gz**.

#sudo tar -C /mnt/ -xzf root.tar.gz

- Desmontamos la partición **/**.

#sudo umount /dev/sdc3

Finalmente volvemos a la herramienta “Discos” desde la cual comenzamos y expulsamos la tarjeta SD con seguridad.

**Ilustración 34: Instalación uElastix 7**

El último paso sería introducir la tarjeta SD en la Raspberry Pi y ponerla en funcionamiento con el adaptador de corriente y cable de Ethernet, teniendo en cuenta que debemos gestionar Uelastix desde su servidor Web alojado en la dirección IP 192.168.1.251 y sabiendo que el usuario por defecto es admin y la contraseña palosanto.

La instalación de uElastix en Raspberry Pi ha sido muy tediosa ya que la información obtenida de Internet era errónea y bastante confusa. Normalmente cuando deseamos instalar una distribución de Linux en Raspberry Pi sólo es necesario descargar la imagen en formato .img y descomprimirla en la tarjeta SD pero en este caso ha sido necesario seguir todos los pasos anteriormente explicados para conseguir una instalación exitosa, hasta que descubrí donde se encontraba el fallo invertí varias semanas. [19]

Cabe destacar que navegando por diferentes foros de Internet relacionados con VoIP encontramos un compañero de profesión que había creado una imagen de Uelastix empaquetada, es decir, la distribución de Uelastix en formato Uelastix.img,

por la cual pedía 22,50\$ (uElastix for Raspberry Pi (.img included)) por su descarga, esto nos hace pensar que los pasos a seguir no son triviales. [20]

4.3. Instalación de equipos en laboratorio

En este apartado explicaremos detalladamente los pasos que hemos seguido para la instalación de todos los equipos en el laboratorio 4 de informática en la Escuela Politécnica de Jaén.

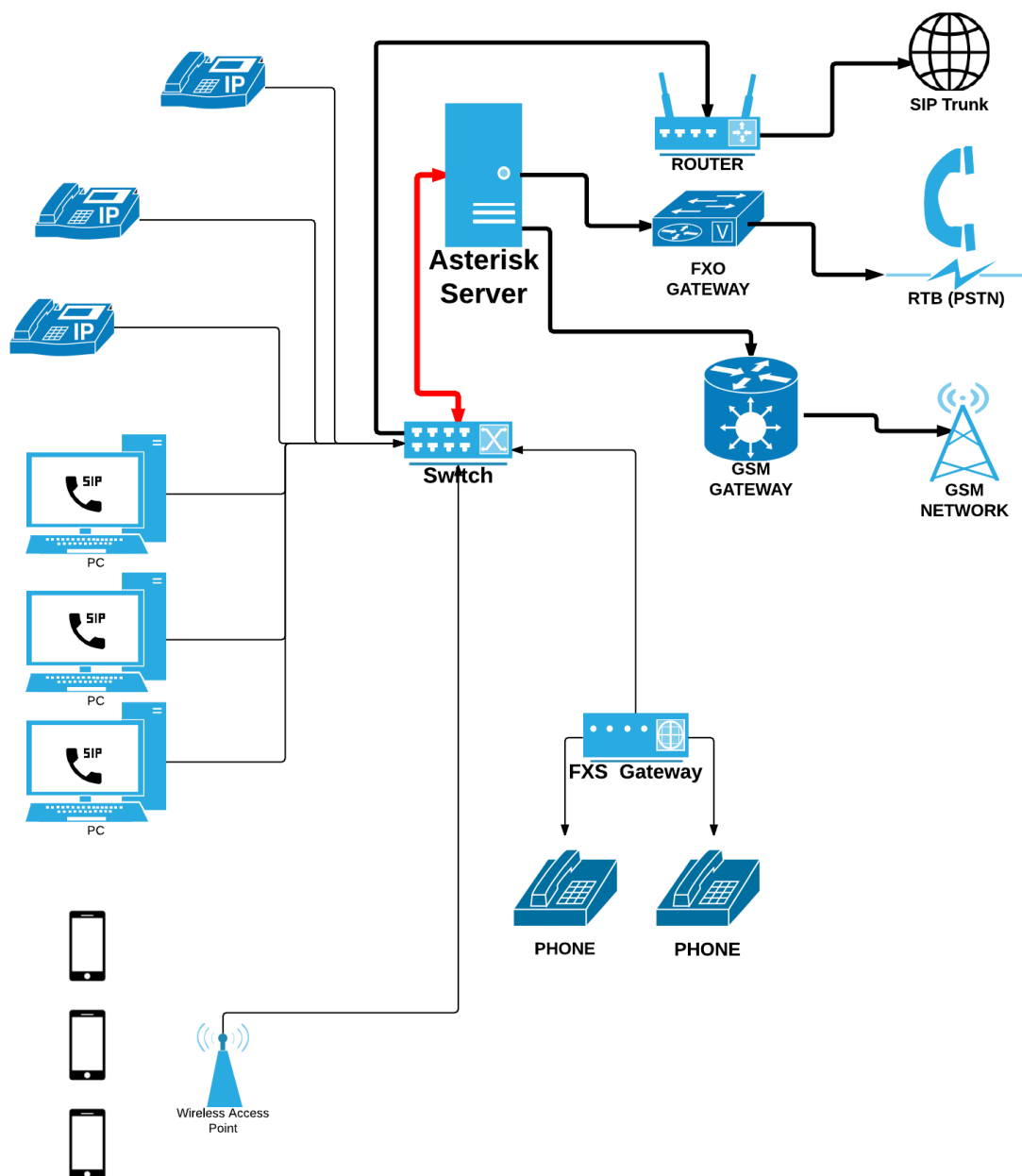


Ilustración 35: Esquema de red, distribución de equipos

Basándonos en el esquema anterior indicaremos todos los terminales compatibles para recibir llamadas VOIP, al igual que todos los dispositivos utilizados para enrutar llamadas hacia el exterior.

En primer lugar disponemos de los teléfonos Cisco ya mencionados en capítulos anteriores, los cuales conectamos al switch para así poder acceder a sus páginas de configuración desde el navegador.



Ilustración 36: Instalación en laboratorio 1

Los teléfonos estaban conectados a un switch POE y este al switch central del laboratorio, es decir que todos los equipos trabajaban en la red 36 (192.168.36.0), esta red era la que el laboratorio 4 tenía asignada por el servicio de informática de la universidad.

Además al comprobar que nuestra Raspberry Pi soportaba todas las llamadas simultaneas entre los teléfonos VoIP , hicimos uso de los equipos disponible en el laboratorio.

Para llevar a cabo esta tarea, instalamos la aplicación Xlite en cada uno de los equipos y posteriormente registramos una extensión por equipo.



Ilustración 37: Instalación en laboratorio 2

Haciendo uso del punto de acceso inalámbrico con el que contaba nuestro router Cisco, utilizamos la red Wi-Fi y registramos varias extensiones SIP por medio de la aplicación Zoiper instalada en nuestros SmartPhones.

Finalmente configuramos el Gateway GrandStream FXS ATA cedido por la universidad de Jaén, siguiendo unos pasos similares a los teléfonos VoIP. Estos dispositivos también cuenta con servidor web desde donde poder gestionarlos, fue posible instalar dos terminales analógicos ya que contaba con dos puertos FXS.

En este punto ya contábamos con terminales suficientes para hacer pruebas reales y comprobar cuantas llamadas simultaneas era capaz de soportar nuestra Raspberry Pi.

4.4. Pruebas en laboratorio

- **Máximo número de llamadas simultaneas**

Una vez que habíamos instalado y configurado todos los 12 teléfonos VoIP disponibles para prácticas en el laboratorio 4 comenzamos a realizar llamadas dos a dos para comprobar el rendimiento máximo que podía ofrecer la Raspberry Pi.

Como es lógico, para realizar una llamada es necesario utilizar dos terminales, es decir, con 12 teléfonos VoIP mantuvimos 6 llamadas simultáneas sin afectar a la calidad de audio en las mismas.

Viendo los resultados obtenidos repetimos la misma prueba con todos los equipos del aula y aparente las llamadas eran gestionadas sin problemas, posteriormente analizamos el tráfico existente entre teléfonos con la aplicación WireShark y descubrimos dos datos importantes:

El servidor no gestiona el audio de los terminales, es decir, solamente se encarga encaminar llamadas, sin llegar a procesar el audio, por esta razón es capaz de gestionar un gran número de llamadas sin sufrir colapsos.

Los ordenadores del laboratorio no transmitieron audio alguno ya que no contaban con micrófono ni altavoces, por lo tanto estos equipos no nos fueron útiles para obtener información precisa acerca del máximo número de llamadas soportadas por la Raspberry Pi.

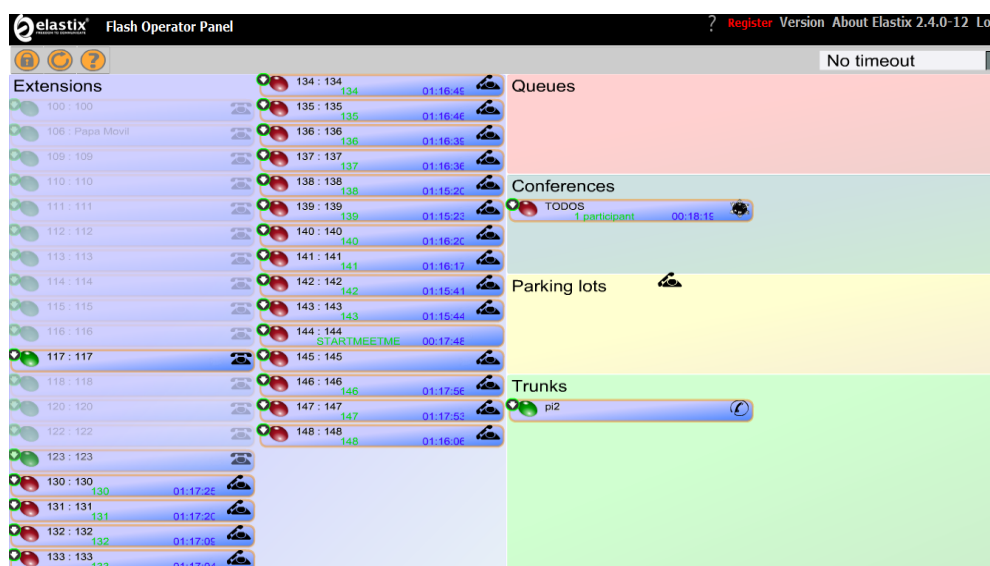


Ilustración 38: Llamadas simultaneas en uElastix

Como resultado obtuvimos que la Raspberry Pi ver. 1 modelo B fue capaz de gestionar 6 llamadas simultáneas sin problema, resultado más que favorable si contamos con las limitaciones de la placa.

- Conferencias

Conferencias es una opción disponibles en las distribuciones de RasPBX y uElastix diseñada para hablar todos los participantes en una misma conversación.

Esta función nos fue muy útil para comprobar realmente el rendimiento de nuestra placa ya que en este caso el audio de todas las llamadas es gestionado por el servidor.

El audio generado en un teléfono es reenviado a todos los demás participantes de la conferencia y viceversa. Al comenzar con la prueba, rápidamente saturamos la capacidad de procesamiento de la placa.

Como resultado obtuvimos que como máximo, podemos mantener una conferencia con 3 participantes, puesto que a partir del cuarto, la calidad de audio baja drásticamente y la conversación comienza a entrecortarse.

- **Grupo de Timbrado**

Tanto en RasPBX como uElastix es posible configurar grupos de timbrado, se trata de crear grupos de extensiones y asignarles una nueva extensión a esos grupos, de tal manera que cuando realizamos una llamada a esa nueva extensión todos los terminales suenan simultáneamente y el primer teléfono que descuelgue la llamada toma el control, silenciando al resto de teléfonos del grupo.

Esta prueba funciono fue realizada sin problema puesto que de nuevo no afecta al rendimiento de las Raspberry Pi.

- **Análisis de protocolos mediante WireShark**

Como ya comentamos en capítulos anteriores, WireShark es una herramienta muy útil para el análisis de protocolos de red. En las pruebas de laboratorio hemos obtenido mucha información referente a las llamadas gracias a esta herramienta.

- **Analizando el protocolo SIP:**

- Obtuvimos todos los registros de las extensiones en el servidor VoIP.

- Los mensajes REGISTER, INVITE, ACK y BYE generados por cada extensión y enviados al servidor VoIP.
- El puerto de registro utilizado por las extensiones, (5060 udp).
- Origen y destino de las llamadas.
- **Analizando el protocolo RTCP:**
 - Obtuvimos el tamaño de los datagramas enviados desde las extensiones al servidor.
 - Tasa de paquetes perdidos.
 - Mensajes de control de llamadas, comprobando que el destinatario sigue estando disponible.
- **Analizando el protocolo RTP:**
 - Obtuvimos el códec de audio utilizado.
 - Cabeceras de los mensajes enviados desde los teléfonos al servidor.
 - Número de secuencia.
 - Datagramas completos codificados.
 - Dirección IP del emisor y receptor.
 - Puerto de origen y destino de las llamadas.

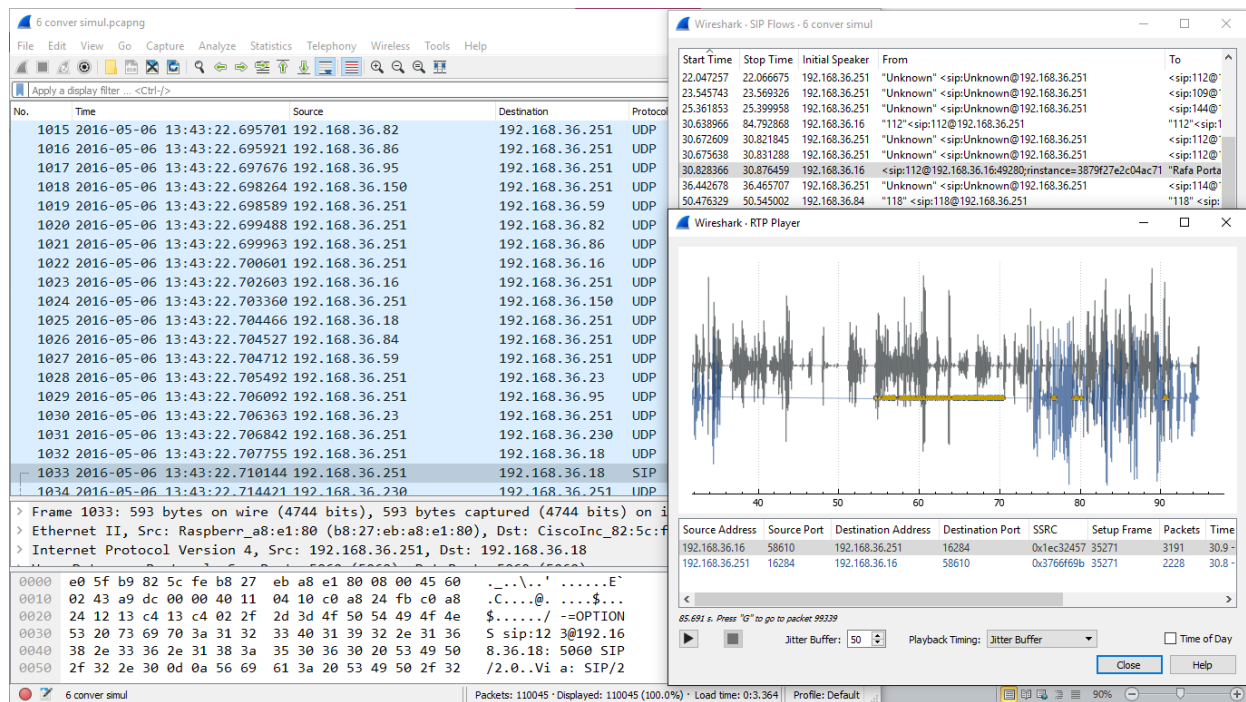


Ilustración 39: Análisis Wireshark

4.5. Conexión entre diferentes sedes mediante una troncal IAX

A continuación explicamos cómo configurar una troncal IAX para interconectar dos sedes.

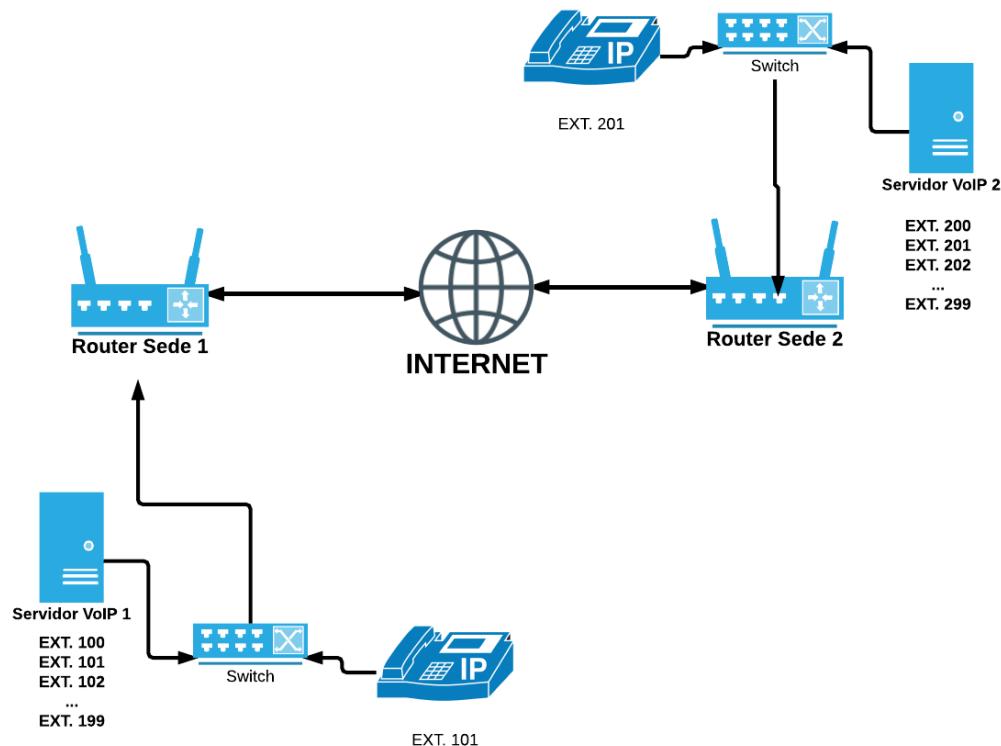


Ilustración 40: Esquema de red, Troncal IAX

En primer lugar debemos abrir el puerto **4569** en UDP en el cada router, como ya explicamos en el capítulo de IAX, este protocolo transmite la información multiplexada a través de este puerto.

Posteriormente debemos de definir un rango de extensiones en cada servidor ubicados en diferentes sedes, es decir, el servidor 1 utilizará extensiones de la 100 a 199 y el servidor 2 de la 200 a 299.

En la siguiente tabla definimos los parámetros necesarios para configurar correctamente las direcciones IP de ambos servidores VoIP y las rutas que deben seguir.

	Dir. IP LAN Servidor	Dir. IP LAN Router	Dir. IP WAN Router	Rango de extensiones
Servidor 1 (Pi1)	192.168.1.250	192.168.1.1	217.217.129.28	100 - 199
Servidor 2 (Pi2)	192.168.2.251	192.168.2.1	150.214.170.21	200 - 299

En el router de cada sede debemos abrir el puerto 4569 y redirigirlo desde el puerto WAN a la dirección IP local del servidor VoIP, respetando las direcciones descritas en la tabla anterior.

Posteriormente configuraremos las troncales IAX en cada uno de los servidores VoIP.

Creamos una troncal de tipo IAX en el servidor 1 rellenando los siguientes campos:

Trunk Name: **Pi2**

Outgoing Settings: **pi2**

host=150.214.170.21

username=pi2
secret=123456
encryption=aes128
auth=md5
type=friend
context=from-internal
trunk=yes
requirecalltoken=no

The screenshot shows the Asterisk PBX web interface. On the left is a sidebar menu with categories like 'Outbound Routes', 'Trunks', 'Inbound Call Control', etc. The 'Trunks' section is selected. The main content area shows the configuration for a trunk named 'pi2'. The 'General Settings' section includes fields for 'Trunk Name' (pi2), 'Outbound Caller ID', 'CID Options' (set to 'Allow Any CID'), 'Maximum Channels', 'Disable Trunk' (unchecked), and 'Monitor Trunk Failures' (unchecked). The 'Dial Number Manipulation Rules' section has buttons for '(prepend)', '+ prefix', and 'match pattern', along with '+ Add More Dial Pattern Fields' and 'Clear all Fields'. Below this is a 'Dial Rules Wizards' dropdown set to '(pick one)' and an 'Outbound Dial Prefix' field. The 'Outgoing Settings' section shows 'Trunk Name' as 'pi2' and a 'PEER Details' text area containing the following configuration:


```
host=192.168.0.6
username=pi1
secret=123456
encryption=aes128
auth=md5
type=friend
context=from-internal
trunk=yes
requirecalltoken=no
```

Ilustración 41: Configuración Troncal IAX 1

Para crear la rutas salientes hacia el servidor 2 pinchamos sobre **Outbound Routes** y definimos los siguientes parámetros.

Route Name: ***RPi2***

Match Pattern: 2XX

Trunk Sequence: Pi2

Edit Route
Delete Route RPi2

Route Settings

Route Name: RPi2

Route CID: ☐ Override Extension

Route Password:

Route Type: ☐ Emergency ☐ Intra-Company

Music On Hold? default

Time Group: ---Permanent Route---

Route Position: Last after RPi2

Additional Settings

PIN Set: None

Dial Patterns that will use this Route

(prepend) + prefix | 4XX / CallerId

(prepend) + prefix | [match pattern] / CallerId

+ Add More Dial Pattern Fields

Dial patterns wizards: (pick one)

Trunk Sequence for Matched Routes

0 P12

1

Ilustración 42: Configuración Troncal IAX 2

La configuración en el servidor 2 sería idéntica modificando la dirección IP del servidor remoto y el rango de extensiones a los que podemos llamar.

Trunk Name: **Pi1**

Outgoing Settings: **pi1**

host=217.217.129.28

username=pi1

secret=123456

encryption=aes128

auth=md5

type=friend

context=from-internal

trunk=yes

requirecalltoken=no

Y las rutas salientes se configurarían de la siguiente manera.

Route Name: **RPi1**

Match Pattern: **1XX**

Trunk Sequence : **Pi1**

A partir de este momento quedaría configurada la troncal IAX en los dos servidores y podríamos realizar llamadas a extensiones registradas en el servidor de la otra sede como si se trataran de extensiones locales.

Esta configuración es muy utilizada en empresas multinacionales para interconectar sus distintas sedes y realizar llamadas entre ellas sin coste alguno.

Para establecer una comunicación segura entre las dos sedes sería muy recomendable configurar una **VPN** como ya definimos en la parte teórica puesto que necesitamos tener acceso al router de cada sede. Establecer una VPN con una empresa proveedora de telefonía IP no sería posible a menos que la empresa nos proporcione ese servicio.

De igual modo podríamos llevar a la práctica la configuración de una red virtual **VLAN** para separar la red LAN utilizada para los equipos telefónicos del resto de equipos informáticos.

4.6. Configuración de llamadas externas mediante troncal SIP

En primer lugar hemos contratado una troncal SIP en una empresa proveedora de telefonía voz sobre IP, las características de esta troncal son:

- 3 canales, (3 llamadas simultaneas)
- 1 número nacional
- Codecs compatibles G711 y G729
- Todos los servidores ubicados en España
- Latencia mínima garantizada

Una vez contratado el servicio, la empresa nos proporcionó todos los datos correspondientes a la configuración de la troncal SIP en nuestro servidor.

A continuación describiremos todos los pasos necesarios para realizar llamadas a través de una troncal SIP. Debemos crear una troncal SIP y posteriormente adjudicarle rutas salientes y entrantes para enrutar todas las llamadas realizadas.

Desde la página de configuración del servidor VoIP , pulsamos sobre CONNECTIVITY creamos una nueva troncal de tipo SIP.

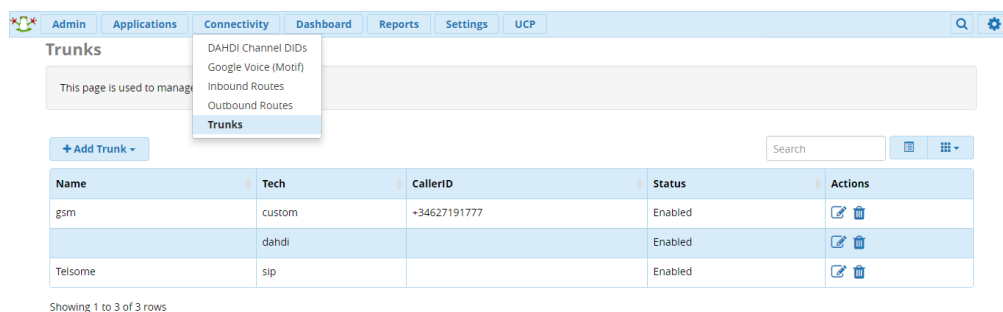


Ilustración 43: Configuración Troncal SIP 1

Posteriormente creamos un nombre de troncal, este caso Telsome y seleccionamos el máximo número de canales a utilizar.

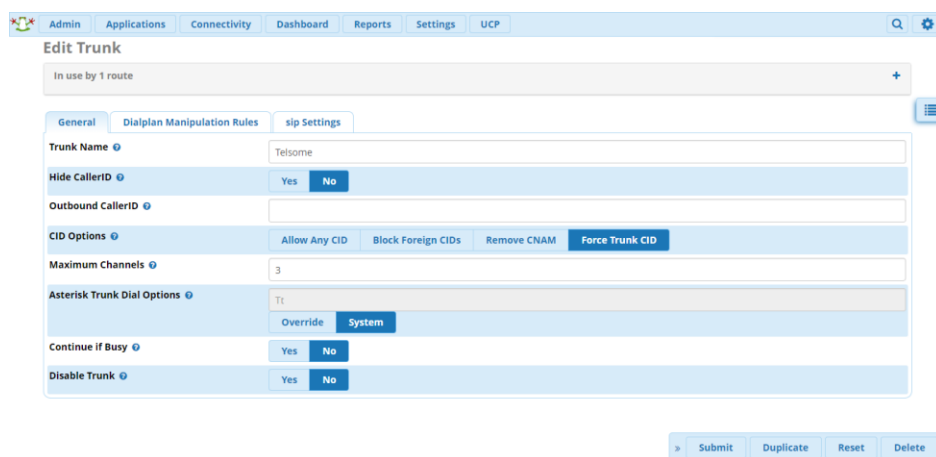


Ilustración 44: Configuración Troncal SIP 2

En la última pestaña, “ Sip Settings ” elegimos la pestaña OUTGOING y un nombre de troncal para realizar llamadas salientes, telsome_out y escribimos todos los parámetros referentes a la troncal SIP.

host=92.54.10.189 (dirección IP de la empresa proveedora de telefonía VoIP)

username=5929*601 (usuario asignado)

secret=JS3Q#r*s (contraseña asignada)

sendrpid=yes (envío de la extensión del usuario que realiza las llamadas)

qualify=yes

fromdomain=192.168.0.251 (dirección local del servidor)

type=peer (tipo de conexión)

maxexpirey=600 (máximo tiempo de espera antes de que expire la conexión)

defaultexpire=180

allow=ulaw&alaw (códec de audio permitidos)

Ilustración 45: Configuración Troncal SIP 3

A continuación, en la pestaña Incoming definimos la siguiente configuración.

User context: **telsome_in**

secret= JS3Q#r*s

type=user

context=from-trunk&from-trunk

qualify=yes

fromuser=5929*601

canreinvite=no

insecure=very

Por último añadimos en **Register String** la siguiente cadena formada por:

nombre_usuario:contraseña@dir_ip_servidorSIP/nombre_usuario

5929*601:JS3Q#r*s@92.54.10.189/5929*601

Ilustración 46: Configuración Troncal SIP 4

Con estos pasos habremos configurado la troncal SIP, ahora debemos establecer rutas salientes y entrantes para enrutar las llamadas.

Añadimos una ruta saliente pulsando en la pestaña Connectivity y Outbound Routes.

Name	Outbound CID	Attributes	Actions
gsmout			
telsomeout			

Ilustración 47: Configuración Troncal SIP 5

A continuación definimos un nombre de ruta saliente y elegimos en el apartado “Trunk Sequence for Matched Routes” la troncal SIP previamente creada.

Ilustración 48: Configuración Troncal SIP 6

Ahora indicamos los destinos a los que deseamos llamar desde esta troncal, siguiendo los patrones de marcado.

X: representa un número entre 0 y 9

Z: representar un número entre 1 y 9

N: representar un número entre 2 a 9

. : Cuando hay un punto en el patrón, lo que significa es que puede haber cualquier número después del punto.

En nuestro caso, queremos realizar llamadas a teléfonos fijos nacionales, escribimos “9.” Para indicar que todos los números que empiecen por 9 marcados desde las extensiones van a ser enrutados a través de esta troncal.

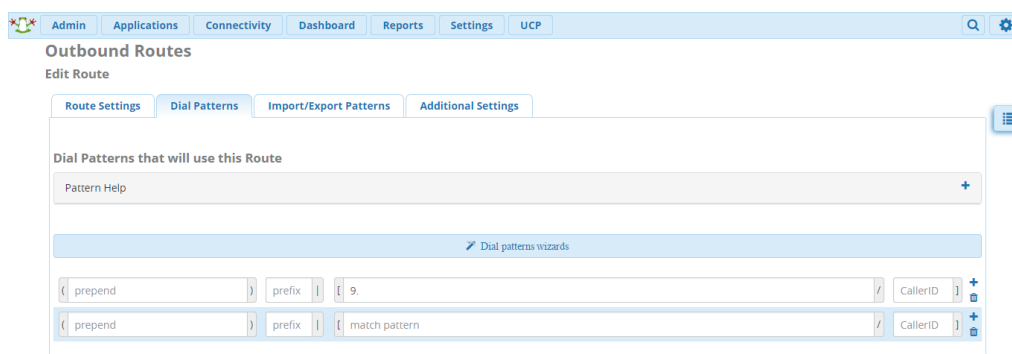


Ilustración 49: Configuración Troncal SIP 7

Finalmente definimos las rutas entrantes para indicar el destino de las llamadas correspondientes al número de teléfono asignado por la compañía. Para ellos pulsamos nuevamente sobre Connectivity e Inbound Routes.

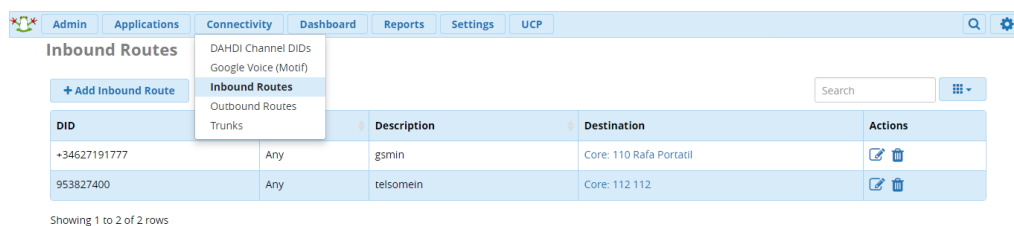


Ilustración 50: Configuración Troncal SIP 8

Aquí debemos indicar el número de teléfono asignado y el destino de tales llamadas. Es posible elegir entre posibles destinos a grupos de timbrado, conferencias, extensiones o incluso IVR.

The screenshot shows the Asterisk web interface with the 'Inbound Routes' section selected. The route 'telsomein' is being edited. The configuration fields are as follows:

Field	Value
Description	telsomein
DID Number	953827400
CallerID Number	ANY
CID Priority Route	Yes
Alert Info	Double-Click to see options or type freeform
CID name prefix	
Music On Hold	Default
Set Destination	Extensions
	112 112

Ilustración 51: Configuración Troncal SIP 9

Con esos pasos quedaría configurada la troncal SIP y desde cualquier extensión registrada en el servidor se podrían realizar llamadas a teléfonos fijos nacionales.

4.7. Configuración de llamadas externas mediante troncal GSM

En primer lugar debemos acceder al terminal de la Raspberry Pi por medio de SSH, una vez dentro tenemos que instalar el modem GSM en el servidor, previamente este ha sido conectado en uno de los puertos USB de la Raspberry Pi.

Tecleamos *install-dongle*.

```

192.168.0.6 - PuTTY
Ign http://archive.raspberrypi.org jessie/main Translation-en
Ign http://archive.raspberrypi.org jessie/ui Translation-en_GB
Ign http://archive.raspberrypi.org jessie/ui Translation-en
Get:3 http://mirrordirector.raspbian.org jessie/contrib armhf Packages [37.5 kB]
Get:4 http://mirrordirector.raspbian.org jessie/non-free armhf Packages [70.3 kB]
Get:5 http://mirrordirector.raspbian.org jessie/rpi armhf Packages [1,356 B]
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/main Translation-en
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en
Fetched 9,091 kB in 25s (361 kB/s)
Reading package lists... Done
Your system is up to date.
root@raspbx:~# install-dongle
Installing GSM VoIP gateway with chan_dongle.
Please read the documentation at http://raspbx.org/dongle

Please enter the phone number of your SIM card
(defaults to +1234567890 if left blank):

```

Ilustración 52: Configuración Troncal GSM 1

A continuación debemos introducir el número de teléfono (+34) de la tarjeta SIM insertada dentro del Modem GSM y seguir los pasos hasta completar la instalación del mismo.

Posteriormente accedemos a la página de configuración del servidor VoIP para crear una troncal para nuestro modem GSM.

Debemos crear una troncal del tipo Custom en la pestaña de Connectivity.

The screenshot shows the 'Edit Trunk' configuration page. At the top, there's a navigation bar with tabs: Admin, Applications, Connectivity, Dashboard, Reports, Settings, and UCP. Below this, the 'Edit Trunk' section is displayed. It includes a status bar 'In use by 1 route' with a plus icon. The 'General' tab is selected, showing various configuration options: 'Trunk Name' is 'gsm', 'Hide CallerID' is 'No', 'Outbound CallerID' is '+34627191777', 'CID Options' has 'Allow Any CID' selected, 'Maximum Channels' is '1', 'Asterisk Trunk Dial Options' is 'Tt', 'Continue if Busy' is 'No', and 'Disable Trunk' is 'No'. There are also buttons for 'Block Foreign CIDs', 'Remove CNAM', and 'Force Trunk CID'.

Ilustración 53: Configuración Troncal GSM 2

En la configuración de la troncal es necesario asignarle un nombre a la troncal, definir de nuevo el número de teléfono desde el que realizaremos las llamadas utilizando el prefijo (+34) y en la pestaña de **Custom Settings** definir la siguiente cadena de caracteres.

dongle/dongle0/\$OUTNUM\$

This screenshot shows the 'Edit Trunk' page with the 'Custom Settings' tab selected. The 'Custom Dial String' field is populated with the value 'dongle/dongle0/\$OUTNUM\$'. The rest of the page structure is consistent with the previous screenshot.

Ilustración 54: Configuración Troncal GSM 3

Una vez creada la troncal GSM, configuraremos las rutas entrantes y salientes del mismo modo que en la configuración de la troncal SIP, pero con la peculiaridad que en este caso esta troncal será utilizada para enrutar llamadas a teléfonos móviles, es decir, en el plan de marcado de rutas salientes definimos:

6.

De esta manera todos los teléfonos marcados que comiencen por 6 serán enrutados por esta troncal.

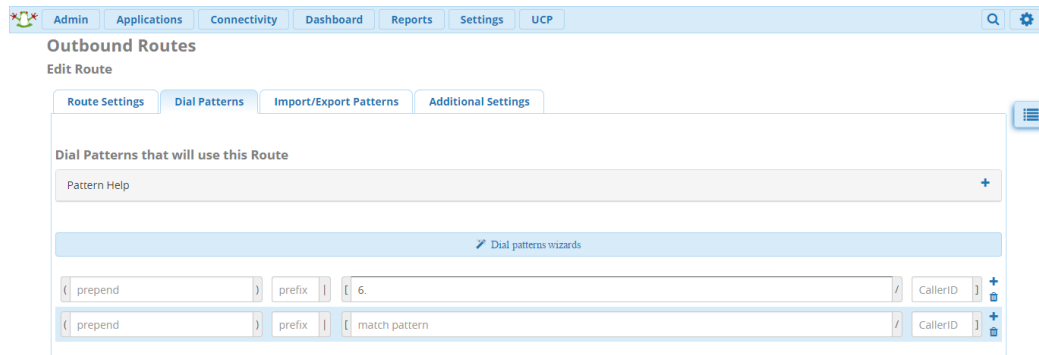


Ilustración 55: Configuración Troncal GSM 4

4.8. Configuración de llamadas externas mediante troncal FXO

Esta configuración es algo más complicada ya que debemos sincronizar el servidor VoIP con el Gateway FXO.

En primer lugar debemos de crear una troncal del tipo SIP como la anteriormente configurada, pero con algunas modificaciones ya que en esta ocasión no registramos nuestra troncal en servidores remotos de una compañía sino que la registramos en el Gateway FXO.

En nombre de troncal ponemos ht503 haciendo referencia al dispositivo, en Outbound CallerID indicamos el número de teléfono fijo desde el cual deseamos realizar las llamadas.

Edit Trunk

In use by 1 route

General | **Dialplan Manipulation Rules** | **sip Settings**

Trunk Name: ht503

Hide CallerID: Yes **No**

Outbound CallerID: 953232266

CID Options: **Allow Any CID** Block Foreign CIDs Remove CNAM Force Trunk CID

Maximum Channels:

Asterisk Trunk Dial Options: Tt
Override **System**

Continue if Busy: Yes **No**

Disable Trunk: Yes **No**

Ilustración 56: Configuración Troncal FXO 1

En la pestaña de **sip Setting** rellenamos el apartado de **Outgoing** con la siguiente información.

Trunk Name: **ht503**

Peer Details:

host=192.168.0.250 (dirección IP del Gateway FXO)

type=peer

canreinvite=no

insecure=very

dtmfmode=rf2833

quality=yes

port=5062

Edit Trunk

In use by 1 route

General | **Dialplan Manipulation Rules** | **sip Settings**

Outgoing | Incoming

Trunk Name: ht503

PEER Details:

host=192.168.0.250
type=peer
canreinvite=no
insecure=very
dtmfmode=rfc2833
quality=yes
port=5062

Ilustración 57: Configuración Troncal FXO 2

En el apartado de **Incoming** indicamos los siguientes parámetros:

User Context: **ht503_la_entrada**

User Details:

context=from-trunk
host=dynamic
insecure=very
type=friend
dtmfmode=rfc2833
secret=1234aa
nat=yes

El apartado de Register String tan importante en la configuración de la troncal SIP ahora lo dejamos en blanco ya que no registramos este troncal en servidores remotos.

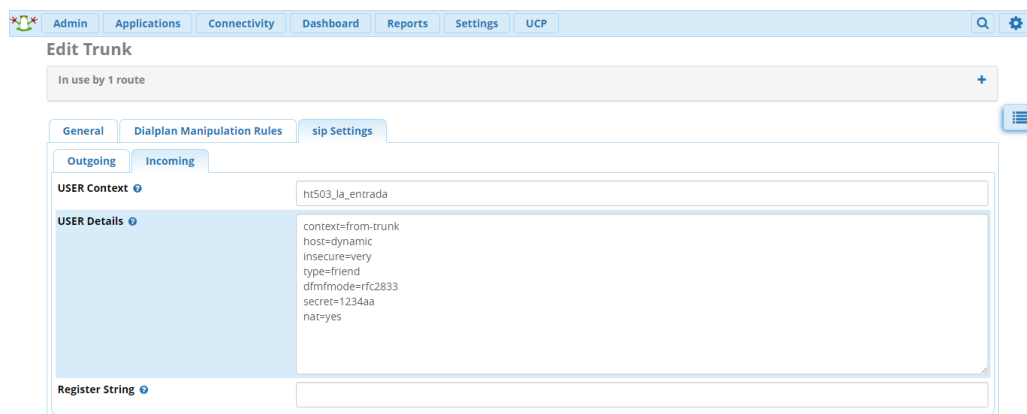


Ilustración 58: Configuración Troncal FXO 3

A continuación debemos configura las rutas entrantes y salientes de igual modo que los casos anteriores, en rutas salientes definimos esta troncal para hacer llamadas tanto a número móviles como a fijos de tal manera que escribimos en el apartado de planes de marcado:

9.

6.

Recordemos que esta troncal será utilizada en caso de respaldo, es decir si las troncales SIP (para fijos) y GSM (para móviles) fallasen, esta troncal entraría en funcionamiento evitando que la llamada no llegue a destino, para ello seleccionamos en las rutas salientes de las troncales SIP y GSM la opción de tomar como troncal alternativa la troncal FXO, colocándola en la segunda posición en el panel de “Trunk Sequence for Matched Routes”.

Finalmente debemos configurar el Gateway FXO para enrutar las llamadas procedentes del servidor VoIP. Debemos establecer una dirección IP estática para este dispositivo en la pestaña de **Basic Settings**.

Grandstream Device Configuration

STATUS BASIC SETTINGS ADVANCED SETTINGS FXS PORT FXO PORT

End User Password: (purposely not displayed for security protection)

Web Port: 80 (default for HTTP is 80)

Telnet Server: ☐ No ☒ Yes

IP Address: ☒ dynamically assigned via DHCP

DHCP hostname: (optional)

DHCP vendor class ID: HT500 (optional)

☐ use PPPoE

PPPoE account ID:

PPPoE password:

PPPoE Service Name:

Preferred DNS server: 0 .0 .0 .0

☒ statically configured as:

IP Address: 192 .168 .0 .250

Subnet Mask: 255 .255 .255 .0

Default Router: 192 .168 .0 .1

DNS Server 1: 62 .81 .16 .148

DNS Server 2: 62 .81 .16 .213

Ilustración 59: Configuración Troncal FXO 4

Al final de esta pestaña configuramos la opción de **Unconditional Call Forward to VoIP** para reenviar todas las llamadas entrantes por la línea analógica al servidor VoIP.

953232266 @ 192.168.0.6 : 5060

	User ID	Sip Server	Sip Destination Port
Unconditional Call Forward to VOIP:	953232266	@ 192.168.0.6	: 5060

Ilustración 60: Configuración Troncal FXO 5

A continuación debemos configura la pestaña de **FXO PORT**, en el que definimos en primer lugar la dirección IP donde se encuentra es servidor VOIP, también definimos el nombre de usuario y contraseña que hemos descrito al configurar el apartado INCOMING de la troncal.

Primary SIP Server: **192.168.0.6**

Sip User ID: **ht503_la_entrada**

Authenticate ID: **ht503_la_entrada**

Authenticate Password: **1234aa**

Grandstream Device Configuration

STATUS BASIC SETTINGS ADVANCED SETTINGS FXS PORT FXO PORT

Account Active: ☐ No ☒ Yes

Primary SIP Server: (e.g., sip.mycompany.com, or IP address)

Failover SIP Server: (Optional, used when primary server no response)

Prefer Primary SIP Server: ☒ No ☐ Yes (yes - will register to Primary Server if Failover registration expires)

Outbound Proxy: (e.g., proxy.myprovider.com, or IP address, if any)

SIP Transport: ☒ UDP ☐ TCP ☐ TLS (default is UDP)

NAT Traversal: ☒ No ☐ Keep-Alive ☐ STUN ☐ UPnP

SIP User ID: (the user part of an SIP address)

Authenticate ID: (can be identical to or different from SIP User ID)

Authenticate Password: (purposely not displayed for security protection)

Name: (optional, e.g., John Doe)

Ilustración 61: Configuración Troncal FXO 6

Más abajo en el apartado de **Dial Plan** definimos el tipo de números a los que podemos llamar, en este caso hemos definido números de 9 dígitos, para ello, colocamos la siguiente cadena en tal apartado.

Dial Plan: {xxxxxxxxxx}

Por orden de preferencia elegimos los codecs de audio que deseamos utilizar, en este caso le damos preferencia a los codecs sin compresión como

PCMA (g711 a-law) y PCMU (g711 u-law).

Use First Matching Vocoder in 200OK SDP: ☒ No ☐ Yes

Preferred Vocoder: (in listed order)

choice 1:	PCMA ▼
choice 2:	PCMU ▼
choice 3:	G723 ▼
choice 4:	G729 ▼
choice 5:	G726-32 ▼
choice 6:	iLBC ▼
choice 7:	G729E ▼
choice 8:	AAL2-G726-16 ▼

Ilustración 62: Configuración Troncal FXO 7

Por último definimos el país donde deseamos utilizar este dispositivo ya que en función al mismo los tonos de marcado son distintos.

FXO Termination

Enable Current Disconnect: ☐ No ☒ Yes (Default Yes. If set to yes, enter threshold below)

Current Disconnect Threshold (ms): (50-800 milliseconds. Default 100 milliseconds)

Enable PSTN Disconnect Tone Detection: ☐ No ☒ Yes (Default No)

(If set to yes, the following tone is used as the disconnect signal)

PSTN Disconnect Tone: (Syntax: f1=freq@vol, f2=freq@vol, c=on1/off1-on2/off2-on3/off3;)
(Allowed Range: freq = 0 to 4000Hz; vol = -40 to -24dBm)
(Default: Busy Tone: f1=480@-32,f2=620@-32,c=500/500;)

AC Termination Model ☐ Country-based ☒ Impedance-based (Default Country-based)

Country-based:

Impedance-based:

Ilustración 63: Configuración Troncal FXO 8

4.9. Comparación con soluciones comerciales

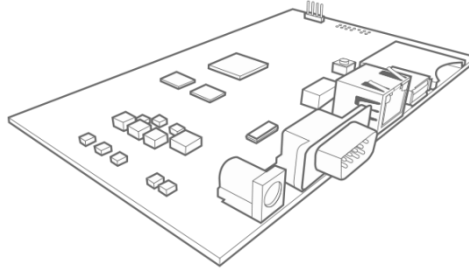
Actualmente en el mercado existe un producto similar a nuestro proyecto con la peculiaridad que su precio es cuatro veces superior y el rendimiento inferior según el fabricante.

La propia empresa Elastix además de ofrecer el software, también ofrece una gama de centralitas PBX hardware con Elastix corriendo en su interior, en este caso hablamos del modelo **Elastix MicroUCS** que cuenta con una placa **ASIRI** de características inferiores a Raspberry Pi, ya que su procesador es sólo de 400MHz frente a los 700MHz de nuestra placa y cuenta con 256 MB de memoria RAM frente a los 512MB de nuestra solución.

Según indican en su página web, con este dispositivo se pueden registrar hasta 15 extensiones y mantener hasta 4 llamadas simultáneas sin perjudicar la calidad de audio. [21]



Ilustración 64: Elastix MicroUCS

**Ilustración 65: Placa ASIRI**

El precio de esta centralita Elastix ronda los 200€, sin embargo el rendimiento del producto es inferior al nuestro, es decir que nuestro proyecto se podría llevar a cabo con fines comerciales y con un precio mucho más competitivo que los productos similares del mercado.

5. Estudio de viabilidad

En este apartado describiremos realizaremos un estudio acerca de la viabilidad del proyecto en cuestiones de tiempo y costes, con ello tratamos de conocer si la idea es viable y por lo tanto rentable.

5.1. Planificación temporal

A la hora de comenzar un proyecto es crucial la planificación previa de las actividades a realizar, su definición y asignación de espacio temporal, de manera que marquemos un objetivo a cumplir durante los intervalos definidos

5.1.1. Tareas

- **Análisis**
 - **Búsqueda de información**

Documentación acerca de la situación actual de las tecnologías VoIP e investigación sobre el desarrollo de esta tecnología.

- **Análisis de soluciones hardware**

Estudio y comparación sobre las diferentes opciones disponible en la actualidad acerca de la infraestructura VoIP.

- **Diseño**

- **Diseño de esquemas de red**

Desarrollo de la interconexión de los diferentes elementos de red utilizados en este proyecto, estudio de la compatibilidad hardware.

- **Diseño y distribución de servidores**

Desarrollo de las funcionalidades de los servidores VoIP y estructuración de parámetros como extensiones, troncales y comportamiento autónomo.

- **Instalación y puesta en marcha**

- **Instalación hardware**

Instalación de todos los dispositivos hardware necesarios para la interconexión entre servidores.

- **Instalación Software**

Configuración de los sistemas operativos, servidores VoIP, troncales y enrutamiento .

- **Pruebas**

- **Pruebas de funcionamiento**

Testeo de las diferentes soluciones anteriormente configuradas, realización de llamadas internas, externas y conferencias.

Análisis de protocolos VoIP y estudio del comportamiento de las llamadas.

- **Memoria**
 - **Redacción de la memoria**

Redacción de todo el contenido anteriormente descrito, documentado desde el comienzo del proyecto y desarrollado simultáneamente con la realización del mismo.

- **Revisión de la memoria**

Revisión de la documentación en cuestiones de presentación, índice, numeración y normalización.

La realización del proyecto se ha llevado a cabo durante cinco meses y medio en los cuales se ha seguido un procedimiento de ejecución desde un punto de vista teórico hasta llevarlo a la práctica.

A continuación veremos un diagrama en el cual se indican la distribución del tiempo en función a las tareas.

Las tareas del proyecto en las que se han invertido más tiempo han sido el diseño y la instalación y puesta en marcha ya que se encontraron dificultades a la hora de compatibilizar sistemas.

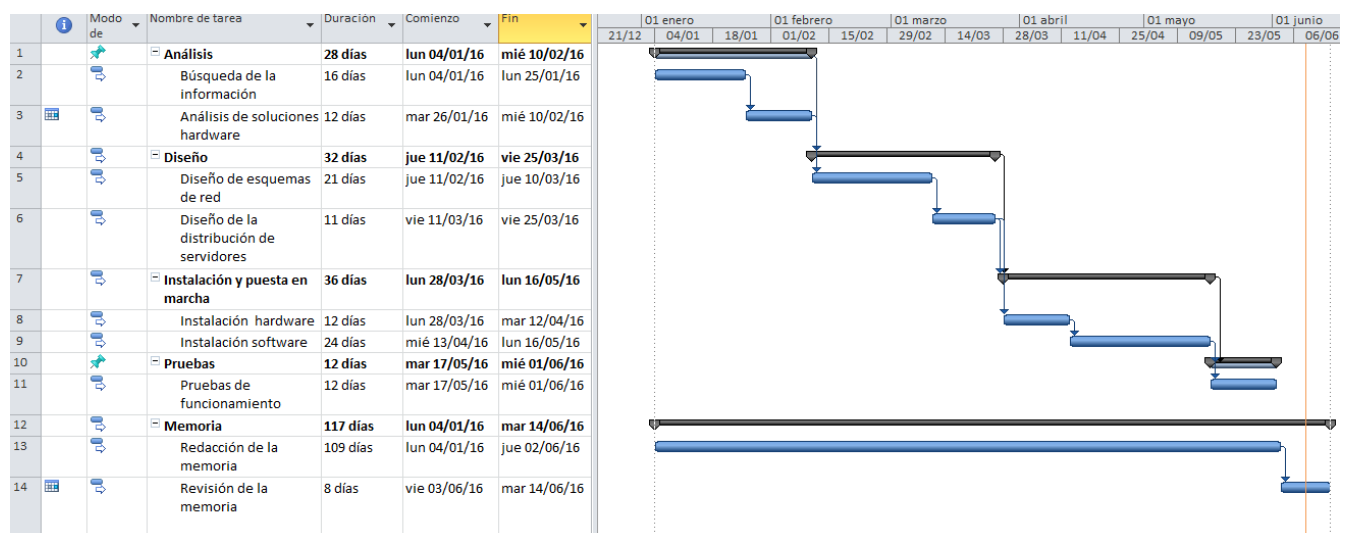


Ilustración 66: Diagrama de Gantt

5.2. Estimación de costes

5.2.1. Presupuesto

Para la realización del presupuesto tenemos que contemplar tres factores principales, el personal necesario, el gasto en materiales y servicios contratados.

Para llevar a cabo este proyecto ha sido necesarios los siguientes perfiles:

- **Jefe de proyecto**

Persona encargada de supervisar las tareas que realiza el resto del equipo y garantizar que se cumplen los objetivos y plazos establecidos. El jefe de proyecto será la figura encargada de la redacción de la memoria final.

- **Analista**

El trabajo que desempeña este perfil de trabajador es identificar los requisitos técnicos necesarios en el proyecto que posteriormente serán proporcionados al diseñador.

- **Diseñador**

Persona encargada de crear una solución viable para el proyecto, basándose en la información anteriormente recopilada por el analista.

- **Técnico**

Persona que lleva a cabo la instalación y configuración de todos los equipos definidos en pasos previos

- **Tester**

Este perfil de trabajador se encarga de comprobar la configuración del sistema, realizando todo tipo de pruebas en busca de fallos, es el paso previo a la entrega del producto al cliente, cuanto más exhaustivo sea el trabajo del tester, menos problemas encontrará el usuario final.

A continuación se muestra una tabla con los costes en personal.

Trabajador	Coste/ hora	Horas totales	Coste total
Jefe de proyecto	50€	25	1250€
Analista	25€	72	1800€
Diseñador	25€	86	2150€
Técnico	20€	254	5000€
Tester	12€	46	552€
TOTAL	-	483	10.752€

Además de recursos humanos, necesitaremos adquirir diferentes materiales para la realización del proyecto.

Los materiales a utilizados en el proyecto han sido:

Ordenador portátil de gama media con las siguientes características,

- Procesador Intel Core i5 M430.
- 6 Gb de Memoria RAM.
- 500 Gb de disco duro

Un ordenador portátil con unas características similares puede costar actualmente unos 550€.

La vida útil de estos equipos es de 5 años aproximadamente, teniendo en cuenta que el proyecto ha sido desarrollado durante 5 meses y medio, obtenemos que el coste de este equipo durante la realización del mismo es de 50,4€.

Teléfono Móvil de gama media con sistema operativo Android 5.0 utilizado para la realización de pruebas inalámbricas, el valor de este dispositivo es de 186€, de los cuales sólo hemos abonado para el proyecto 17,1€.

Para el resto de materiales hemos respetado la misma amortización de los equipos informáticos.

Producto	Unidades	Coste total	Coste durante el proyecto
Ordenador portatil	1	550€	50,4€
SmartPhone	1	186€	17,1€
Router	1	286€	26,2 €
Switch	1	190€	17,4€
Teléfonos VoIP	8	752€	68,9€.
Gateway FXS	1	36€	3,3€
Gateway FXO	1	62€	5,7€
Gateway GSM	1	17€	1.5€
Raspberry Pi 1			
Mod. B+	1	25€	2,30€
Raspberry Pi 2			
Mod. B	1	35€	3,20€
Tarjeta SD	2	12€	1,1€
Transformadores			
RPi	2	16€	1.5€
TOTAL		2167€	198,6€

En el apartado de servicios contemplamos todos los gastos mensuales asociados a los servicios contratados.

Servicio	Coste mensual	Coste total en el proyecto
Internet +Teléfono	52€	286€
Troncal SIP	2,5€	13.8€
Línea móvil	9€	49.5€
TOTAL	63.5€	349,6€

En el coste de estos servicios no se han incluido las llamadas realizadas a través de las diferentes troncales, ya que no es un gasto fijo, en función del uso puede variar.

Realizando la suma total de los costes:

Recurso	Coste Total	Coste para el proyecto
Gasto en personal	10.752€	10.752€
Gasto en materiales	2167€	198.6€
Gasto en servicios	349.6€	349.6€
TOTAL	13.268,6€	11.300,2€

El coste total del proyecto asciende hasta 11.300,2€ ya que la utilización de materiales se ha realizado durante 5,5 meses.

La implantación de este sistema en una pequeña o mediana empresa podría suponer una inversión importante, sin embargo, gracias al ahorro en costes mensuales de telefonía, esta opción sería viable.

Para una empresa multinacional con un gran volumen de llamadas al extranjero este proyecto es imprescindible puesto que el coste de las llamadas internacionales a través de Troncales SIP son mucho más económicas que por medio de líneas analógicas.

Este proyecto podría haberse reducido mucho el coste utilizando los ordenadores de cada uno de los empleados como teléfonos VoIP instalando un SoftPhone, ya que uno de los mayores gastos de materiales son estos dispositivos.

Otro dato a tener en cuenta es que la mayoría de los materiales empleados cumplen una doble función, ya que el router, switch, ordenador portátil y teléfonos móviles pueden ser utilizados en otra tarea distinta a este proyecto.

6. Conclusiones

En este capítulo definiremos todas las conclusiones que hemos obtenido a lo largo del desarrollo del este proyecto. Una vez realizadas todas las pruebas pertinentes relacionadas con Raspberry Pi y telefonía VoIP hemos llegado a la conclusión de:

- **Ventajas**

La placa Raspberry Pi es una solución muy económica y fiable para ser utilizada como servidor VoIP por particulares o empresas con un volumen de llamadas medio.

Teniendo en cuenta las características de la placa y los pocos recursos necesarios para hacer funcionar el sistema, la impresión percibida es bastante positiva.

Podríamos decir con total seguridad que con este dispositivo es posible crea una centralita de llamadas VoIP para gestionar un máximo de 6 llamadas simultáneas (comprobadas en laboratorio) y registrar más de 30 extensiones.

Gracias a la unión de dispositivos de bajo coste como Raspberry Pi, modem GSM y Gateway FXO hemos sido capaces de crear un producto muy útil para todo tipo de público y siendo conscientes que actualmente en el mercado existen productos similares por un precio 4 veces superior al de nuestro proyecto.

Personalmente me siento muy satisfecho de haber conseguido configurar correctamente todos los dispositivos con los que cuenta este proyecto ya que teniendo bien afianzados los conocimientos teóricos, el sistema no siempre tiene el mismo comportamiento una vez llevado a la práctica.

- **Inconvenientes**

Con las características de nuestra placa, no es posible gestionar conferencia de voz con más de 3 participantes.

Para utilizar este proyecto en ambientes empresariales sería necesario dotarlo de un mejor sistema de seguridad para evitar accesos indeseados.

En el apartado de conferencia este sistema no destaca ya que, siendo reiterativo, las especificaciones de la placa no lo permiten. En el laboratorio fuimos capaces de mantener una videoconferencia entre dos terminales móviles pero rápidamente el procesador la Raspberry Pi se saturaba.

Al ser un dispositivo con un tamaño tan reducido no es posible ampliar puertos o insertar nuevas tarjetas de expansión, como soluciones en equipos de sobremesa, todos los dispositivos que necesitemos añadir a esta placan deben ser configurados externamente como en el caso del Gateway FXO.

En el caso de un corte eléctrico todos los dispositivos asociados a este proyecto dejaría de funcionar y por lo tanto el servicio de llamadas no estaría disponible. Asociados a este proyecto existen tres tipos de respaldos, de los cuales solo hemos solventado uno de ellos por cuestiones económicas.

- Respaldo en troncales
- Respaldo en conexión a Internet
- Respaldo en el suministro eléctrico

Los fallos en las troncales han sido solventados con sistemas redundantes en las troncales.

Los fallos en el suministro de Internet se solventaría contratando una segunda conexión a Internet con diferente proveedor a la primera e instalando un equipo de balanceador de carga para evitar cortes en el suministro.

Por últimos los fallos en el sistema eléctrico se solventarían con un equipo SAI (Sistema de Alimentación Ininterrumpida) el cual proporcione corriente eléctrica a todos los dispositivos mientras el suministro eléctrico sea reestablecido.

7. Líneas de Futuro

- **Servidor VoIP de alta disponibilidad**

Como continuación de este proyecto nos habría gustado proporcionar un sistema redundante compuesto por dos Raspberry Pi con el que crear un sistema de alta disponibilidad.

Trataría en clonar la misma configuración en ambas Raspberry Pi y conectarlas a un tercer equipo balanceador que compruebe constantemente si los servidores proporcionan el servicio de telefonía, uno de ellos estaría en producción mientras el segundo se mantendría en estado latente y en el caso de fallar el primero, el segundo servidor entraría en juego de manera totalmente automática.

Con este tipo de instalaciones estamos proporcionando un servicio mucho más fiable y orientado al sector empresarial.

- **Solución de comunicación unificada**

En este proyecto hemos instalado y configurado una serie de dispositivos totalmente independientes unos de otros, por lo tanto ha sido necesario adquirir cada dispositivo por separado y posteriormente configurarlos entre ellos.

Como solución alternativa y proyectos futuros me gustaría unificar todos los dispositivos en un mismo hardware para facilitar la interconexión de las diferentes troncales.

En el mismo hardware sería viable unificar la placa Raspberry Pi, Gateway GSM y Gateway FXO, para crear una centralita VoIP con diferentes opciones de enrutamiento de llamadas, tal y como demostramos en este proyecto.

Todos los dispositivos con los que hemos desarrollado este proyecto trabajan bajo licencias GPL de software libre, es decir, que podemos hacer uso de ellas sin coste alguno pero no podemos obtener beneficio ya que su adquisición ha sido gratuita.

Bibliografía

- [1] **Internet RDSI:** Sistemas de telefonía, José Manuel Huidobro Moya, Rafael Conesa Pastor 5ª edición, Editorial Paraninfo.
- [2] **VPN y VLAN:** Redes de computadoras, 5ta Edición – Andrew S. Tanenbaum y David J. Wetherall.
- [3] **QoS:** Redes e Internet de alta velocidad. Rendimiento y calidad de servicio William Stallings, editorial Pearson.
- [4] **Teléfonos Cisco Linksys SPA 512 G:** http://www.cisco.com/c/en/us/products/collateral/collaboration-endpoints/spa512g-1-line-gige-ip-phone/c78-698950_data_sheet.html
- [5] **Gateway FXS y FXO:** VoIP y Asterisk, Redescubriendo la Telefonía, Editorial Ra-Ma
- [6] **Protocolos VoIP:** Asterisk the future of the telephony 2º Edición, Jim Van Meggelen, Leif Madsen, and Jared Smith, editorial O'Reilly
- [7] **Diferencias entre IAX y SIP:** <http://elastixtech.com/protocolo-iax/>
- [8] **RTP Y RTCP:** VoIP y Asterisk Redescubriendo la telefonía, Julio Gómez López y Francisco Gil Montolla, editorial RA-MA 2008
- [9] **Códecs de audio:** Asterisk the future of the telephony 2º Edición, Jim Van Meggelen, Leif Madsen, and Jared Smith, editorial O'Reilly
- [10] **Códec de video:** Multimedia Systems, Parag Havaladar, Gerard Medioni (Capítulo 8)
- [11] **Historia de Raspberry Pi:** <http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>
- [12] **Modelos Raspberry Pi:** <https://www.raspberrypi.org/products/>
- [13] **Comparativa entre Raspberry Pi:** <http://www.xataka.com/componentes/nueva-raspberri-pi-vs-sus-competidores-buscando-el-mejor-mini-ordenador-barato>
- [14] **Asterisk:** <http://www.asterisk.org/get-started>
- [15] **Historia Asterisk:** http://comunidad.asterisk-es.org/index.php?title=Introduccion_a_Asterisk
- [16] **FrePBX:** <https://www.freepbx.org/>
- [17] **Elastix:** <http://www.elastix.org/>
- [18] **WireShark:** https://www.wireshark.org/docs/wsug_html_chunked/
- [19] **Instalación Elastix:** <http://www.opendireito.com/2013/07/13/probando-uelastix-1/>
- [20] **uElastix.img:** <http://dostmuhammad.com/uelastix-for-raspberry-pi-img-included/>
- [21] **Elastix MicroUCS:** <http://www.elastix.com/portfolio-item/microucs/#tab-id-3>