

UACM

Universidad Autónoma
de la Ciudad de México

Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE
TELECOMUNICACIONES

Sistemas de Telefonía IP basado en Asterisk y Raspberry Pi 3

T E S I S

QUE PARA OPTAR POR EL TÍTULO DE
LICENCIADO EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE
TELECOMUNICACIONES

P R E S E N T A :

DIEGO AXEL CORTÉS RUIZ

D I R E C T O R

M. EN I. OSCAR RENÉ VALDEZ CASILLAS

Ciudad de México, abril de 2021

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS ©

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

Agradecimientos.

Agradezco a mi padre Hugo, madre Olivia y hermano Armando por el amor, dedicación y paciencia con la que cada día se preocupaban por mi formación académica, siendo mis padres los principales promotores de que mis sueños se cumplieran.

Gracias a mis padres por acompañarme en cada paso dado durante mi trayecto de vida, anhelando siempre lo mejor para mí. Estimo sus consejos, palabras y apoyo en cada acción mal tomada en mis tiempos difíciles.

También agradezco a mis profesores y compañeros, debido a sus enseñanzas y consejos pude lograr una meta más en mi vida.

Contenido

Índice de Figuras.....	4
Índice Tablas	12
Justificación	14
Objetivos.....	15
Introducción	16
Capítulo 1 Red telefonía	17
1. Historia de la telefonía.....	17
1.1 Red Telefónica Pública Conmutada (PSTN).....	21
1.2 Sistemas de Conmutación	24
1.2.1 El aparato telefónico	28
1.2.2 Señalización.....	35
1.3 Digitalización de la telefonía.....	37
1.3.1 Filtrado	38
1.3.2 Muestreo.....	38
1.3.3 Cuantificación	39
1.3.4 Compresión y codificación.....	41
1.4 Convivencia de la telefonía y las redes de datos.....	45
1.4.1 RDSI o ISDN	46
1.4.2 Tecnologías DSL.....	51
Capítulo 2 Red de datos y telefonía IP	54
2. Redes de datos.....	54
2.1 Historia.	54
2.2 Medios físicos para la transmisión de datos.....	57
2.2.1 Par trenzado	58
2.2.2 Cable Coaxial	61
2.2.3 Líneas Eléctricas	62
2.2.4 Fibra Óptica (FO)	63
2.2.5 Transmisión inalámbrica.	66
2.2.6 Espectro electromagnético.....	67
2.3 Modelo de referencia OSI de ISO.....	68
2.3.1 Capa Física.....	70
2.3.2 Capa de Enlace de datos	70

2.3.3 Capa de Red	71
2.3.4 Capa de Transporte	73
2.3.5 Capa de Sesión	75
2.3.6 Capa de Presentación	75
2.3.7 Capa de Aplicación.....	75
2.4 Protocolos TCP/IP (Transmission Control Protocol e Internet Protocol) ..	78
2.4.1 La Capa de Enlace	79
2.4.2 La Capa de Interred	79
2.4.3 La Capa de Transporte.....	80
2.4.4 La Capa de Aplicación	82
2.5 Voz sobre IP (VoIP).	83
2.5.1 Componentes de la red VoIP.	84
2.5.2 Protocolos VoIP.....	86
2.6 Consideraciones de calidad de servicio (QoS) para VoIP y ToIP.....	97
Capítulo 3 Asterisk y Raspberry Pi.....	102
3. Características de un sistema de telefonía IP	102
3.1 Extensiones telefónicas por software.....	102
3.1.1 Extensiones en servidor	103
3.1.2 Extensiones en terminales	105
3.2 Asterisk.....	106
3.2.1 Historia	107
3.2.2 Sistemas Operativos donde se puede instalar Asterisk.	108
3.2.3 Requerimientos de Hardware.	109
3.2.4 Tarjetas especializadas.....	112
3.3 Opciones Comerciales de sistemas de telefonía IP. Opciones de tipo Open Source. Comparación.....	114
3.4 Raspberry Pi 3.....	117
3.4.1 Arquitectura de Hardware.....	120
3.4.2 Arquitectura ARM.	121
3.4.3 Sistemas Operativos soportados.....	122
3.4.4 Puertos utilizados para conectar un servidor de telefonía.	126

Capítulo 4. Implementación de un sistema de telefonía IP con Raspberry Pi y Asterisk.....	128
4 Implementación.....	128
4.1 Descarga e instalación del sistema operativo.	128
4.2 Configuración del Firewall y servicio SSH.	130
4.3 Instalación y configuración del Servicio Asterisk	133
4.4 Configuración Radius.....	138
4.5 Extensiones de llamadas y mensajería.....	140
4.6 Aplicación cliente.	145
4.7 Administración.....	153
4.8 Manual de usuario.	158
Capítulo 5 Conclusiones	160
Apéndice. Comandos de configuración.....	162
Bibliografía	166

Índice de Figuras.

Capítulo 1

Figura 1.1 Teletrófono de Meucci.....	17
Figura 1.2. Composición interna del Teletrófono de Meucci.....	18
Figura 1.3. Teléfono de Reis.....	18
Figura 1.4. Operadoras de una central telefónica de 1878.....	19
Figura 1.5. Teléfono con Disco de Marcado.....	20
Figura 1.6 Topología inicial de la red telefónica.....	22
Figura 1.7 Red jerárquica de telefonía.....	22
Figura 1.8 Red Local de telefonía.....	23
Figura 1.9 Central Regional e Internacional.....	23
Figura 1.10 Conmutación de circuitos.....	24
Figura 1.11 Conmutación de paquetes.....	25
Figura 1.12 Transmisión por datagramas.....	26
Figura 1.13 Circuito Virtual entre las terminales A y D.....	27
Figura 1.14 Sistema telefónico.....	29
Figura 1.15 Circuito telefónico simple.....	29
Figura 1.16 Comportamiento interno de un micrófono de carbón.....	30
Figura 1.17 Composición del auricular.....	31

Figura 1.18 Ubicación de la bobina de inducción.....	31
Figura 1.19 Conmutador de gravedad.....	32
Figura 1.20 Pulsos de marcado.....	33
Figura 1.21 Disco de marcado.....	34
Figura 1.22 Tonos DTMF.....	35
Figura 1.23 Señalización CAS y CCS.....	36
Figura 1.24 Proceso de la digitalización.....	37
Figura 1.25 Señales PAM.....	39
Figura 1.26 Muestreo Real y Muestreo Ideal.....	39
Figura 1.27 Señal cuantificada.....	40
Figura 1.28 Error de cuantificación.....	40
Figura 1.29 Curva de Ley A.....	42
Figura 1.30 Curva de Ley μ	42
Figura 1.31 Bits de palabra PCM.....	43
Figura 1.32 Ubicación de segmento.....	44
Figura 1.33 Ubicación de intervalos.....	45
Figura 1.34 Red Digital de Servicios Integrados.....	46
Figura 1.35 Grupos funcionales y Puntos de Referencia.....	47
Figura 1.36 Interfaz BRI (2B+D).....	49

Figura 1.37 Conexión BRI (2B+D).....	49
Figura 1.38 Interfaz PRI (T1 Y E1).....	50
Capítulo 2	
Figura 2.1 Red ARPANET en 1973.....	55
Figura 2.2 Red NSFNET en 1992.....	56
Figura 2.3 Medios de transmisión de datos.....	58
Figura 2.4 Cable coaxial.....	61
Figura 2.5 Medio compartido por cable eléctrico.....	62
Figura 2.6 Fibra óptica.....	64
Figura 2.7 Fibra óptica – Multimodo.....	64
Figura 2.8 Fibra óptica – Multimodo de índice gradual.....	65
Figura 2.9 Fibra óptica - Monomodo.....	65
Figura 2.10 Atenuación de la fibra óptica.....	66
Figura 2.11 Señales de radio.....	67
Figura 2.12 Modelo de referencia OSI.....	69
Figura 2.13 Relación entre paquetes y tramas.....	70
Figura 2.14 Comunicación lógica de la capa de transporte.....	74
Figura 2.15 Protocolo HTTP.....	76
Figura 2.16 Arquitectura del servicio de correo electrónico.....	77

Figura 2.17 Capas del modelo TCP/IP.....	78
Figura 2.18 Protocolos por capa.....	79
Figura 2.19 Protocolo IP.....	80
Figura 2.20 Protocolo TCP. Ventana = 1 paquete.....	81
Figura 2.21 Protocolo UDP.....	82
Figura 2.22 Flujo de paquetes de voz.....	83
Figura 2.23 Flujo de paquetes de voz con una PBX.....	84
Figura 2.24 Función de una PBX.....	85
Figura 2.25 Protocolos en una red H.323.....	87
Figura 2.26 Proceso de llamada H.323.....	88
Figura 2.27 Protocolos usados en la red SIP.....	89
Figura 2.28 Función de Servidores proxy como Cliente y Servidor.....	93
Figura 2.29 Servidor proxy como Cliente y Servidor.....	93
Figura 2.30 Servidor de Registro.....	94
Figura 2.31 Servidor de Redireccionamiento.....	94
Figura 2.32 Reserva del ancho de banda.....	100

Capítulo 3

Figura 3.1 Registro de extensiones.....	102
Figura 3.2 Extensiones (Asterisk).....	103

Figura 3.3 Logo del programa Asterisk.....	106
Figura 3.4 Logo del programa Digium.....	108
Figura 3.5 Logo del programa Sangoma.....	108
Figura 3.6. Servidor Asterisk (Pasatiempo).....	110
Figura 3.7 Logo del software Elastix.....	114
Figura 3.8 Logo del software 3CX.....	114
Figura 3.9 Logo del software sipXcom.....	115
Figura 3.10 Logo del software FreeSWITCH.....	115
Figura 3.11 Logo del software OpenSIPs.....	115
Figura 3.12 Logo del software Kamailio.....	116
Figura 3.13 Logo del software Issabel.....	116
Figura 3.14 Logo de la Raspberry Pi.....	118
Figura 3.15 Raspberry Pi 3 modelo B.....	121
Figura 3.16 BCM2837 de Raspberry Pi.....	122
Figura 3.17 Logo del Sistema Operativo Raspbian.....	122
Figura 3.18 Logo del Sistema Operativo Ubuntu Mate.....	123
Figura 3.19 Logo del Sistema Operativo Ubuntu.....	124
Figura 3.20 Logo del Sistema Operativo Risc OS.....	125
Figura 3.21 Logo del Sistema Operativo LibreElec.....	125

Figura 3.22 Logo del Sistema Operativo Windows 10 IoT Core.....	126
Figura 3.23 Logo del Sistema Operativo Arch Linux ARM.....	126
Figura 3.24 Periféricos usados por un PBX.....	127
Capítulo 4	
Figura 4.1 Ubuntu MATE.....	128
Figura 4.2 Inicio de instalación del Sistema Operativo 0%.....	129
Figura 4.3 Inicio de instalación del Sistema Operativo 99%.....	129
Figura 4.4 Comando iptables –L.....	130
Figura 4.5 Activación y verificación del puerto 22.....	130
Figura 4.6 Generar contraseña.....	131
Figura 4.7 Estado del servicio SSH.....	131
Figura 4.8 Logo del software Putty.....	132
Figura 4.9 Ingreso inicial a la RBP 3 con dirección IP y puerto por medio de Putty.....	132
Figura 4.10 Ingreso a la RBP 3 con Usuario y contraseña por medio de Putty.....	133
Figura 4.11 Versión de Asterisk.....	134
Figura 4.12 Paqueterías de Asterisk a instalar.....	137
Figura 4.13 Error de ubicación del archivo radiusclient.conf.....	138
Figura 4.14 Enrutamiento correcto del archivo radiusclient.conf.....	139

Figura 4.15 Estado activo y correcto de Asterisk.....	139
Figura 4.16 Directorios de Asterisk.....	140
Figura 4.17 Comando de configuración de llamadas en el archivo sip.conf.....	141
Figura 4.18 Comando en el archivo sip.conf para mensajería.....	141
Figura 4.19 Plantilla con el tipo de servicio de llamada, tipo de host, tipo de tono y el contexto para mensajería.....	142
Figura 4.20 Registro de Extensiones en el archivo sip.con.....	142
Figura 4.21 Consola de Asterisk.....	143
Figura 4.22. Información de consola de Asterisk.....	143
Figura 4.23 Registro de las Extensiones en consola de Asterisk.....	143
Figura 4.24 Extensiones para llamadas en el archivo extensions.conf.....	144
Figura 4.25 Extensiones para mensajes en el archivo extensions.conf.....	144
Figura 4.26 Extensiones de mensajería con serie consecutiva de acciones.....	145
Figura 4.27 Registro y petición de la extensión en Smartphone.....	146
Figura 4.28 Dirección IP del servidor en Smartphone.....	146
Figura 4.29 Confirmación de la extensión y de la dirección IP como servidor en Smartphone.....	146
Figura 4.30 Protocolo de envío de datos usado: UDP.....	147
Figura 4.31 Registro de la Extensión101 en consola de Asterisk.....	147

Figura 4.32 Registro y petición de la extensión en Laptop.....	147
Figura 4.33 Dirección IP del servidor en Laptop.....	148
Figura 4.34 Confirmación de la extensión y de la dirección IP como servidor en Laptop.....	148
Figura 4.35 Protocolo de envío de datos usado: UDP.....	148
Figura 4.36 Registro de la Extensión 103 en consola de Asterisk.....	149
Figura 4.37 Comprobación correcto del registro de Extensiones en consola de Asterisk.....	149
Figura 4.38 Extensión 101 realizando una llamada a la extensión 102.....	149
Figura 4.39 Llama en curso entre las Extensiones 101 y 102.....	150
Figura 4.40 Extensión 103 realizando una llamada a la extensión 101.....	150
Figura 4.41 Llama en curso entre las Extensiones 101 y 103.....	151
Figura 4.42 Información de acciones en Consola de Asterisk de una llamada desviada.....	151
Figura 4.43 Mensajes entre las Extensiones 101 y 102.....	152
Figura 4.44 Información de acciones en Consola de Asterisk en el envío de mensajes.....	152
Figura 4.45 Registro y petición de la extensión.....	158
Figura 4.46 Confirmación de la extensión y de la dirección IP como servidor.....	158
Figura 4.47 Error en la extensión o error dirección IP como servidor.....	159

Índice Tablas

Tabla 1.1 Comparación de la Conmutación de Circuitos y la Conmutación de Paquetes.....	28
Tabla 1.2 Teclado multifrecuencia.....	34
Tabla 1.3 Codificación de segmentos.....	43
Tabla 1.4 Codificación de intervalos.....	44
Tabla 2.1 Tipos de cables de par trenzado.....	60
Tabla 2.2 Características de transmisión del cable coaxial.....	62
Tabla 2.3 Parte I. Uso del espectro electromagnético.....	67
Tabla 2.3 Parte II. Uso del espectro electromagnético.....	68
Tabla 2.4 Respuestas SIP.....	91
Tabla 2.5 Anchos de banda dependiendo el Códec.....	98
Tabla 3.1 Acciones en las extensiones (Asterisk).....	104
Tabla 3.2 Parte I. Aplicaciones VoIP	105
Tabla 3.2 Parte II. Aplicaciones VoIP	106
Tabla 3.3 Opciones de sistemas operativos para Asterisk.....	109
Tabla 3.4 Recomendaciones para la implementación de Asterisk.....	110
Tabla 3.5 Tarjetas analógicas Digium.....	113
Tabla 3.6 Tarjetas digitales Digium.....	113

Tabla 3.7 Parte I. Comparación entre IP PBX comerciales.....	116
Tabla 3.7 Parte II. Comparación entre IP PBX comerciales.....	117
Tabla 3.8 Parte I. Características Raspberry Pi.....	118
Tabla 3.8 Parte II. Características Raspberry Pi.....	119
Tabla 3.8 Parte III. Características Raspberry Pi.....	120
Tabla 4.1 Tipos de comando para type.....	155

Justificación

En el Laboratorio de Telecomunicaciones de la Universidad Autónoma de la Ciudad de México en el plantel de Casa Libertad se encuentra material parcialmente obsoleto y con poco aporte académico de los mismos para los estudiantes.

Este proyecto de tesis se realizará para agilizar y facilitar la explicación hacia los alumnos de dicha universidad, los diversos temas de las siguientes materias:

- Comunicaciones Analógicas y Digitales.
- Líneas de Transmisión.
- Sistemas de Telefonía.
- Aplicaciones con microprocesadores y microcontroladores.
- Estándares en redes de área local y amplia. TCP/IP.
- Procesamiento digital de señales.
- Topologías y diseño de redes.

Objetivos

- Se elaborará una guía que pueda ser utilizada por los estudiantes en los cursos de “Sistemas Telefónicos”, “Estándares de área local y amplia; TCP/IP” y “Topologías y diseño en redes” para la implementación de un sistema de telefonía IP basado en software **Open Source** (código abierto) disponible en el Laboratorio de Telecomunicaciones para su descarga, configuración y uso utilizando un sistema embebido como equipo de cómputo.
- Se establecerá con el sistema desarrollado, la posibilidad de implantar sobre una red de área local un sistema de telefonía con los medios para usar terminales tipo **Softphone** (terminal telefónica para computadora y en Smartphone) y conexión a dispositivos móviles que tengan capacidad de conexión a la red IP de forma inalámbrica.

Introducción

La intención inicial de esta tesis es aportar las bases fundamentales de la tecnología de redes, telefonía y el trabajo en conjunto entre ambas: Voz sobre el Protocolo de Internet (VoIP), así como brindar una perspectiva de las posibles alternativas de software del tipo Open Source y los diferentes microordenadores de Raspberry Pi. Se presenta el software Asterisk trabajando dentro de una Raspberry Pi 3 modelo B como base para un sistema de telefonía IP funcional para el Laboratorio de Telecomunicaciones en el plantel Casa Libertad de la Universidad.

El Capítulo 1 proporciona una descripción de los conceptos de telefonía, que servirán de apoyo para los posteriores capítulos de la presente tesis; en él se detalla el funcionamiento de la red de telefonía, desde cada elemento dentro de la red como es el aparato telefónico y las centrales de conmutación hasta el proceso que se realiza en la red para el establecimiento de la comunicación de voz.

En el Capítulo 2 se estudian los conceptos de redes de datos, en donde se explican los diferentes medios de transmisión, los cuales funcionan para establecer las diferentes tecnologías de comunicación existentes, además de las diferentes capas del modelo de referencia OSI y las del Protocolo TCP/IP. Por último dentro de este capítulo, se expone los elementos usados para la tecnología Voz sobre el Protocolo de Internet, así como los protocolos y calidad del servicio.

Tras la introducción en conceptos de telefonía y redes de datos en los capítulos anteriores, el Capítulo 3 brinda la historia del desarrollo del software Asterisk con sus diferentes características y sugerencias para emprender un sistema de VoIP, como son los diferentes sistemas operativos que soportan el software, extensiones, requerimientos del hardware y tarjetas especializadas. Este capítulo también ofrece un panorama amplio de las diferentes alternativas de software del tipo Open Source y hardware (Raspberry Pi) con sus diferentes sistemas operativos soportados. Así, logrando desarrollar un sistema de Telefonía IP activo y personalizado.

Finalmente en el Capítulo 4 se expone la implementación, configuración y activación por líneas de código del software Asterisk trabajando como servidor en una Raspberry Pi 3 modelo B, así como los manuales para el administrador y el de usuario.

Capítulo 1 Red telefonía

1. Historia de la telefonía.

En 1820 Hans Christian Oersted descubre la relación entre las fuerzas eléctricas y magnéticas, realizando el experimento del movimiento de una aguja imantada en relación al paso de una corriente eléctrica, así, halló la existencia del campo magnético generado por el flujo de corriente eléctrica. Finalmente en ese mismo año André-Marie Ampère formuló las leyes del electromagnetismo.

En 1825 William Sturgeon construyó un electroimán que consistía en un trozo de hierro envuelto por un cable por el que podía fluir corriente eléctrica.

En 1831 Michael Faraday descubre la generación de corriente eléctrica a partir de los campos magnéticos.

En 1835 Michael Faraday establece la Ley Faraday-Henry la cual describe como un campo magnético variable en el tiempo es capaz de producir una **F.E.M.** (Fuerza Electromagnética) y esta a su vez un flujo de corriente sobre el conductor.

En 1837 Joseph Henry descubre que la corriente alterna que circula por un solenoide genera vibraciones.

En 1838 Samuel Morse perfecciona el método por el cual todo símbolo era transmitido individualmente mediante rayas y puntos.

En 1841 Samuel Morse patenta el American Morse Code.

En 1845 Se establecen las Leyes de Kirchhoff basándose en la conservación de la energía y la carga en los circuitos eléctricos por Gustav Robert Kirchhoff.

En 1854 Charles Bourseul primer personaje con la idea de transmitir la voz ocupando los medios electromagnéticos.

Antonio Meuca inventa el **teletrófono** (Figura 1.1 y Figura 1.2)¹ pensado en conectar la oficina que se encontraba en el segundo piso con el dormitorio en donde se encontraba confinada su esposa debido a su padecimiento reumático.

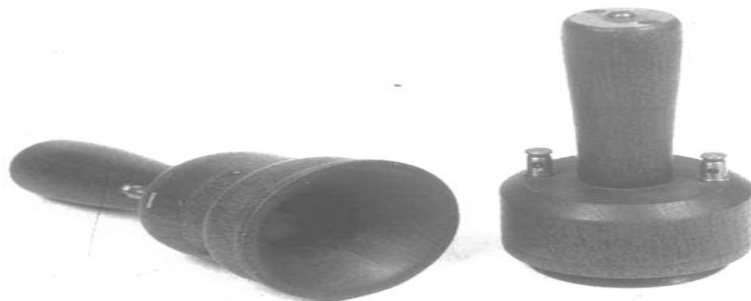


Figura 1.1 Teletrófono de Meucci.

¹ El teléfono se inventó en La Habana: <https://almejeiras.wordpress.com/2012/05/18/el-telefono-se-invento-en-la-habana-antonio-meucci/>

Estructura interna del Teletrófono de Meucci.

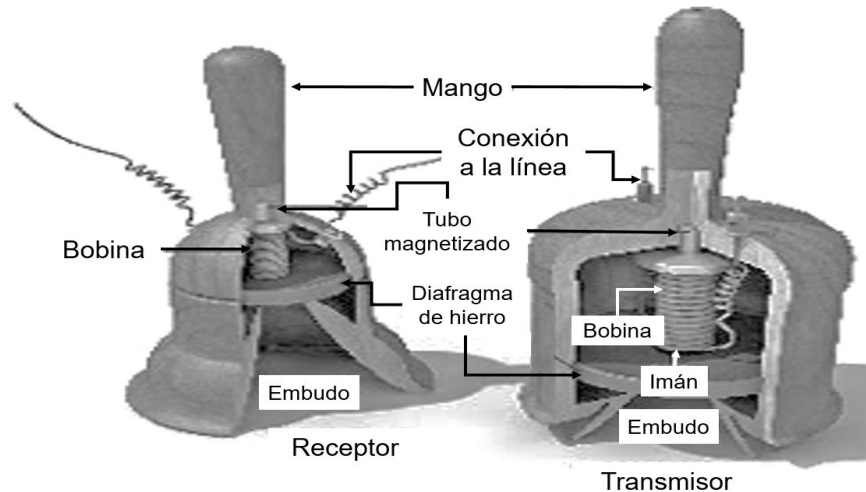


Figura 1.2. Composición interna del Teletrófono de Meucci.

En 1861 James Clerk Maxwell formula cuatro ecuaciones basadas en las leyes de Ampere, Gauss y Faraday conocidas como las Ecuaciones de Maxwell. Ese mismo año, Philipp Reis estudia el aparato auditivo, específicamente la membrana timpánica el cual es capaz de reproducir las ondas de sonido.

En 1864 Philipp Reis presenta por primera vez un artículo relacionado con la telefonía (Telefonía por medio de la corriente eléctrica).

En 1867 Philipp Reis fabrica un aparato capaz de transmitir a una distancia de hasta 100 metros en una sola dirección llamándolo teléfono con un altavoz y una bobina enrollada en una varilla de hierro funcionando como el receptor (Figura 1.3)². En su primera transmisión pronuncio la frase: “**El caballo no come ensalada de pepinos**”.

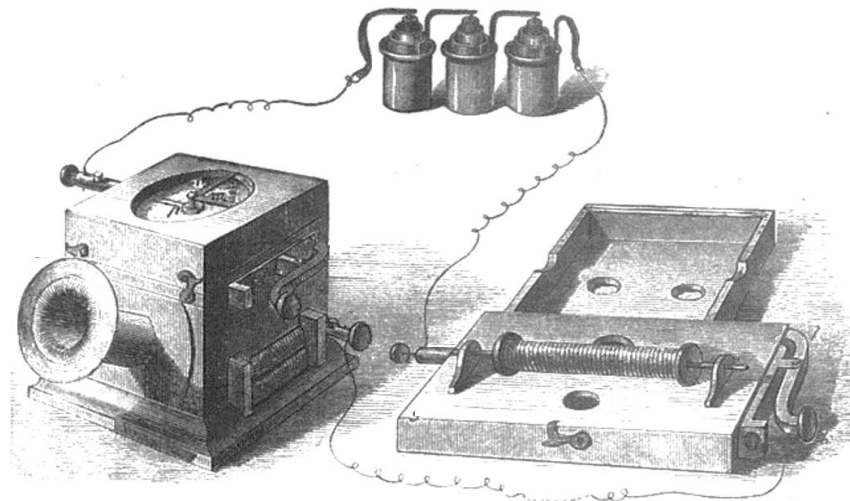


Figura 1.3. Teléfono de Reis.

² Teléfonos: <https://www.timetoast.com/timelines/phones-462d9043-84fb-41b9-97ad-e7629a777eee>

En 1871 Antonio Meucci presenta un **Caveat** (Aviso Patente) el cual es la documentación previa a la Patente y con vigencia de 1 año por una tarifa de \$10, debido a que la Patente tenía una tarifa de \$30 la el cual no podía pagar por falta de dinero.

En 1872 Antonio Meucci presenta su invento al presidente de la NY Telegraph Corporate el cual nunca le devolvió el aparato.

En 1876 Graham Bell obtiene la patente de la invención del teléfono, horas después Elisha Gray presenta documentos para patentar el mismo dispositivo.

En 1877 Henry Hummings elabora y patenta un transmisor telefónico de resistencia variable que usaba carbón molido con un cono de voz unido al diafragma resolviendo los problemas presentes en los transmisores electromagnéticos. Ese mismo año se funda la compañía telefónica Bell Telephone Company.

En 1878 Se inaugura el servicio comercial (Figura 1.4)³ por parte de la compañía Bell Telephone Company la primera central telefónica con un total de 21 abonados en New Haven, Estados Unidos. Funcionando de manera completamente manual para poner en contacto a 2 líneas distintas, la operadora realizaba la conexión de ambos clientes por medio de un cuadro de comunicación simple, derivado de los que se usaban para la telegrafía.

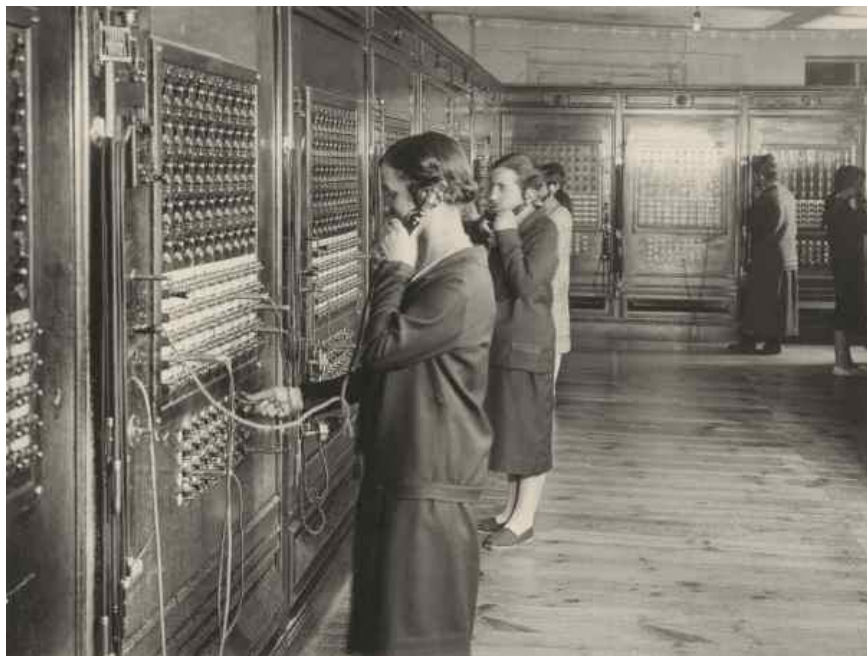


Figura 1.4. Operadoras de una central telefónica de 1878.

En 1886 Almon Strowger elabora un Conmutador de líneas de 5 hilos poco eficiente conocido como Conmutador Paso a Paso, sustituyendo a los operadores telefónicos.

³ Historia de la telefonía en España. Telefonistas en España:
<https://historiatelefonía.com/2015/07/31/telefonistas-en-espana/>

En 1891 Almon Strowger patentó el Conmutador Strowger.

En 1892 Se inaugura la primera Central Automática Telefónica diseñada para un máximo de 99 líneas telefónicas en Indiana, Estados Unidos.

En 1895 Alexander Keith, John Erickson y Charles Erickson logran la patente del Disco de Marcado (Figura 1.5)⁴ de 3 hilos sustituyendo al botón pulsador.



Figura 1.5. Teléfono con Disco de Marcado.

En 1913 La empresa alemana Siemens & Halske AG incorpora a su sistema de telefonía el Tono de Marcado.

En 1921 Se logra la primera conversación telefónica con una distancia de 10,000 Km entre la Isla Catalina, Estados Unidos y la Habana, Cuba.

En 1960 Se incorporó un sistema digital con conmutación de paquetes⁵ a los antiguos sistemas telefónicos con el fin de establecer mejores líneas y conexiones de comunicación.

En 1964 Se brinda el servicio de Marcado por Tonos entre las ciudades de Carnegie y Greensburg pertenecientes al estado de Pensilvania, Estados Unidos.

En 1972 El **CCITT** (Comité Consultivo Internacional Telegráfico y Telefónico) actualmente **ITU-T** (International Telecommunication Union - Telecommunication), define la **RDSI** (Red Digital de Servicios Integrados) o en inglés **ISDN** (*Integrated Services Digital Network*) como un sistema para la transmisión de voz y datos.

En 1973 Martin Cooper en conjunto con la empresa Motorola crean el primer auricular de telefonía móvil inalámbrico del proyecto DynaTAC 8000X con un peso mayor a 2 Kg y un costo de \$1, 000,000 dólares.

⁴ Historia de la telefonía en España. Conmutación automática. Orígenes:

<https://historiatelefonía.com/2018/10/30/la-conmutacion-automatica-origenes/>

⁵ Historia de la telefonía: <https://historiatelefonía.com/2018/10/30/la-conmutacion-automatica-origenes/>

⁶ El hombre que inventó el teléfono celular:

https://www.bbc.com/mundo/ciencia_tecnologia/2010/04/100426_inventor_telefono_celular_pl

En 1979 Se crea la red de comunicaciones móviles comercial⁶ de Primera Generación (**1G**) lanzado por **NTT** (*Nippon Telegraph and Telephone Corporation*) en Japón con servicios de solo voz.

En 1990 Llega la Segunda Generación (**2G**) de comunicaciones móviles con servicios como voz, **SMS** (mensajes cortos, *Short Message Service*), llamada en espera, retención de llamada, roaming internacional, transferencia de llamadas, bloqueo de llamadas e identificación de llamadas.

En 2001 Se obtiene el sistema de Tercera Generación (**3G**) ofreciendo una mayor tasa de transmisión de datos, internet inalámbrico, localización, servicios multimedia de poca calidad, correo electrónico y los servicios que ya proveía la tecnología **2G**.

En 2002 Antonio Meucci es reconocido como el verdadero inventor del aparato telefónico por el Congreso de los Estados Unidos y no Alexander Graham Bell.

En 2010 Se establece la Cuarta generación (**4G**) ofreciendo mayor velocidad, calidad y seguridad en los dispositivos, brindando servicios como telefonía IP, videojuegos, TV móvil, videoconferencia y multimedia.

En 2015 Se establece la idea de la red de Quinta Generación (**5G**) en donde el enrutamiento se basa en las direcciones IP.

1.1 Red Telefónica Pública Conmutada (PSTN).

La **PSTN** (*Public Switched Telephone Network*) o también conocido como **POTS** (Servicio de Telefonía Convencional, *Plain Old Telephone Service*) es una red de conmutación de circuitos, que realiza la interconexión de varias centrales telefónicas a nivel jerárquico en el mundo con el fin de brindar el servicio de voz en tiempo real.

La **PSTN** se encarga de enlazar dentro de un solo canal de comunicación los aparatos telefónicos que desean comunicarse por medio de las centrales telefónicas, dependiendo de la lejanía de ambos clientes.

Históricamente después de que Graham Bell patentara el aparato telefónico comercialmente la demanda creció exponencialmente para adquirir los aparatos, la venta de los teléfonos era en pares y la conexión entre ambos mediante un cable que era necesariamente suministrado por los propios clientes, generando así problemas, pues una persona que contaba con un teléfono tenía que realizar la conexión independiente con cada uno de sus contactos, de esta manera, las ciudades se cubrían con los cableados telefónicos (Figura 1.6).

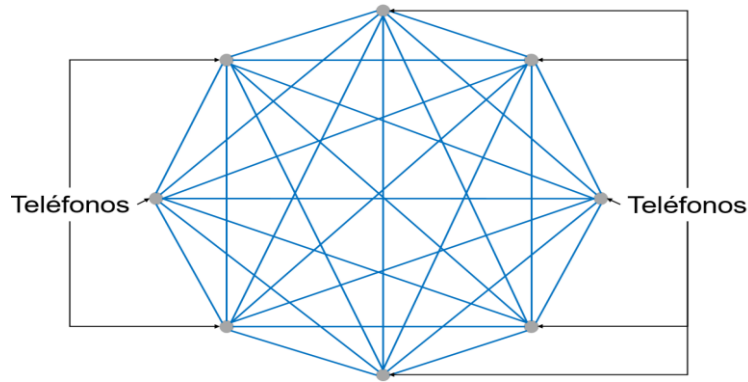


Figura 1.6 Topología inicial de la red telefónica. [Elaboración propia]

La compañía Bell tenía previsto los problemas de la red en donde un teléfono tenía n cables de conexión en la ciudad para n teléfonos como se muestra en la anterior Figura 1.6. Inicializando así un sistema jerárquico de centrales telefónicas (Figura 1.7) en donde se encontraban las oficinas de conmutación de forma manual para posteriormente ser remplazada por centrales de conmutación automática, las cuales son capaces de comunicar entre sí a los diferentes abonados (clientes o teléfonos) de forma automática y sin intervención humana [1.2].

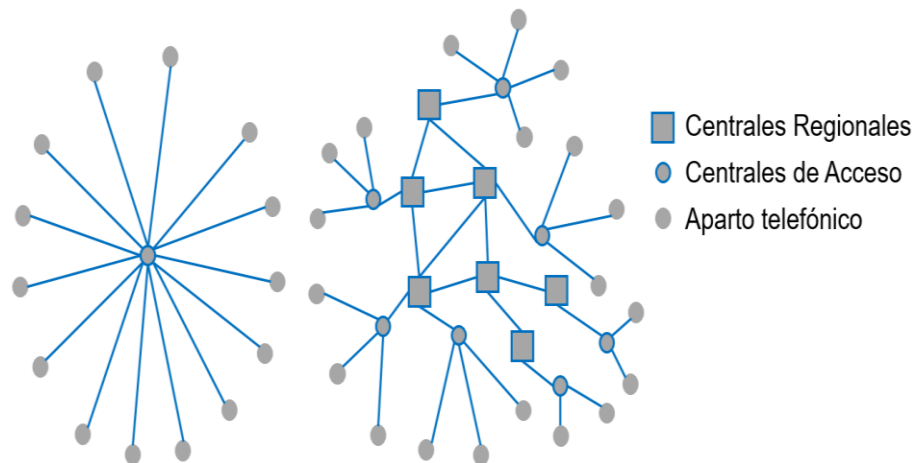


Figura 1.7 Red jerárquica de telefonía. [Elaboración propia]

Para la configuración en la comunicación entre los abonados la **PSTN** consta de los sistemas de transmisión, el sistema de conmutación y los sistemas de señalización. La **PSTN** consta de la siguiente jerarquía.

1. **Centrales Locales u Oficina Final:** Se puede considerar que existen 2 tipos de oficinas:
 - **Centrales de Acceso:** Realizan la conexión de los abonados conectados a la Oficina Final y también lleva acabo la conexión con la red superior en la jerarquía en caso de que la llamada salga del área local.
 - **Centrales Tándem:** Funciona como una red complementaria de tránsito, conectada solo a Centrales de Acceso.

Las conexiones entre el abonado y la Red Local se le conocen como Lazo Local o **Local Loop** (Bucle Local) (Figura 1.8).

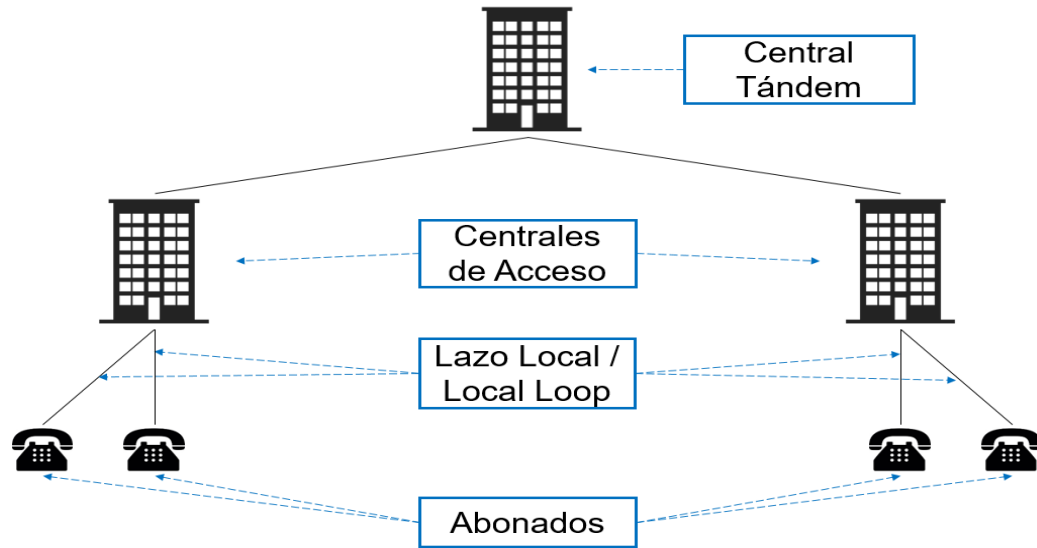


Figura 1.8 Red Local de telefonía. [Elaboración propia]

2. **Redes de Funciones o Centrales Regionales:** Interconectan un grupo de Centrales Locales a una Central Troncal (Figura 1.9).
3. **Red Troncal o Centrales Internacionales:** Proporcionan conexiones de larga distancia a nivel nacional e internacional (Figura 1.9).

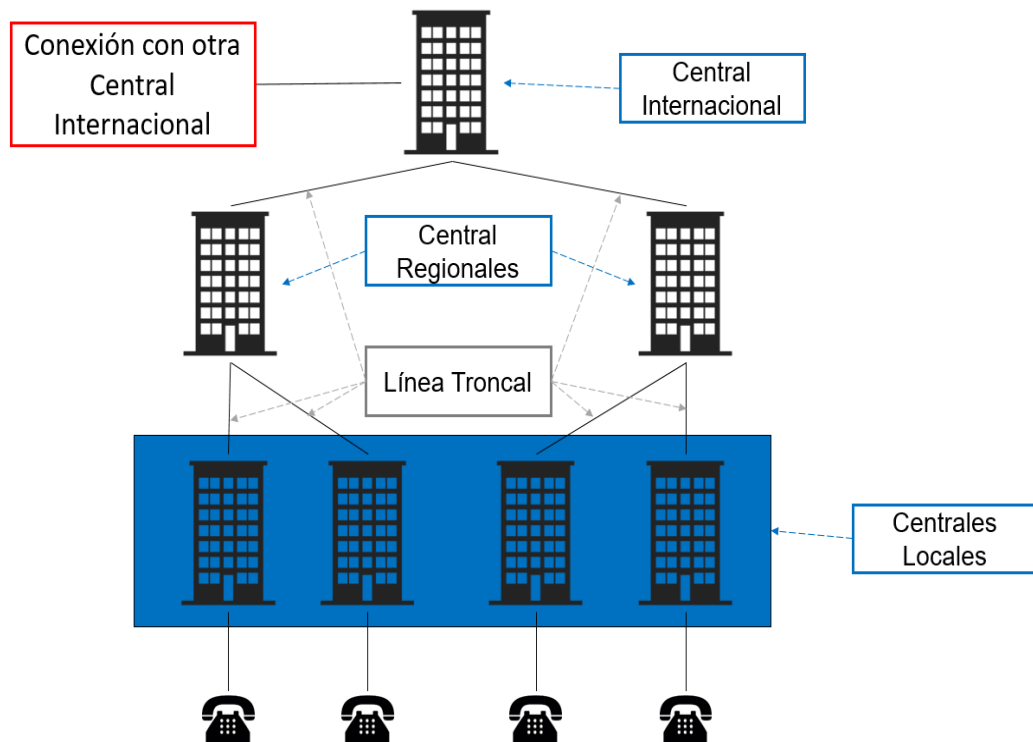


Figura 1.9 Central Regional e Internacional. [Elaboración propia]

1.2 Sistemas de Conmutación

La función principal de un sistema de conmutación de telecomunicaciones es proporcionar los medios para transmitir información de cualquier dispositivo terminal a cualquier otro dispositivo terminal seleccionado por la terminal origen [1.6], es decir, se proporcionan los medios para conectar cada terminal telefónica a cualquier otra en la misma central o a cualquier otra que esté conectada al troncal.

Existen 3 tipos de conmutación que difieren en las formas de conmutación en los nodos que existen desde el origen hasta el destino: En telecomunicaciones se utilizan la conmutación de circuitos y la conmutación de mensajes, por ejemplo, la técnica utilizada en la red de comunicación informática o la transferencia de datos es la conmutación de paquetes.

- **Conmutación de circuitos:** El sistema telefónico tradicional está basado en este tipo de conmutación (Figura 1.10). Cuando se realiza una llamada, se crea una sola trayectoria dedicada entre ambos abonados que seguirá estando activa hasta el momento en que termine la llamada, proporcionando un ancho de banda fijo y un retardo fijo muy corto.

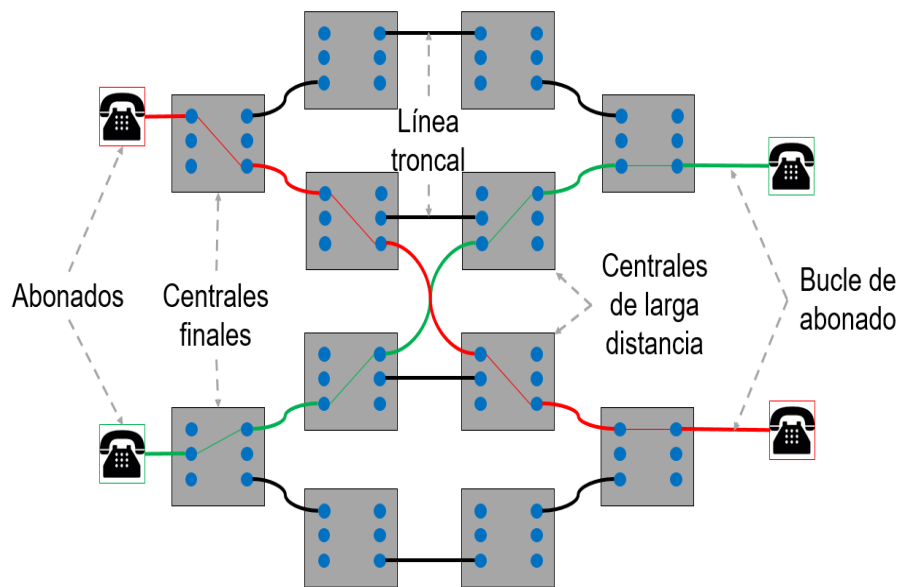


Figura 1.10 Conmutación de circuitos [Elaboración propia].

Un sistema de conmutación comercial brinda los siguientes servicios:

- La capacidad de comunicarse con cualquier otro usuario.
- El tiempo de conexión relativamente pequeño en comparación con el tiempo de espera o el tiempo de conversación.
- El grado de servicio de llamadas completadas puede promediar tan bajo como el 95%, aunque lo ideal generalmente para el sistema debe ser del 99%.
- El usuario asume la privacidad de la conversación.
- El sistema debe estar disponible en todo momento que desee utilizarlo.

Las funciones de conmutación son controladas por la terminal que realiza la llamada, ya sea un abonado local o de larga distancia. Las instrucciones se transmiten a la central vía remota con la acción de descolgado, colgado o marcado [1.4]. La comunicación establecida implica 3 facetas: establecer el circuito dedicado por medio de peticiones de enlace antes de enviar la señal de comunicación deseada, transferencia de la señal de comunicación deseada (datos) entre las terminales y finalmente la desconexión del circuito dedicado por orden de una de las terminales involucradas.

- **Conmutación de mensajes:** Muy utilizado en los años de 1960 y 1970, también conocida como conmutación de almacenamiento y reenvío, este tipo de conmutación almacena los mensajes y luego reenvía cada mensaje al siguiente nodo o destino sin que exista una relación entre el emisor y el receptor, solo incluye un encabezado que contiene la dirección del destino, información de enrutamiento e información de prioridad.
- **Conmutación de paquetes:** La tecnología de voz sobre IP está basado en este tipo de conmutación (Figura 1.11)⁷, en la cual los datos de origen se dividen en paquetes que contienen identificaciones de ruta o destino y se envían una vez que estén disponibles en los enrutadores sin necesidad de establecer una ruta dedicada para la transferencia de los paquetes, por lo que cada paquete puede ser enviado por diferentes rutas a través de la red y debido a ello pueden llegar en completo desorden sin que esto afecte en la comunicación.

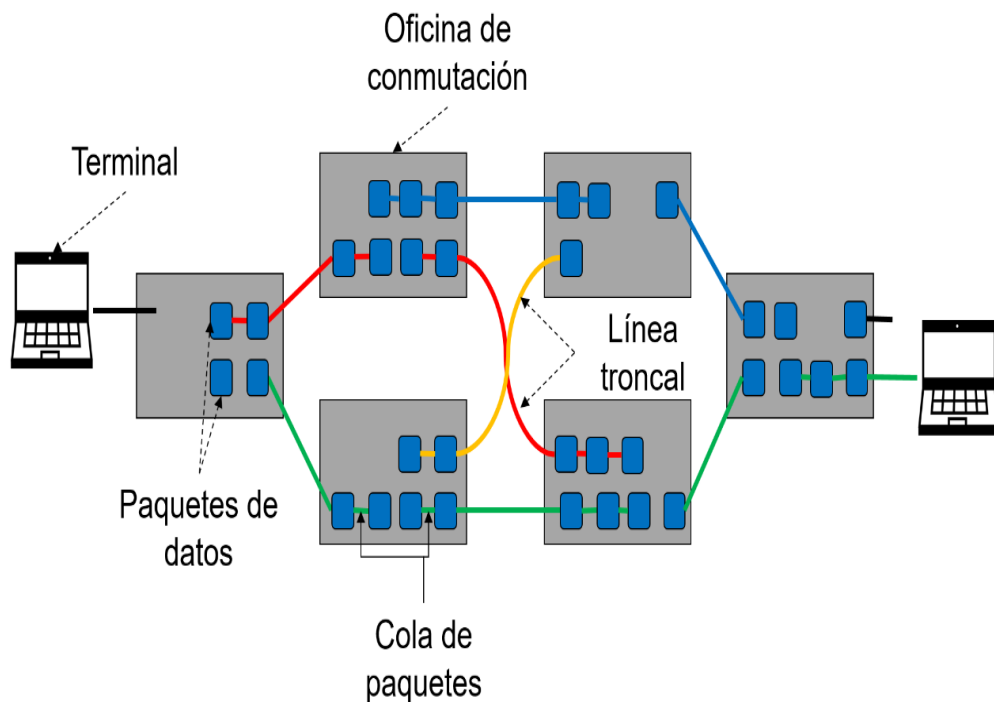


Figura 1.11 Conmutación de paquetes.

⁷ [1.10] "Conmutación de paquetes" Página 139.

Técnicas de conmutación:

1. **Datagramas** (Figura 1.12)⁸: En esta técnica los paquetes al ser enviados desde el origen se enrutan de forma independiente. Una tabla de enrutamiento almacenada en el conmutador especifica el destino, la tabla puede ser estática o puede actualizarse periódicamente. En el segundo caso, el enrutamiento depende de la estimación del enrutador del trayecto más corta al destino [1.6].

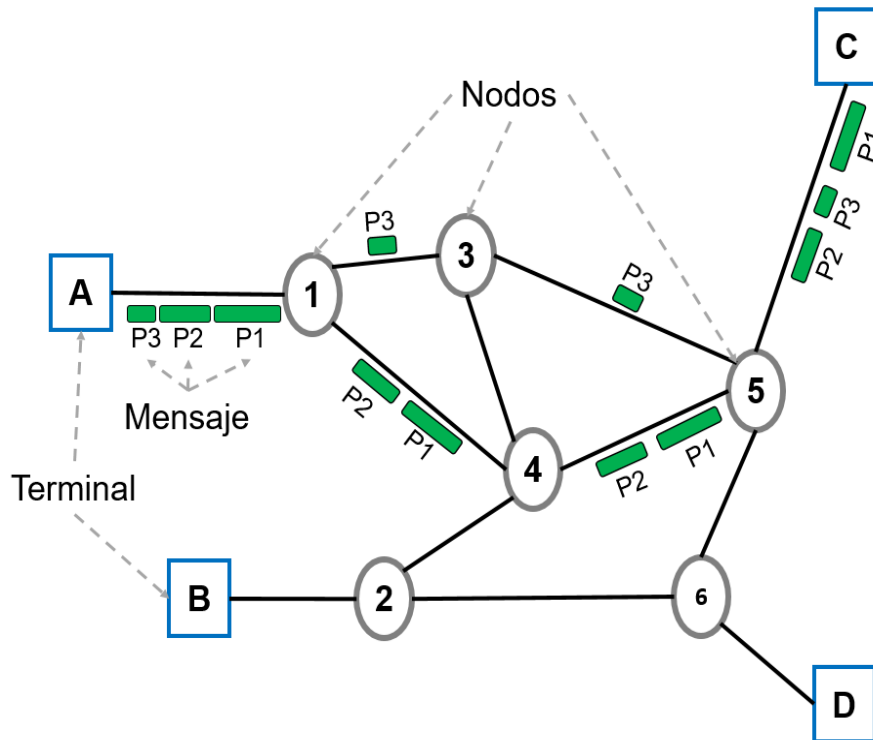


Figura 1.12 Transmisión por datagramas.

2. **Circuitos Virtuales** (Figura 1.13)⁹: La conexión virtual se establece al inicio de cada conversación de forma permanente y cada paquete perteneciente a una determinada conexión se transmite por la misma ruta establecida [1.1], muy parecido a la conmutación de circuitos físicos pero la principal diferencia entre ambos es que en los circuitos virtuales muchos usuarios comparten la capacidad de las líneas de transmisión y los canales entre los nodos de la red, así, en un momento determinado los usuarios activos pueden usar toda la capacidad disponible si los otros usuarios no están transmitiendo datos.

Para esta técnica de conmutación la información de ruta y dirección no es necesaria en los paquetes cuando se establece la conexión fija, solo contiene

⁸ [1.6] "Concepto de datagrama" Página 269.

⁹ [1.6] "Concepto de datagrama" Página 270.

el **VCI** (Identificador de Circuito Virtual, *Virtual Circuit Identifier*) de conexión en cada paquete para definirlo como parte de la conexión.

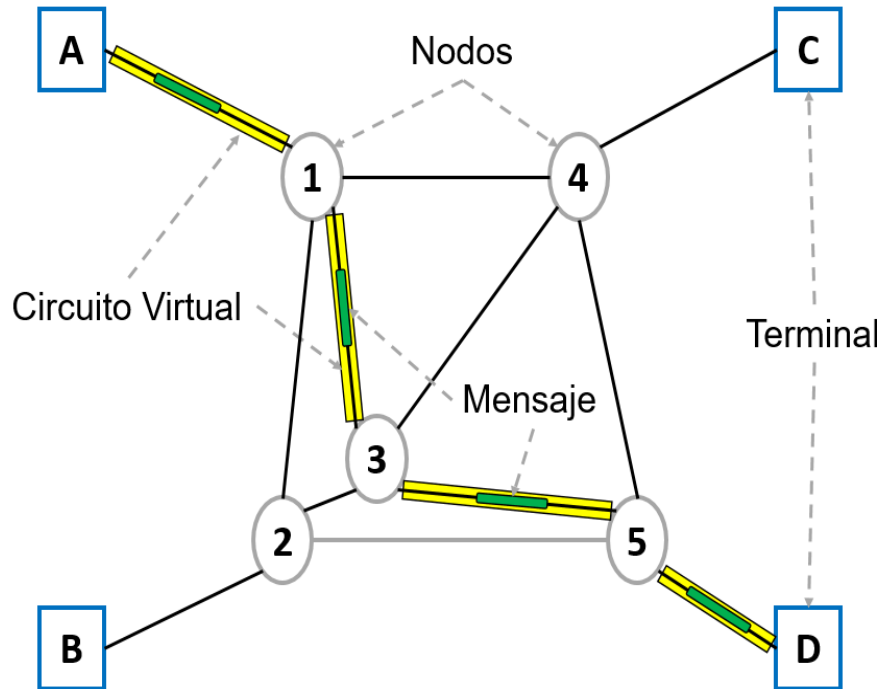


Figura 1.13 Circuito Virtual entre las terminales A y D.

La conmutación de circuitos y de paquetes difieren entre otros aspectos (Tabla 1.1)¹⁰, por ejemplo, la conmutación de paquetes no se reserva un ancho de banda para la transmisión de información por lo que tal vez los paquetes tengan un retraso para ser reenviados y provoque una congestión si se envían demasiados paquetes al mismo tiempo. Pero no existe peligro de no poder usar la red u obtener un tono de ocupado como en el caso de la conmutación de circuitos.

La congestión existe en diferentes situaciones para cada tipo de conmutación, para la de circuitos ocurre en el momento en que se establece la conexión o llamada y en la de paquetes al mismo tiempo en que se envían los paquetes.

¹⁰ [1.10] "Comparación de las redes de conmutación de circuitos y conmutación de paquetes" Página 142.

Acción	Conmutación de Circuitos	Conmutación de Paquetes
Establecimiento de llamadas	Requerido	No necesario
Trayectoria dedicada	Sí	No
Todos los paquetes siguen la misma trayectoria	Sí	No
Los paquetes recibidos llegan de forma ordenada	Sí	No
La falla de un conmutador	Fatal	Sin problemas
Ancho de banda disponible	Predeterminado	Dinámico
Posible congestión	Durante el establecimiento de la llamada	En el envío de los paquetes
Ancho de banda desperdiciado	Alto	Bajo
Transmisión de almacenamiento y reenvío	No	Sí

Tabla 1.1 Comparación de la Conmutación de Circuitos y la Conmutación de Paquetes.

1.2.1 El aparato telefónico

El teléfono es el dispositivo final en la red **PSTN** y funciona como un transductor (Figura 1.14)¹¹, el cual se encarga de transformar algún tipo de energía en otra; el teléfono se encarga de transformar la energía sonora en energía eléctrica por medio del elemento transmisor (micrófono) y viceversa por medio del receptor (auricular).

¹¹ [1.3] "Equivalencia entre la comunicación humana y el sistema telefónico" Pagina 10.

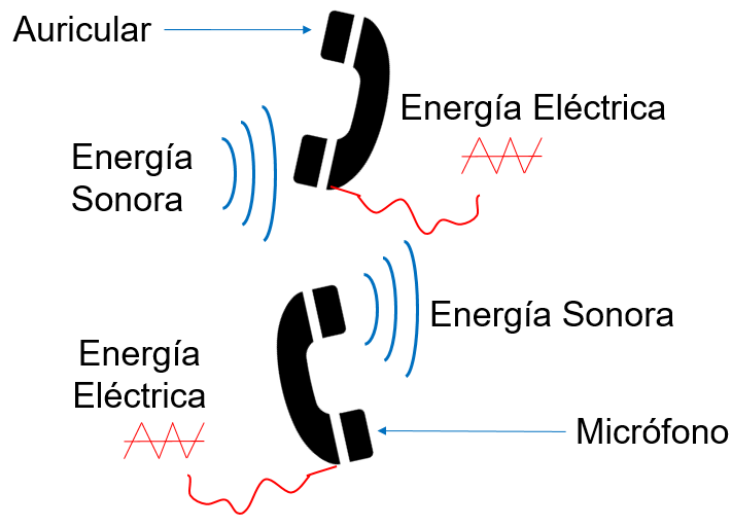


Figura 1.14 Sistema telefónico.

El aparato telefónico cuenta con 3 elementos fundamentales (Figura 1.15): un micrófono, una batería y un auricular. Además, llevan en su interior algunos componentes más que, sin ser indispensables, mejoran su funcionamiento como el circuito de timbre, el sistema de marcación y unos pocos componentes más que sirven para mejorar el fenómeno de autoescucha [1.3].

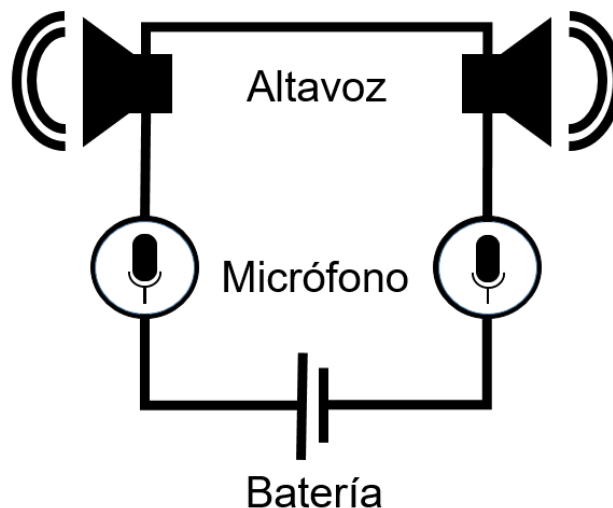


Figura 1.15 Circuito telefónico simple. [Elaboración propia]

1.2.1.1 Elemento transmisor: Micrófono

El micrófono se encarga de convertir la energía acústica (sonidos) en energía eléctrica (corriente), existen diferentes tipos como los de cristal, cinta, condensador, electrito y carbono, este último siendo el primero en ser y mayormente utilizado en los sistemas telefónicos.

Los micrófonos de carbono tienen un diafragma con pequeños recipientes que contienen granos de carbono tapados por una membrana (Figura 1.16)¹², que al momento de vibrar por la inserción del sonido produce deformaciones en los recipientes que al cambiar su forma física, funcionando como resistencia eléctrica variable para el voltaje de la batería, variando la corriente que cruza entre los granos de carbón; cuando las ondas sonoras presionan los granos de carbono con más fuerza la resistencia del bucle disminuye y la corriente aumenta ligeramente [1.1]. La corriente mínima requerida para el correcto funcionamiento de un micrófono de carbono moderno es de aproximadamente 23 mA.

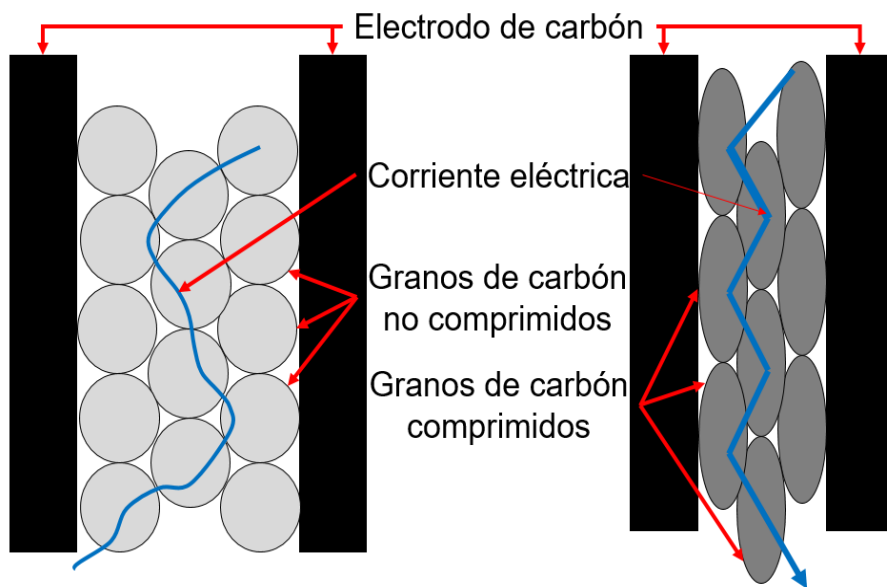


Figura 1.16 Comportamiento interno de un micrófono de carbono.

El funcionamiento básico del teléfono sigue siendo el mismo hoy en día, aunque los teléfonos modernos incluyen micrófonos más sofisticados y de mejor calidad como el electret, utilizado en los Smartphones.

1.2.1.2 Elemento receptor: Auricular

El objetivo del receptor es convertir la energía eléctrica entrante del otro dispositivo telefónico en energía sonora con la mayor fidelidad posible en el margen de frecuencias en las que se encuentra la voz humana, por ello es suficiente un ancho de banda de 300 Hz hasta 3400 Hz.

El auricular cuenta con un diafragma magnético parcialmente cónico con una bobina de 100 vueltas en ambos extremos del electroimán, de resistencia nominal de 400 Ω (Figura 1.17)¹³. La bobina es alimentada por corriente alterna

¹² [1.3] "Granalla sin comprimir" y "Granalla comprimida" Página 14.

¹³ [1.3] "Receptor" Página 15.

producida por el micrófono en el extremo remoto de la conexión, esta corriente genera un campo magnético variable que mueve el diafragma y produce ondas de sonido cercanas al sonido original en el extremo de transmisión [1.1].

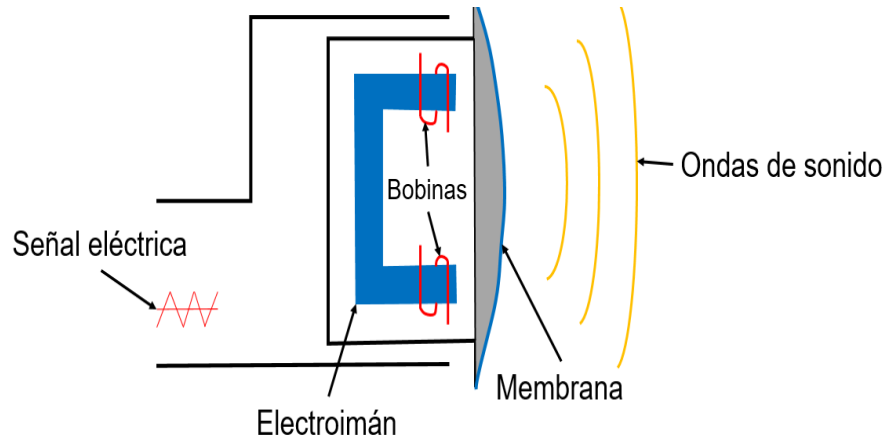


Figura 1.17 Composición del auricular.

1.2.1.3 Transformador de línea o bobina de inducción.

Se trata de un transformador con un par de embobinados (Figura 1.18) dentro de un aparato telefónico conectado en donde el embobinado primario está conectado en serie al micrófono del teléfono y el segundo al auricular seguido de la línea de conexión saliente.

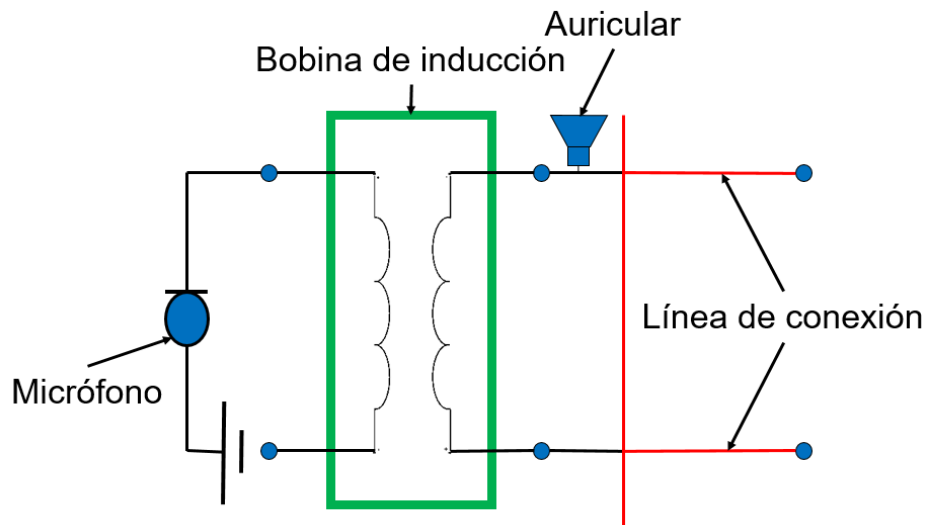


Figura 1.18 Ubicación de la bobina de inducción. [Elaboración propia]

Esencialmente el transformador de línea existe por 3 tareas fundamentales dentro del sistema telefónico.

- Acoplador de impedancias. La impedancia de los micrófonos es normalmente de 24Ω y la línea de 600Ω , por lo que la bobina de

inducción se encarga de adaptar las impedancias de cada componente y obtener el máximo rendimiento del sistema telefónico.

- Aumento del voltaje. Debido a las pérdidas existentes en los conductores dentro de las líneas telefónicas y así mejorar el rendimiento de la transmisión.
- Aislador de corrientes. Separa la corriente alterna de la corriente continua, es decir, separa la información (corriente alterna) útil de la de transporte (corriente continua), funcionando así como un demodulador.

1.2.1.4 Apertura y cierre del bucle local.

Funciona como interruptor en el teléfono, por lo que se encarga de permitir el paso de alimentación de la Central de Conmutación al dispositivo cuando está en uso (**OFF-HOOK**) y en caso de que no esté en uso (**ON-HOOK**) impedir el paso del flujo de corriente.

En general, una Central de Conmutación conoce el estado del teléfono si se encuentra en modo Descolgado si la corriente en el bucle local está por encima de 15 mA y colgado si la corriente está por debajo de un valor de 6 mA (Figura 1.19)¹⁴.

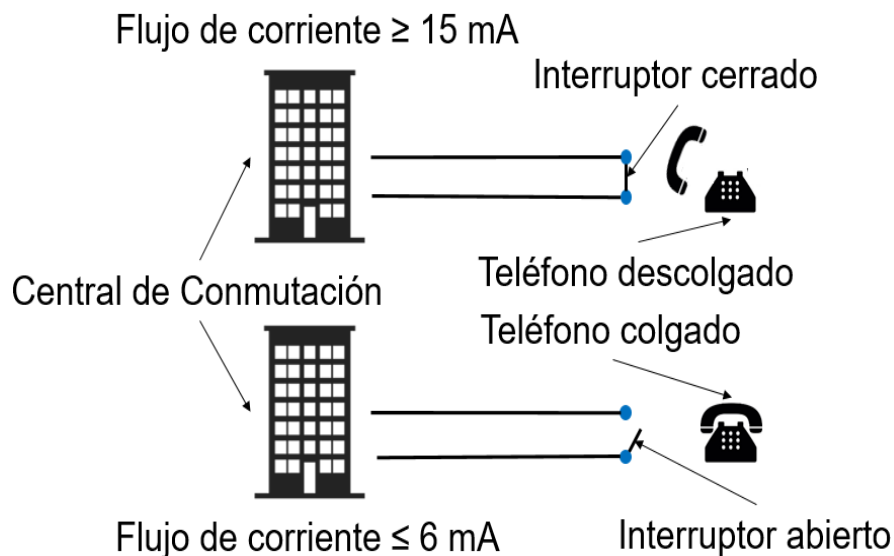


Figura 1.19 Conmutador de gravedad.

El tiempo en que la central de conmutación conoce el cambio de estado del dispositivo telefónico pasando de colgado a descolgado, es relativamente corto imperceptible para el ser humano no superando los 80 ms para el teléfono que realiza la llamada y para el que recibe la llamada tiene un tiempo no mayor a 40 ms. Al momento en finalizar la llamada la central de conmutación comprenderá el cambio de estado de los teléfonos con una apertura en el bucle local en cualquiera de las 2 terminales con un tiempo superior a los 150 ms.

¹⁴ [1.3] "Conmutador de gravedad" Página 19.

Actualmente se encuentra en desuso debido a las modificaciones electrónicas que han ayudado a la mejora del dispositivo telefónico.

1.2.1.5 Marcación telefónica

El sistema de marcación en un dispositivo telefónico es de suma importancia al igual que los puntos antes mencionados, sin él los usuarios no podrían entablar la comunicación telefónica por medio del marcado numérico, por lo que el modo de marcación puede ser por pulsos (actualmente en desuso) con el disco de marcado o por tonos con los teclados multifrecuencia.

- **Marcación por pulsos** (Figura 1.20): Consiste en abrir el bucle a intervalos regulares, tantas veces como indica el número a marcar. A cada apertura y cierre del bucle se le denomina pulso [1.3] conseguidos mediante el disco de marcación en los teléfonos antiguos.

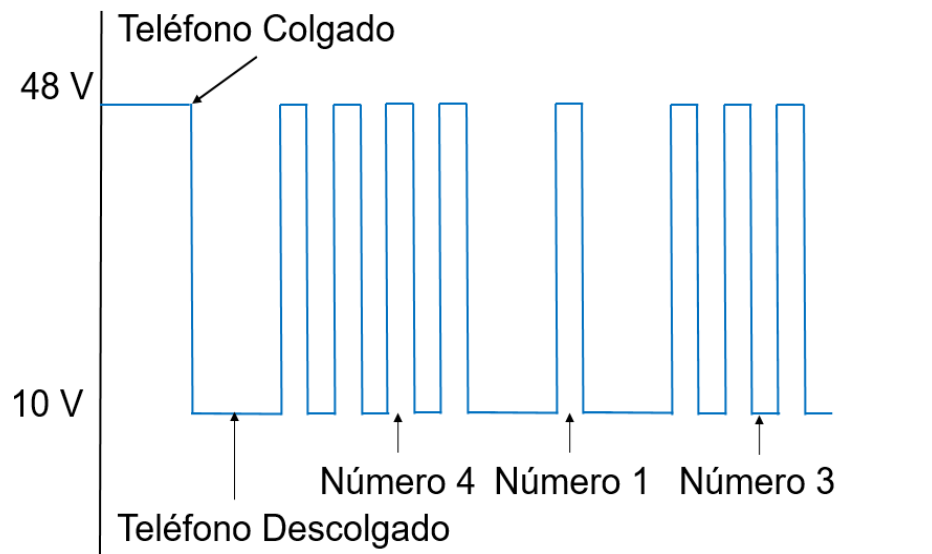


Figura 1.20 Pulsos de marcado. [Elaboración propia]

En el disco de marcado (Figura 1.21) se cuenta con un disco rotatorio en el que se encuentran los dígitos de 0 (10 pulsos) al 9 (9 pulsos), sirviendo para poder ingresar los dígitos del número telefónico con quien se desea comunicar. Rotando dígito por dígito del número destinatario hasta el tope del discado, en su retorno a la posición inicial genera los pulsos de marcado.

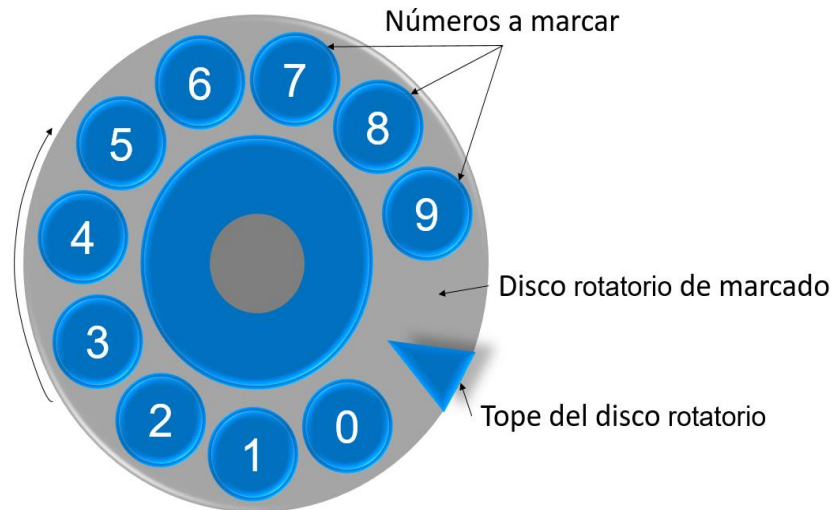


Figura 1.21 Disco de marcado. [Elaboración propia]

- Marcación por tonos:** En la marcación por tonos o también conocido como **DTMF** (*Dual Tone MultiFrequency*) hace uso de la suma de 2 señales senoidales con diferentes frecuencias (tabla 1.2), una alta y una baja por medio del teclado multifrecuencia, en donde cada frecuencia no es múltiplo de otra y la suma o resta de 2 frecuencias no resulta en una existente en el sistema de marcación por tonos.

Frecuencia	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Tabla 1.2 Teclado multifrecuencia. [Elaboración propia]

En tonos **DTMF** (Figura 1.22) producidos existen menos cantidad de señales armónicas que cualquier otra señal de audio y la duración mínima de un tono y la pausa entre dígitos es de 60 ms.

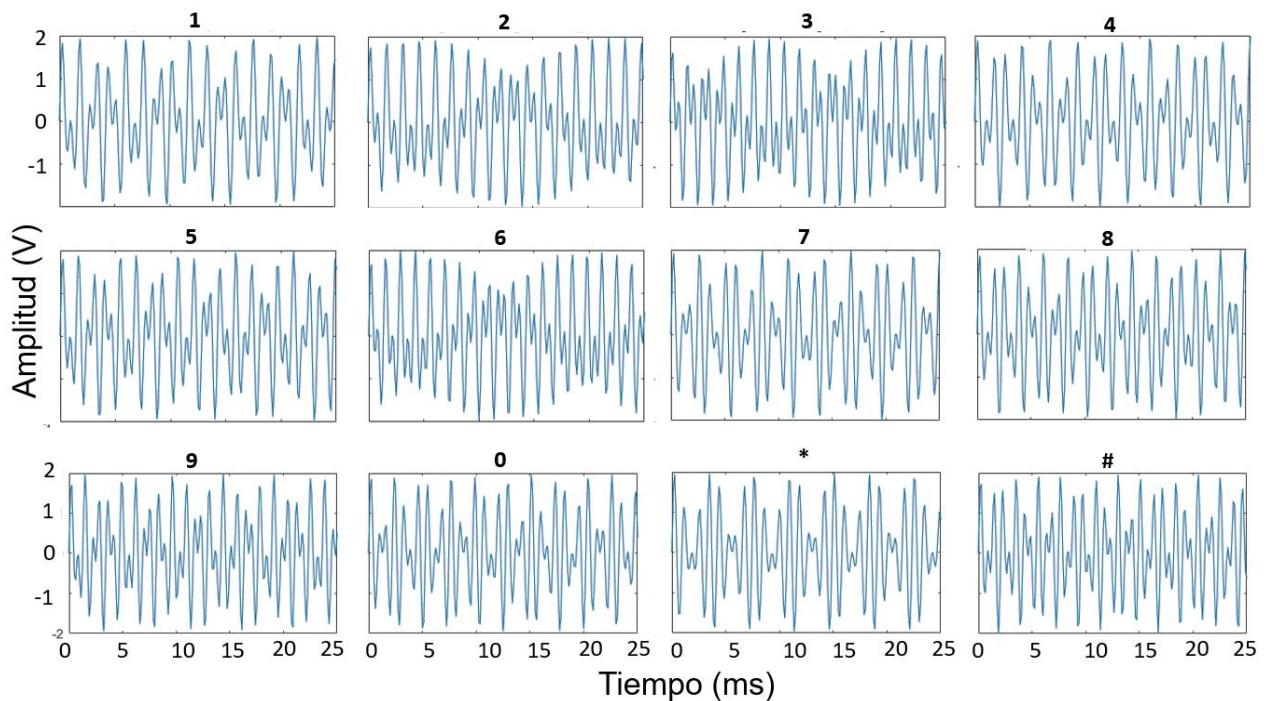


Figura 1.22 Tonos DTMF. [Elaboración propia]

Actualmente el sistema de marcado por tonos es utilizado en todos los teléfonos, sustituyendo a la marcación por pulsos. Las ventajas que presenta son:

- Conmutación más rápida
- Los dígitos tecleados pueden pasar al otro abonado (extensiones o buzones de voz).
- Más caracteres posibles de ingresar.

1.2.2 Señalización.

La señalización se encarga de generar e interpretar las señales de control y supervisión que se necesitan durante un enlace para transmitir voz o datos, además de intercambiar información entre los componentes de un sistema de telecomunicaciones para establecer, monitorear y liberar conexiones, controla las operaciones relacionadas con la red y el sistema [1.6]. En la red telefónica la señalización de una terminal brinda la información necesaria al conmutador encargado de enrutar la llamada, por lo tanto la red telefónica realiza el envío de la señalización en tono de marcado u ocupado y el tono de timbre a la terminal destino.

Existen dos tipos de señalización:

1. **Señalización de abonados** (usuario-red); intercambia la información de control entre el usuario final y la central telefónica (**PSTN** o una Red Telefónica Privada, **PBX**) por medio del sistema marcado **DTMF**.

2. **Señalización entre conmutadores** (red-red) (Figura 1.23); intercambia la información de control entre los switches de las redes telefónicas por medio de la **CAS** (Señalización Asociada al Canal) en donde la señalización de una llamada se transmite por el mismo canal en donde existe flujo de la señal voz o de la **CCS** (Señalización por Canal Común) en el cual existe una red de señalización independiente al canal del flujo de voz.

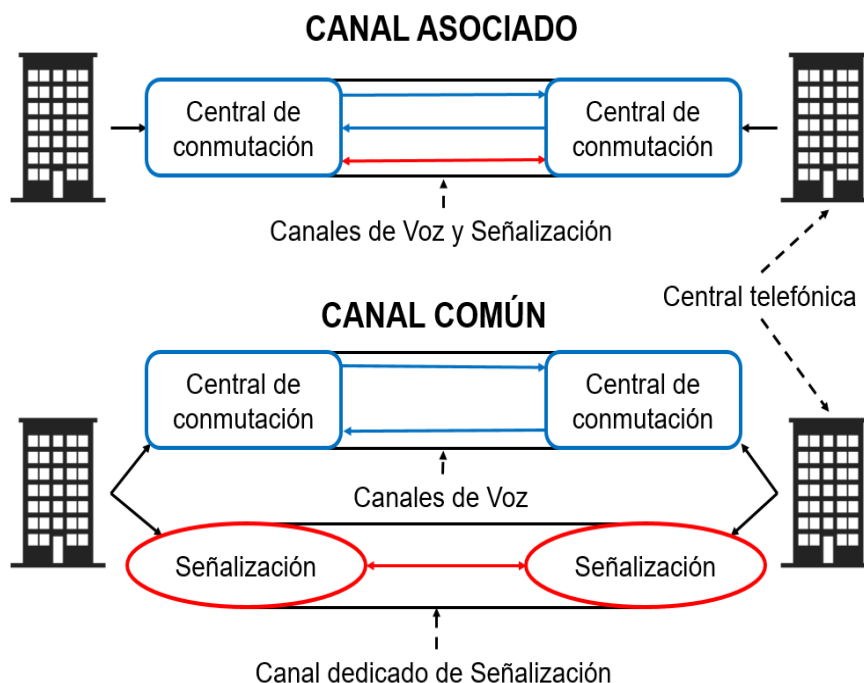


Figura 1.23 Señalización CAS y CCS. [Elaboración propia]

El envío de información entre abonados y entre centrales se puede dividir en 3 clases:

- **Señales de supervisión:** también denominada como Señalización de Bucle de Abonado; son necesarias para el establecimiento de la llamada e inspeccionar la misma, suelen ser del tipo binario (verdadero o falso): solicitud de llamada, respuesta de llamada y aviso de dispositivo ocupado. Informan el estado del abonado llamado y los recursos que tiene la red telefónica; si dicho recurso será utilizado y se encuentra disponible para una llamada, la señal de supervisión lo reserva solo para ese servicio.
- **Enrutamiento de señales o señales de registro:** la información de direccionamiento se encuentra en los dígitos ingresados en el momento en el que el abonado origen ingresa el número telefónico con el que desea comunicarse, en donde le indica a los centros de conmutación el enrutamiento requerido.
- **Gestión de señales o señalización entre registros:** este tipo de señalización implica la conmutación remota de circuitos, enrutamiento, modificación de

enrutamiento, sobrecarga de tráfico, prioridad de las llamadas y clase de servicio. Incluye también parte de la señalización de línea, en donde se obtiene el estado de los abonados mediante los tonos del servicio telefónico.

A diferencia de las señales de gestión de red, las 3 clases anteriores se encargan solamente del establecimiento y finalización del servicio, mientras que las de gestión de red se encargan del mantenimiento y solución de problemas en el momento de uso de la red telefónica.

1.3 Digitalización de la telefonía.

Se le denomina digitalización al proceso de conversión de una señal analógica en una señal digital usando el sistema de codificación mayormente conocido y usado como; **PCM** (Modulación por Pulsos Codificados, *Pulse Code Modulation*). Existen otros tipos para la digitalización de señales analógicas como: la **ADPCM** (Modulación por Pulsos Codificados Diferencial Adaptativa, *Adaptive Differential Pulse Code Modulation*), la **CVSD** (Modulación Delta de Pendiente Continua *Variable, Continuously Variable Slope Delta*), la **LPC** (Codificación Predictiva Lineal, *Linear Predictive Coding*) o la **DSI** (Interpolación Digital del Habla, *Digital Speech Interpolation*), todos con el fin de mejorar la calidad del sistema.

La digitalización de la voz (Figura 1.24)¹⁵ se realiza mediante el elemento **Codec** (Codificador - Decodificador) y el proceso se lleva a cabo mediante las siguientes etapas (**PCM**):

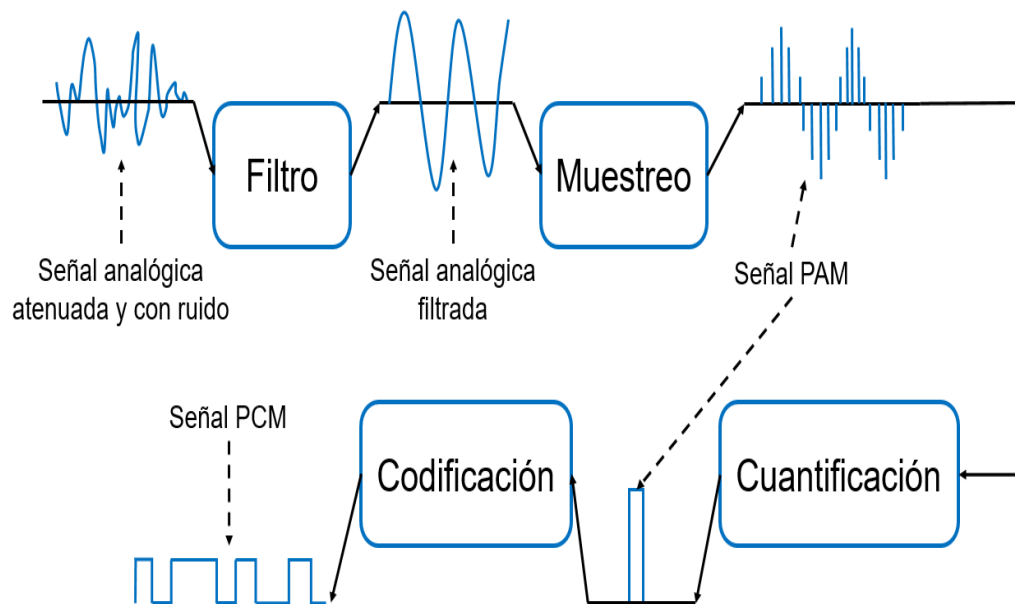


Figura 1.24 Proceso de la digitalización.

¹⁵ [1.6] "El proceso de digitalización" Página 90.

1.3.1 Filtrado

Debido a la existencia de ruidos eléctricos y perturbaciones de diafonía en la señal de voz que se transmite por los cables de abonado durante una llamada, es necesario la implementación primaria de un filtro que elimine frecuencias menores a 300 Hz y superiores a 3400 Hz por el rango de frecuencias que tiene la voz humana, por ello el ancho de banda real de la señal filtrada es de 3100 Hz.

La señal analógica también se atenúa debido a la distancia que existe entre los dispositivos por donde fluye la señal de voz durante una llamada telefónica.

1.3.2 Muestreo

PCM se basa en el Teorema de Muestreo o bien el Teorema Nyquist-Shannon, debido a que la señal analógica muestreada debe llevar la suficiente información para poder ser comprendida por el receptor, es necesario que la señal $f(t)$ (señal analógica), se muestree a intervalos regulares de tiempo (T_m) con una frecuencia mayor al doble (f_m) de la frecuencia más alta de la señal, las muestras así obtenidas contienen toda la información de la señal original [1.9].

Las señales de voz están en un rango de frecuencias entre 300 Hz a 3400 Hz, por lo que la frecuencia de muestreo es de 8000 muestras por segundo (Ecuación 1.1).

$$f_m \geq 2 (f_a) \quad \text{(Ecuación 1.1)}$$

Sustituyendo la frecuencia más alta (f_a) de la señal de voz (3400 Hz) para obtener la frecuencia de muestreo f_m :

$$f_m \geq 2 (3400 \text{ Hz})$$

$$f_m \approx 8000 \text{ Hz}$$

Intervalos de tiempo de muestreo (T_m) (Ecuación 1.2):

$$T_m = \frac{1}{8000 \text{ Hz}} = 125 \mu\text{s} \quad \text{(Ecuación 1.2)}$$

El proceso de muestreo cuenta con un modulador de amplitud de un tren de pulsos de amplitud constante, por lo que se le asigna normalmente el nombre de **PAM** (Modulación de Amplitud de Pulsos, *Pulse Amplitude Modulation*) (Figura 1.25)¹⁶.

¹⁶ [1.6] "Generación de muestras PAM" Página 91.

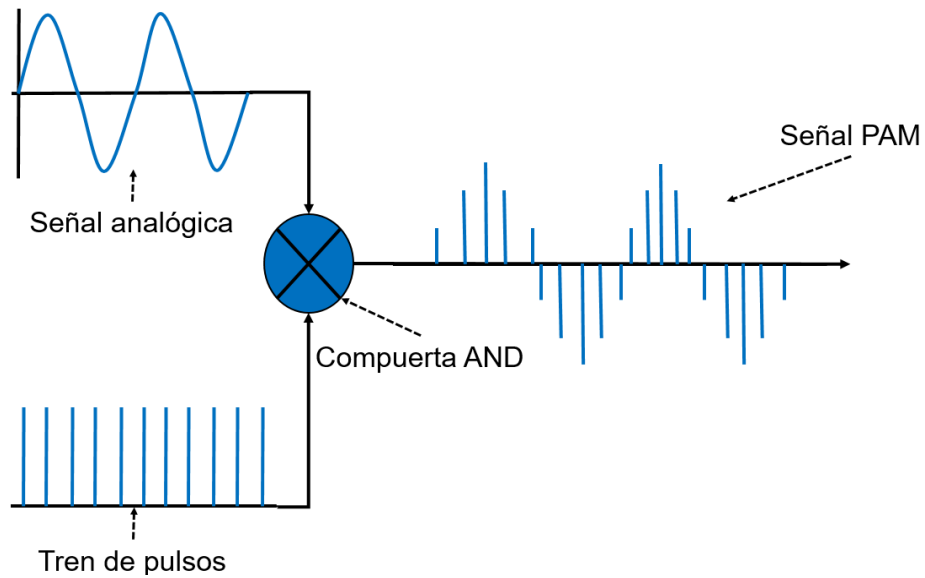


Figura 1.25 Señales PAM.

Existe una variación en la anchura de las muestras de una señal analógica y a esto se le conoce como muestreo real (Figura 1.26), es decir, en el muestreo ideal la toma de muestras consta de espigas bien definidas mientras que en el muestreo real las espigas presentan ensanchamiento.

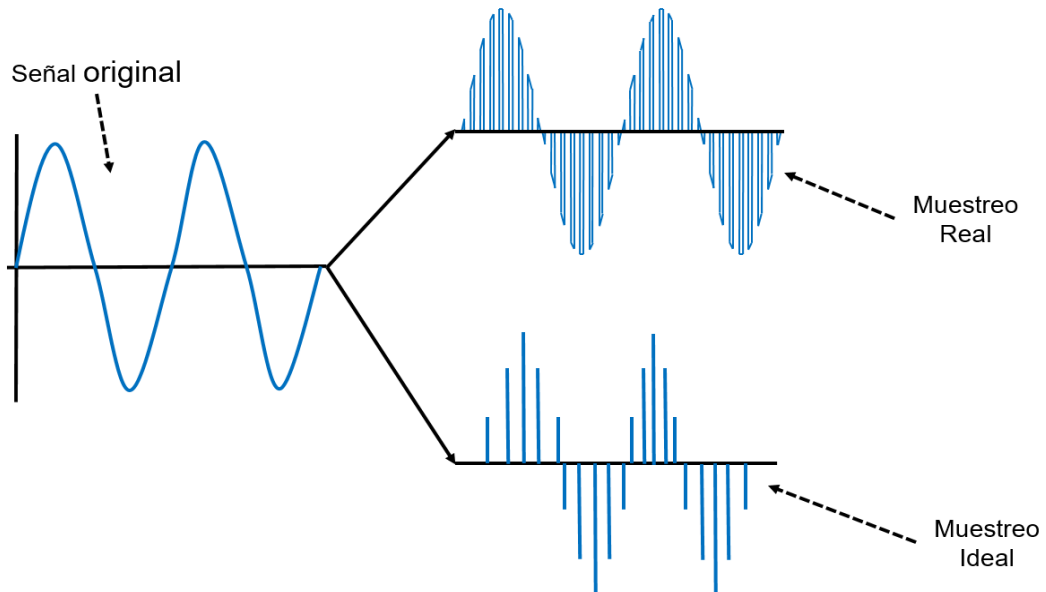


Figura 1.26 Muestreo Real y Muestreo Ideal. [Elaboración propia]

1.3.3 Cuantificación

La función principal de la cuantificación es la asignación de valores a las muestras de la señal analógica dentro de un rango de posibles valores conocidos como intervalos de cuantificación (Figura 1.27)¹⁷; los cuales pueden estar por encima o por debajo del

¹⁷ [1.3] "Cuantificación" Página 100.

valor real debido a que aún no pueden ser enviadas debido a que las amplitudes están limitadas y pueden tomar valores infinitos.

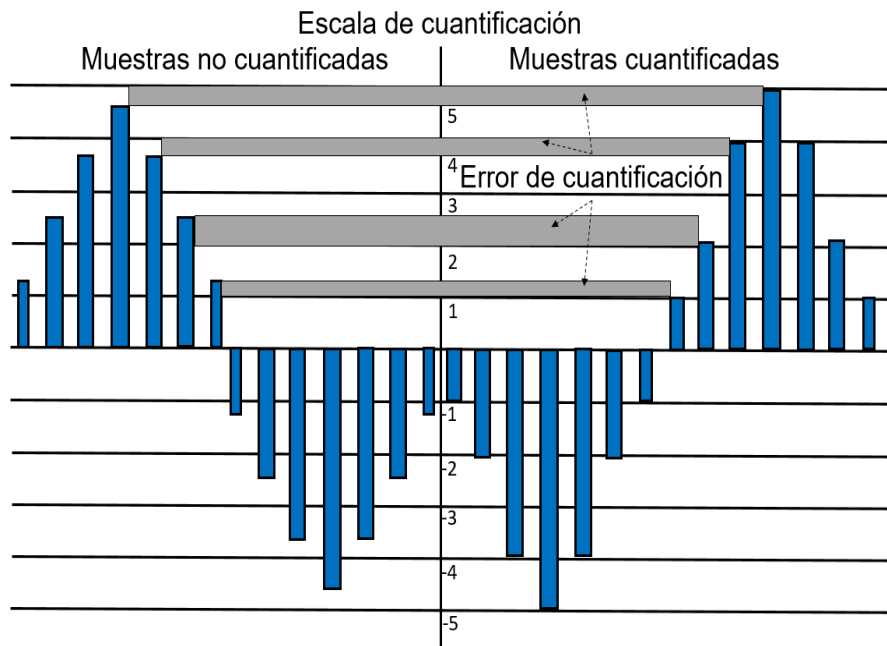


Figura 1.27 Señal cuantificada.

El mismo hecho de redondear o aproximar los valores de las muestras a intervalos de valores conocidos, producen pérdidas inevitables de información al igual que en la etapa de filtrado, pues el valor real de las muestras son sustituidas por aproximaciones. Dicho error se le conoce como error de cuantificación o ruido de cuantificación (Figura 1.28)¹⁸.

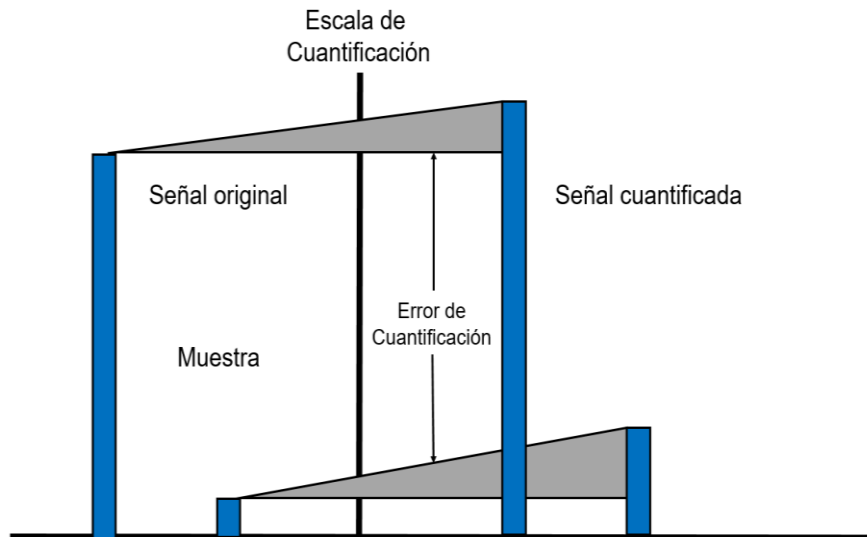


Figura 1.28 Error de cuantificación.

¹⁸ [1.3] "Error en la cuantificación no uniforme" Página 103.

Para disminuir el error de cuantificación existen dos formas de hacerlo:

- Aumentar el número de muestras, así, los intervalos serían demasiado pequeños y la medición podría ser exacta pero no es viable por el aumento que provocaría en el ancho de banda.
- Implementar un sistema de medición no lineal dependiente de la amplitud de la señal: Cuantificación No Uniforme, en donde se toma un número determinado de intervalos y se disminuye de forma no uniforme aproximándolos en los niveles bajos de señal y separándolos en los niveles altos [1.2], es decir, para los niveles bajos o débiles de la señal se utiliza un número muy alto de intervalos de cuantificación y para niveles altos o fuertes los intervalos disminuyen, esto debido a que el oído humano es capaz de percibir mayormente los cambios de amplitud en volúmenes bajos que en altos.

1.3.4 Compresión y codificación.

La última etapa para digitalizar una señal analógica es la parte de codificación y compresión, las cuales se realizan al mismo tiempo; la codificación asigna valores establecidos binarios a los datos previamente cuantificados. Las funciones de compresión y expansión son logarítmicas.

Aunque existen diferentes sistemas de medición, las leyes utilizadas en señales de frecuencias vocales son: la Ley A (**a-law**) en Europa y el resto del mundo y la Ley μ (**μ -law**) en Estados Unidos y Japón, siendo ambos sistemas de Cuantificación No Uniformes de tipo logarítmico.

Las fórmulas logarítmicas para la Ley A son las siguientes (Ecuación 1.3 y Ecuación 1.4):

$$F_A(x) = \left(\frac{a|x|}{1+\ln(A)} \right), \quad 0 \leq |x| \leq \frac{1}{A} \quad (\text{Ecuación 1.3})$$

$$F_A(x) = \left(\frac{1+\ln(A|x|)}{1+\ln(A)} \right), \quad \frac{1}{A} \leq |x| \leq 1 \quad (\text{Ecuación 1.4})$$

Donde $F_A(x)$ es la amplitud de la señal a la salida, x es la amplitud de entrada normalizada y A es el parámetro de compresión (Figura 1.29) positivo mayor a 1 que típicamente puede tomar el valor de 87.6 y se define por la relación señal a ruido (S/D=37.5 dB).

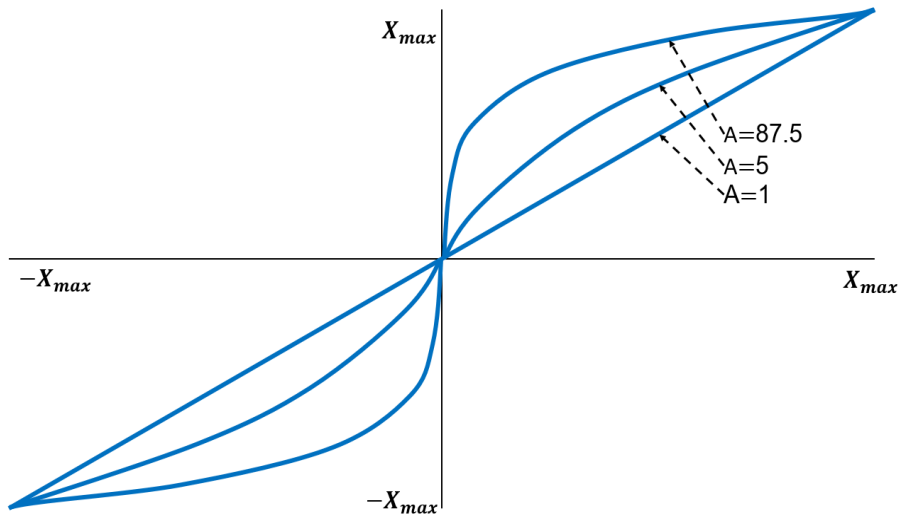


Figura 1.29 Curva de Ley A. [Elaboración propia]

De la misma manera para la Ley μ (Ecuación 1.3):

$$F_{\mu}(x) = \frac{\ln(1+\mu|x|)}{\ln(1+A)} \quad (\text{Ecuación 1.3})$$

Donde $F_{\mu}(x)$ es el valor de la señal de salida, x (Figura 1.30) es el valor de la señal de entrada y μ es el parámetro de compresión positivo mayor a 0 que típicamente puede tomar el valor de 255 y se define por la relación señal a ruido ($S/D=37$ dB).

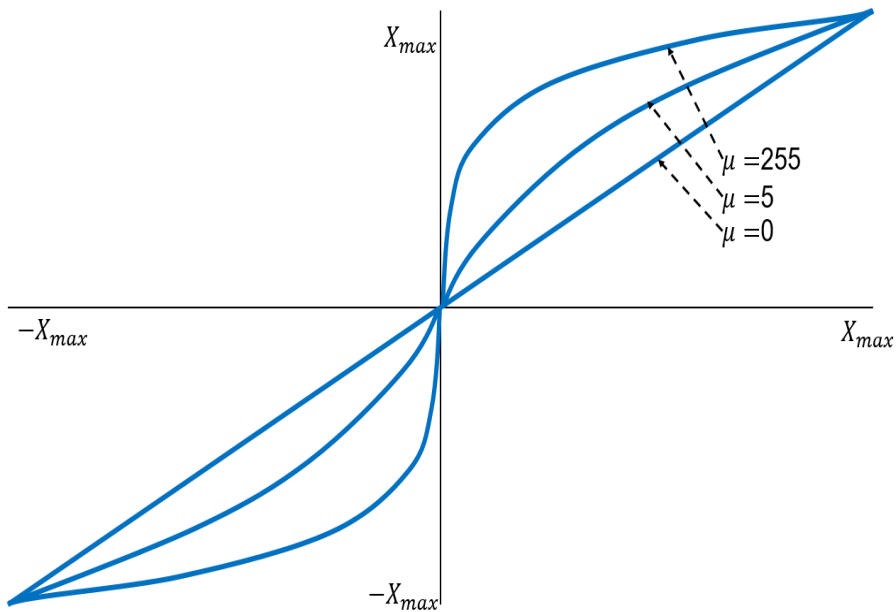


Figura 1.30 Curva de Ley μ . [Elaboración propia]

En cuanto a la codificación se necesitan secuencias binarias de 8 bits (Figura 1.31) para representar las posibles muestras cuantificadas previamente en el proceso de digitalización de la siguiente manera:

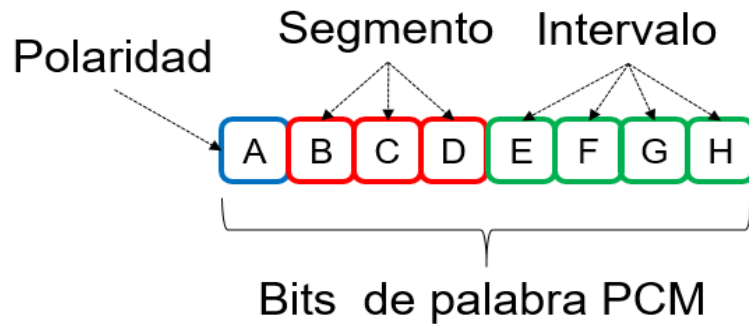


Figura 1.31 Bits de palabra PCM. [Elaboración propia]

El bit A corresponde a la polaridad de la muestra, siendo 1 si es positivo y 0 si es negativo.

Del bit B al D sirven para localizar los 8 segmentos ($2^3 = 8$) por encima o por debajo del origen, por lo cual consta de 16 segmentos en total de los cuales los 4 segmentos centrales serán tomados como solo 1 segmento y por lo tanto quedan 13 segmentos (Tabla 1.3):

Número Segmento	Código	Número Segmento	Código
12 (15)	1111	6 (7)	0111
11 (14)	1110	6 (6)	0110
10 (13)	1101	5	0101
9 (12)	1100	4	0100
8 (11)	1011	3	0011
7 (10)	1010	2	0010
6 (9)	1001	1	0001
6 (8)	1000	0	0000

Tabla 1.3 Codificación de segmentos. [Elaboración propia]

Por último los bits del E al H, sirven para determinar 16 intervalos en cada uno de los segmentos de recta y se codifican de la siguiente manera (Tabla 1.4):

Número Intervalo	Código	Número Intervalo	Código
15	1111	7	0111
14	1110	6	0110
13	1101	5	0101
12	1100	4	0100
11	1011	3	0011
10	1010	2	0010
9	1001	1	0001
8	1000	0	0000

Tabla 1.4 Codificación de intervalos. [Elaboración propia]

Suponiendo que el receptor recibe las siguientes secuencias: 10110100, 10110111 y 10111110 entonces (Figura 1.32)¹⁹:

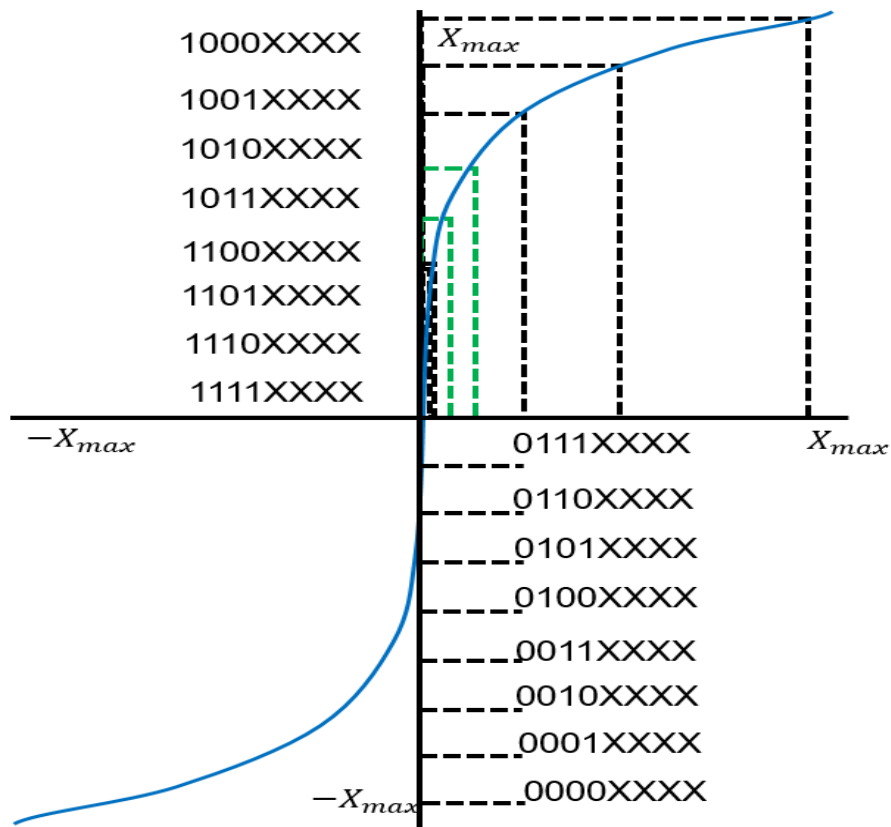


Figura 1.32 Ubicación de segmento.

¹⁹ [1.4] "Aproximación de 13 segmentos de la curva de la ley A utilizada con un equipo E1 PCM." Página 170.

Todas las secuencias de bits están dentro de la sección verde de la Figura 1.32 y haciendo un acercamiento a esta zona se vería de la siguiente manera (Figura 1.33):

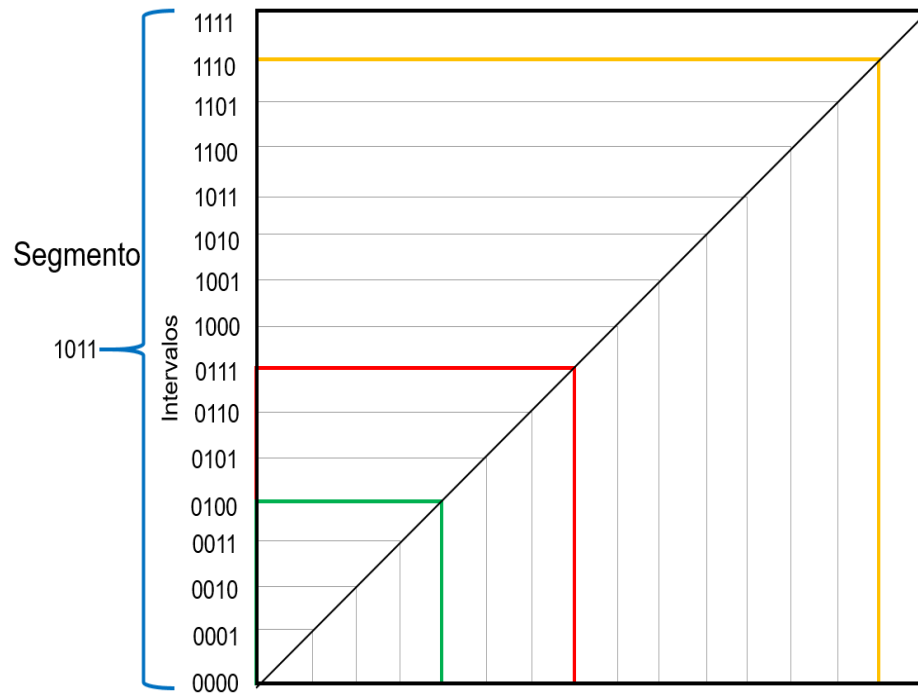


Figura 1.33 Ubicación de intervalos. [Elaboración propia]

- Secuencia 10110100 (intervalo verde).
- Secuencia 10110111 (intervalo rojo).
- Secuencia 10111110 (intervalo naranja).

1.4 Convivencia de la telefonía y las redes de datos.

La red telefónica fue inicializada con la única idea de transmitir voz a través del sistema basado en la conmutación de circuitos de forma analógica. Sin embargo, con el paso del tiempo el ser humano no solo necesitaba el servicio de voz sino también de datos y video, por ende fue necesario la creación de nuevas tecnologías digitales para poder brindar otros servicios con los mismos medios de transmisión que se ocupan para la telefonía, debido a que la arquitectura del sistema telefónico ya se encontraba establecido.

Con el fin de no solo brindar el servicio de telefonía a través de su propia red, se creó la Red Digital de Servicios Integrados²⁰ (Figura 1.34).

²⁰ <https://www.itu.int/rec/T-REC-I/es>: Recomendaciones ITU de la RDSI.

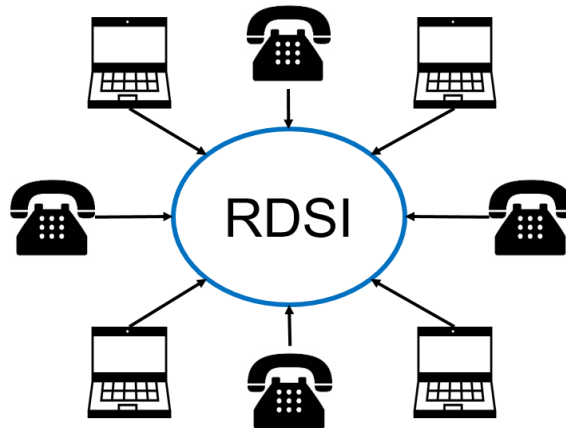


Figura 1.34 Red Digital de Servicios Integrados. [Elaboración propia]

1.4.1 RDSI o ISDN

La RDSI es un conjunto de estándares de transmisión digital que se utilizan para la conectividad digital de extremo a extremo. Servicios integrados se refiere a su capacidad para soportar numerosas aplicaciones [1.6], esta tecnología fue el siguiente paso de una red analógica como la red telefónica, aproximadamente en el año 1970 y se utilizó para los servicios de transmisión de voz y datos, con una capacidad por canal de transmisión de 64 kbps.

Históricamente los servicios integrados fueron idealizados durante la Segunda Guerra Mundial, pues en ese entonces la telefonía y el telégrafo competían a la par, evolucionando de forma similar pero cada uno por separado, el próximo servicio a integrarse a la idea más no a la tecnología fue el servicio de fax; para entonces la red telefónica se encontraba de manera irremplazable, por lo que se empezó con el estudio de poder agregar otros servicios de forma adjunta al de telefonía con los límites establecidos en la red telefónica analógica a través de la **PSTN**.

1.4.1.1 Arquitectura RDSI.

La arquitectura RDSI se encarga de proveer funciones de los dispositivos necesarios para el acceso de los usuarios a la red digital, su composición física se dividen en 2 conceptos fundamentales (Figura 1.35):

- **Grupos Funcionales:** conjunto de funciones necesarios para el acceso de los usuarios a los servicios **RDSI**, los grupos funcionales incluye:
 - **Terminales de red 1 (TR1):** La **TR1** tiene las mismas funciones que una terminal de línea funcionando como multiplexor y demultiplexor.
 - **Terminales de red 2 (TR2):** Las **TR2** o **NT2**, se encargan del multiplexaje y la conmutación funcionando como acceso directo a los dispositivos del usuario como teléfonos o computadoras realizando tareas a nivel de capa física y enlace de datos, el

ejemplo más conocido es la **PBX** (Red Telefónica Privada o Central Privada, *Private Branch eXchange*). La **NT2** coordina las comunicaciones de varias líneas telefónicas y finalmente las multiplexea para que sean enviadas por una terminal **NT1**.

- **Equipo Terminal 1 (TE1)**: Ambos (**TE1** y **TE2**) se refieren al dispositivo de abonado que hace uso de la **RDSI**. El **TE1** son terminales diseñados para conectarse directamente con la **RDSI** cumpliendo con la interfaz estándar de dicha red.
 - **Equipo Terminal 2 (TE2)**: Son dispositivos que no son compatibles con la **RDSI**.
 - **Adaptadores de Terminal (TA)**: Brinda compatibilidad a los dispositivos que no son compatibles con la **RDSI** convirtiendo señales analógicas en digitales y viceversa.
- **Puntos de Referencia**: Son una serie de etiquetas que definen las interfaces entre los grupos funcionales:
 - **Punto de referencia R**: Define la conexión entre el equipo **TE2** y el **AT**, siendo una solución provisional y no un estándar, dependiendo del fabricante.
 - **Punto de referencia S**: Define la conexión entre el equipo **TE1** y el **TR2**.
 - **Punto de referencia T**: Define la conexión entre **TR2** y **TR1**.
 - **Punto de referencia U**: Define la conexión entre **TR1** y la **RDSI**.
 - **Punto de referencia V**: Puede ser una interfaz física real, cuando los equipos de transmisión y conmutación están de manera independiente o virtual cuando en un equipo trabajan en conjunto las funciones de transmisión y conmutación.

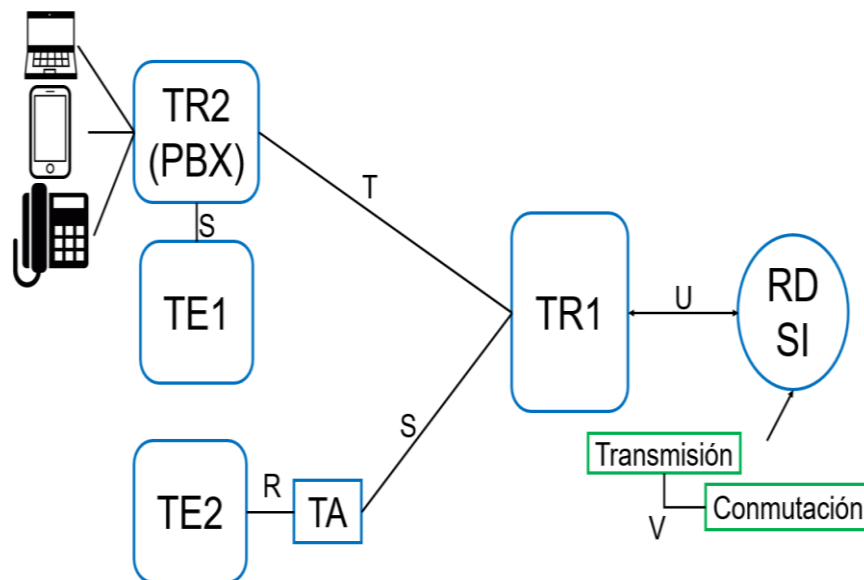


Figura 1.35 Grupos funcionales y Puntos de Referencia. [Elaboración propia]

1.4.1.2 Canales RDSI.

La RDSI consta de 3 tipos de canales:

1.4.1.2.1 Canal Portador (Canal B, Bearer Channel): Transportan datos a 64 kbps de forma dúplex, lo que significa que el ancho de banda está destinado únicamente para la transmisión de información (voz, video y datos) soliendo ser conexiones conmutadas por circuitos de forma física de extremo a extremo.

1.4.1.2.2 Canal Delta (Canal D): Puede trasportar datos de 16 a 64 kbps dependiendo del tipo de comunicación que necesite el usuario, este canal está destinado al envío de la señalización de control y de administración, usando la conmutación de paquetes durante la conexión en donde los canales B se encargan de enviar y recibir la información (voz, video y datos).

1.4.1.2.3 Canal Híbrido (Canal H): es una combinación de canales B para obtener velocidades mayores al momento de la trasferencia de información, adecuados para la transmisión de alta velocidad para video, videollamadas o teleconferencias.

- Canal H0: 6 canales tipo B de 64 kbps.
- Canal H10: 23 canales tipo B de 64 kbps.
- Canal H11: 24 canales tipo B de 64 kbps.
- Canal H12: 34 canales tipo B de 64 kbps.

1.4.1.3 Interfaces ISDN

Dentro de la RDSI se encuentran 2 tipos de interfaces:

1.4.1.3.1 BRI (Interfaz de Tarifa Básica)

La **BRI** (*Basic Rate Interface*) básicamente está compuesto por 2 canales tipo B que sirven para la trasferencia de información, ya sea de tipo de voz o datos con un ancho de banda de 64 kbps conectado a una Terminal de Red 1 (**TR1**, **NT1** o **NTU**, *Network Termination Unit*) y un canal tipo D para la transmisión de la información de señalización con un ancho de banda de 16 kbps (Figura 1.36 y Figura 1.37)²¹. Pensado especialmente para conexiones informáticas de uso individual, de forma que se pueda enviar la mayor cantidad de datos posibles al hogar a través de la red telefónica.

²¹ [1.6] "ISDN BRI" Página 309. "Concepto de BRI" Página 310.

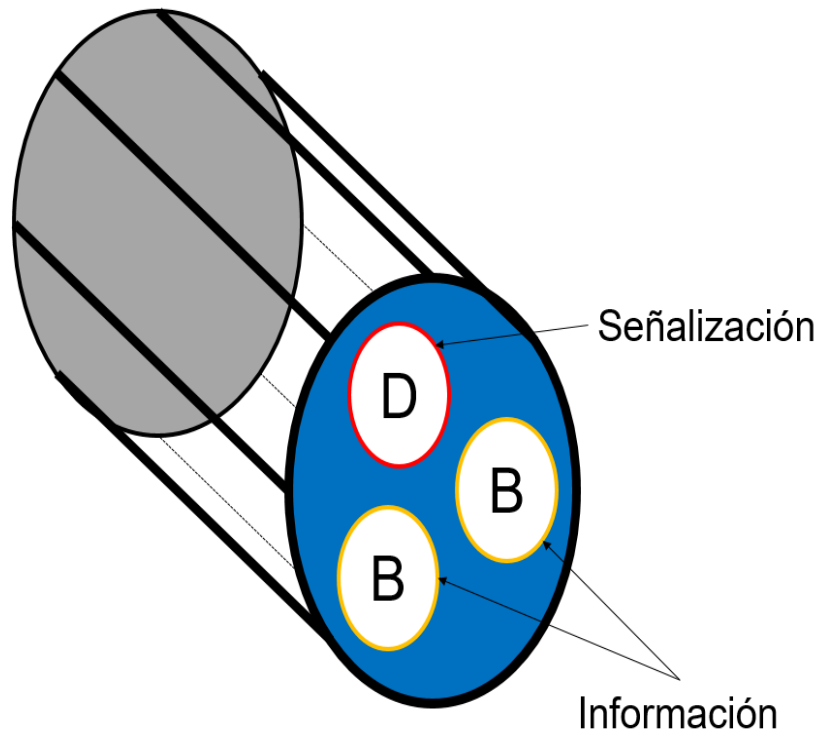


Figura 1.36 Interfaz BRI (2B+D).

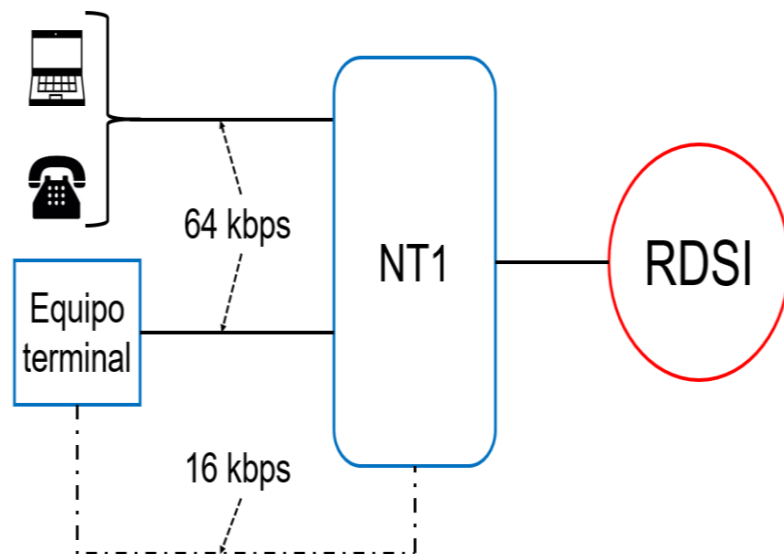


Figura 1.37 Conexión BRI (2B+D).

1.4.1.3.2 PRI (Interfaz de Tasa Primaria)

PRI (*Primary Rate Interface*) se compone por n canales tipo B y 1 tipo D de 64 kbps (Figura 1.38)²², la cantidad de canales B está limitado por el tamaño de la línea troncal utilizada en la región, existen 2 tipos de velocidades:

²² [1.6] "ISDN PRI" Página 310.

- 1.544 Mbts/seg = 23B + 1D siendo la configuración **T1** norteamericana.
- 2.048 Mbts/seg = 30B + 1D siendo la configuración europea **E1** y el resto del mundo.

Está diseñado para dividir las velocidades y entablar cualquier servicio a múltiples usuarios por lo que se puede utilizar de forma combinada los canales B (canal H).

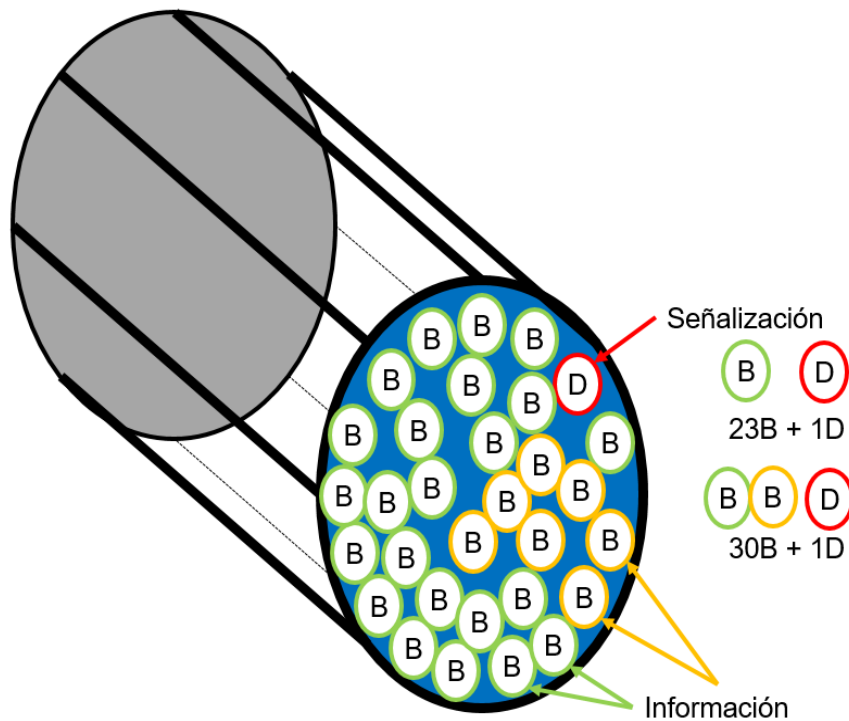


Figura 1.38 Interfaz PRI (T1 Y E1).

1.4.1.4 Tipos de RDSI.

Los sistemas **RDSI** se pueden segmentar según su velocidad de transmisión, y existen 2 tipos de ellos:

1. **RDSI** de banda estrecha (**N-ISDN**, *Narrow Band - ISDN*)

La Red Digital de Servicios Integrados de Banda Angosta es la RDSI original, ésta brinda servicios digitales sobre el par de cobre o la red telefónica existente con velocidades de transmisión de 128 kbps hasta 2 Mbps usando la conmutación de circuitos.

2. **RDSI** de banda ancha (**B-ISDN**, *Broad Band - ISDN*)

B-ISDN brinda servicios digitales estableciendo circuitos virtuales para la transmisión de paquetes con una velocidad de 150 Mbps, usando fibra óptica en todos los niveles de telecomunicaciones en la conmutación de paquetes.

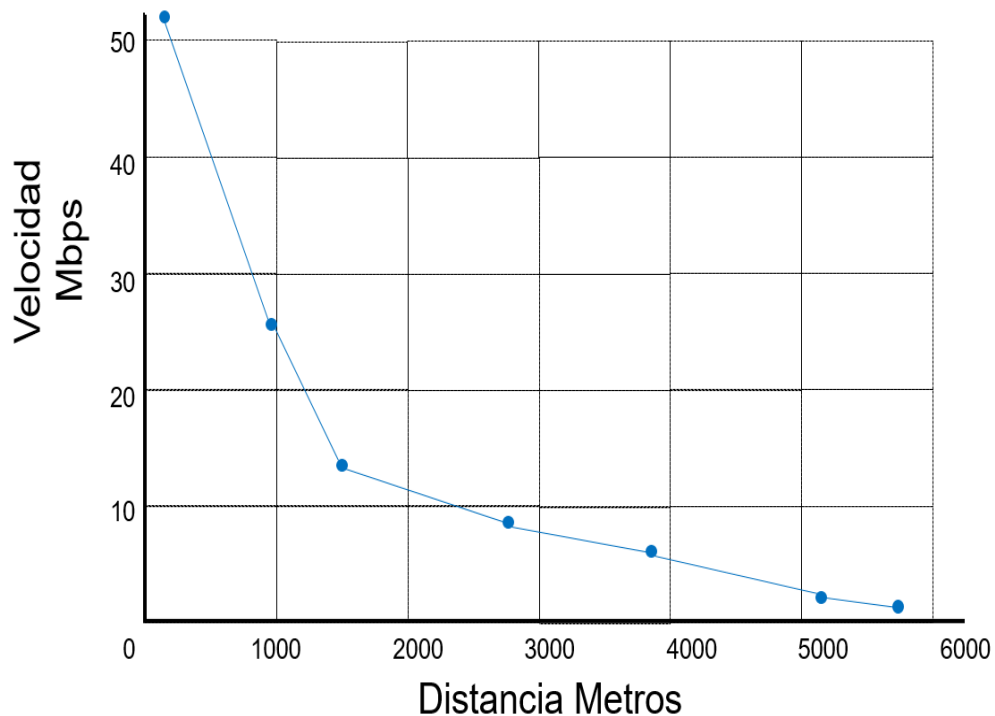
B-ISDN utiliza la tecnología **ATM** (*Asynchronous Transfer Mode*) que es la suma de 2 tecnologías previas; la **TDM** (Multiplexación por División de Tiempo) Síncrona y la **TDM** Estática, se utiliza para la conmutación en la red, la cual permite soportar varios servicios adaptándose a la transferencia de datos a velocidades fijas y variables dependiendo el tipo de servicio.

1.4.2 Tecnologías DSL

En la tecnología RDSI no se utiliza toda la capacidad del par de cobre simétrico del bucle de abonado por lo que se desarrolló la **DSL** (Línea Digital de Abonado, *Digital Subscriber Line*), que es un conjunto de tecnologías para establecer una red de banda ancha sobre la línea telefónica convencional, es decir, proporciona al abonado acceso a internet mediante la transmisión de datos digitales, a través del cable telefónico convencional ya que el bucle de abonado puede transmitir señales de frecuencia muy superior a la empleada por la voz [1.7]. Los servicios **xDSL** (*x Digital Subscriber Line*), donde *x* define al tipo de técnica dentro de las tecnologías **SDSL** (Línea Digital de Abonado Simétrico) y **ADSL** (Línea Digital de Abonado Asimétrico), diseñados bajo los siguientes objetivos:

- Deben ser funcionales en los lazos locales de par trenzado de categoría 3.
- No debe afectar a los otros servicios del cliente.
- Una velocidad mínima de 56 Kbps (Gráfica 1.1)²³.
- Estar siempre disponibles al cliente.
- Tener un costo mensual.

²³ [1.10] "Ancho de banda vs. Distancia sobre cable UTP categoría 3 para DSL" Página 128.



Gráfica 1.1 Funcionamiento DSL en cableado categoría 3.

1. **SDSL**: proporcionan igualdad en la tasa de transmisión de datos para ambas direcciones; subida (ascendente) y de bajada (descendente). Utiliza solo un par de cobre con transmisión dúplex con velocidades de 192 Kbps a 2.312 Kbps.
2. **ADSL**: proporciona una tasa de transmisión de datos distinta, la velocidad de bajada es diferente a la velocidad de subida, esto es que la velocidad de bajada es mayor a la de subida debido a que los usuarios adquieren más datos de los que aportan. Brinda 3 tipos de servicios a través de un solo par de cobre; canal de descarga de alta velocidad (1.5/6.1 Mbps de forma descendente), canal dúplex de velocidad media (16/640 Kbps) y un canal telefónico. Los canales se separan por Multiplexación por División de Frecuencia (MDF).

Técnicas xDSL:

- **ADSL Lite o G.lite (Línea Digital de Abonado Asimétrico Ligero).**

Es una variante de **ADSL** normal pero de baja velocidad, con la diferencia que elimina el divisor del canal telefónico con los otros 2 resultando con un ancho de banda menor. Proporciona una velocidad de bajada máxima de 1.5 Mbps y 512 Kbps de subida.

- **HDSL (Línea de Abonado Digital de Alta Velocidad, High-Bit-Rate DSL)**

Desarrollado en los años 80 por BellCore, es la primera tecnología DSL, su objetivo era mejorar la eficiencia del envío de datos a una velocidad de 1.544

Mbps dependiente del Sistema de Transmisión ¹²⁴ (T1) usando la codificación **AMI** (*Alternate Mark Inversion*) con un ancho de banda de 1.5 MHz y repetidores a 1 Km de distancia entre ellos. Para poder alcanzar la velocidad de 2.048 Mbps fue necesario ocupar la codificación 2B1Q con un ancho de banda de 196 KHz para una distancia máxima de 3,7 Km usando 2 pares de cobre.

- **VDSL (Línea de Abonado Digital de muy alta Tasa de Bits, Very-High-Bit rate DSL)**

Su objetivo es proporcionar un servicio basado en **ADSL** con velocidades de datos muy superiores pero con una distancia menor, usando una señalización **DMT/QAM** (Modulación por Multitono Discreto / Modulación de Amplitud en Cuadratura) con la capacidad de asignar un ancho de banda para cada servicio, la **PSTN** usa un ancho de banda de 0 Hz a 4 KHz, el tráfico de datos de subida es de 300 KHz a 700 KHz y el de bajada mayor a 1 MHz, finalmente cuenta con una velocidad máxima de 52 Mbps con una distancia local de 300 m a 1.8 m pudiendo ser simétrico o asimétrico.

- **RADSL (Línea de Abonado Digital de Tasa Adaptable, Rate-Adaptive DSL)**

En este diseño el software determina y ajusta la velocidad en que los datos pueden ser transmitidos a través del cableado de cobre telefónico con una velocidad descendente de 640 Kbps a 2.2 Mbps y 272 Kbps a 1.088 Mbps de forma ascendente.

- **IDSL (Línea de Abonado Digital basada en ISDN)**

Basado en la RDSI con codificación 2B1Q con velocidades de transferencia de datos de 128 Kbps, utilizando el ancho de banda de 2 canales tipo B de 64 Kbps más el de señalización de 16 Kbps con una distancia de lazo local de 5 Km.

- **DSLAM (Multiplexor de Acceso a Línea de Abonado Digital, Digital Subscriber Line Access Multiplexer)**

Se trata del dispositivo receptor alojado en la central telefónica, trabaja las comunicaciones **DSL**, brindando los servicios comerciales a los clientes.

²⁴ T1: Línea de transmisión de datos basada en PCM y Multiplexaje por División en el Tiempo (TDM) para 24 canales con 8 bits de datos cada uno.

Capítulo 2 Red de datos y telefonía IP

2. Redes de datos.

2.1 Historia.

Entre los años de 1950 y 1960, los datos se administraban con grandes sistemas u ordenadores centrales accesibles a partir de puestos externos [2.3], en el año 1958 se creó por parte del Departamento de Defensa de Estados Unidos la **ARPA** (Agencia de Investigación de Proyectos Avanzados, *Advanced Research Projects Agency*), quien deseaba una red de comando y control que pudiera soportar una guerra nuclear, dado que se encontraban en la cúspide de la Guerra Fría y todas las comunicaciones eran por la red telefónica pública.

ARPA dio origen en 1962 a la **IPTO** (Oficina de Técnicas de Procesamiento de Información, *Information Processing Techniques Office*), siendo el director Joseph Licklider, el encargado de iniciar el proyecto de red conocida como **ARPANET**, la primera red interactiva y dejando atrás la existencia de terminales autónomas diseñado con la tecnología de la conmutación de paquetes.

En 1969 la red **ARPANET** (Figura 2.1)²⁵ interconectaba cuatro host en Estados Unidos: la Universidad de California en los Ángeles, la Universidad de California en Santa Bárbara, la Universidad de Utah y el Instituto de Investigación de Stanford. Con el tiempo fue creciendo, en 1973 ya contaba con una interconexión de 36 host conectados a través de **IMP** (Procesadores de Mensajes de Interfaz) o mejor conocidos como nodos.

²⁵ Internet: así nació y así creció: <https://blog.orange.es/red/historia-de-internet/>

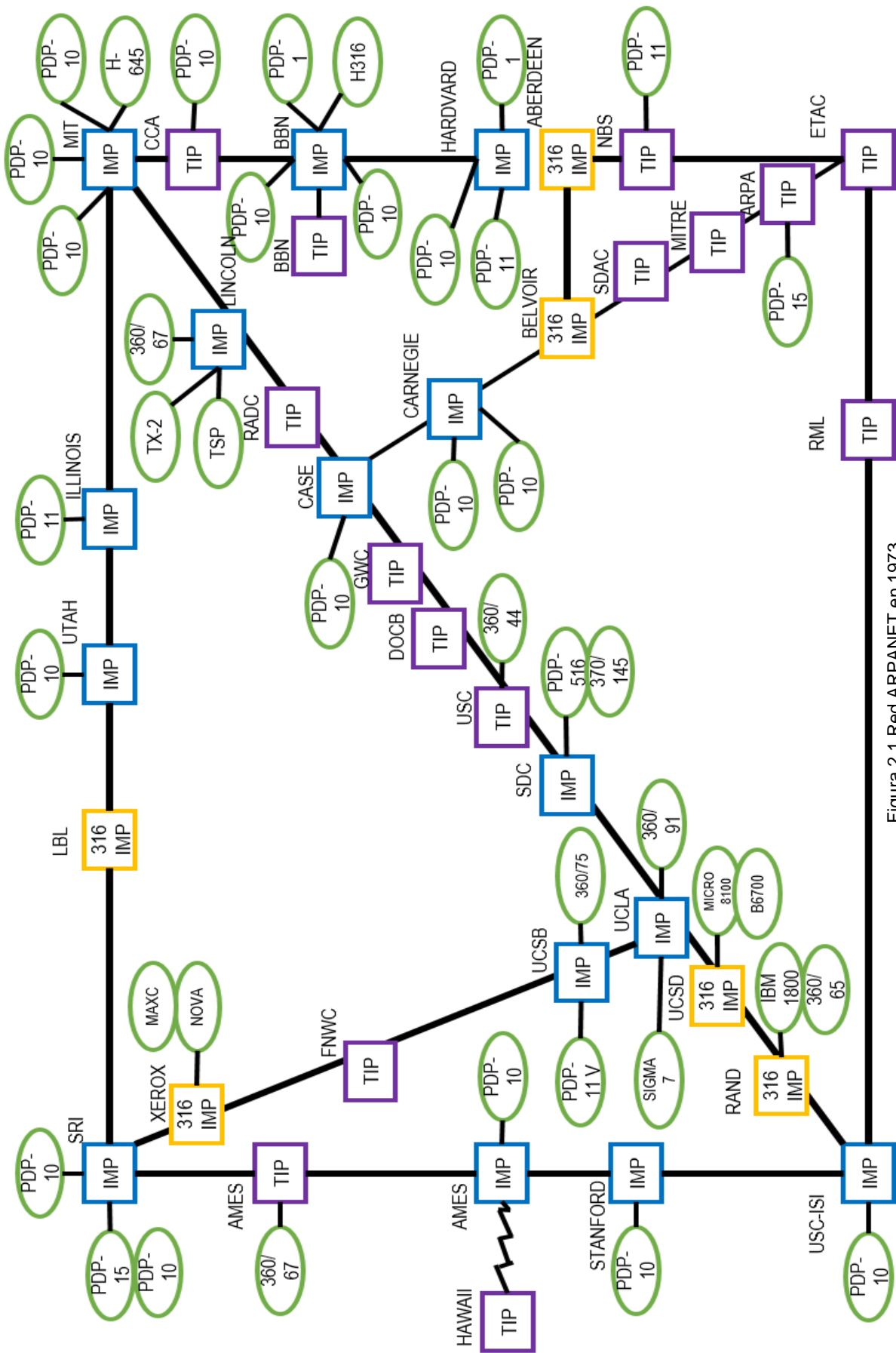


Figura 2.1 Red ARPANET en 1973.

Posteriormente el siguiente paso consistió en posibilitar la conexión de **ARPANET** con otras redes de ordenadores, comenzando por las redes de comunicación que **ARPA** estaba gestionando: **PRNET** (*Packet Radio Network*) y **SATNET** (*Atlantic Packet Satellite Network*). Esta posibilidad introdujo un nuevo concepto: la red de redes [2.2]. Pero el problema era cuando las redes **PRNET** y **SATNET** querían conectarse con **ARPANET**, esto era que todas las redes tenían diseños diferentes, impidiendo compartir recursos entre redes. Gracias a este problema Vinton Cerf y Robert Kahn establecieron unos protocolos de comunicación estandarizados conocido como el modelo **TCP** (Protocolo de Control de Transmisión) y posteriormente **TCP/IP**, así, cualquier red de computadoras que quisiera conectarse a **ARPANET** tenía que contar con este modelo.

Más adelante existiría el problema que la red **ARPANET** no estaba diseñada para ser la red que soportaría todas las conexiones en Estados Unidos, ya que solo tenía su función en investigación y desarrollo y la seguridad de la misma red. Por lo que se creó la **NSFNET** (Red de la Fundación Nacional para la Ciencia, *National Science Foundation's Network*) (Figura 2.2)²⁶ la cual fue diseñada teniendo una red abierta para las universidades y centros de investigación, para tiempo después privatizar el acceso a la red la cual hizo que fracasara ésta, cerrando en 1995.

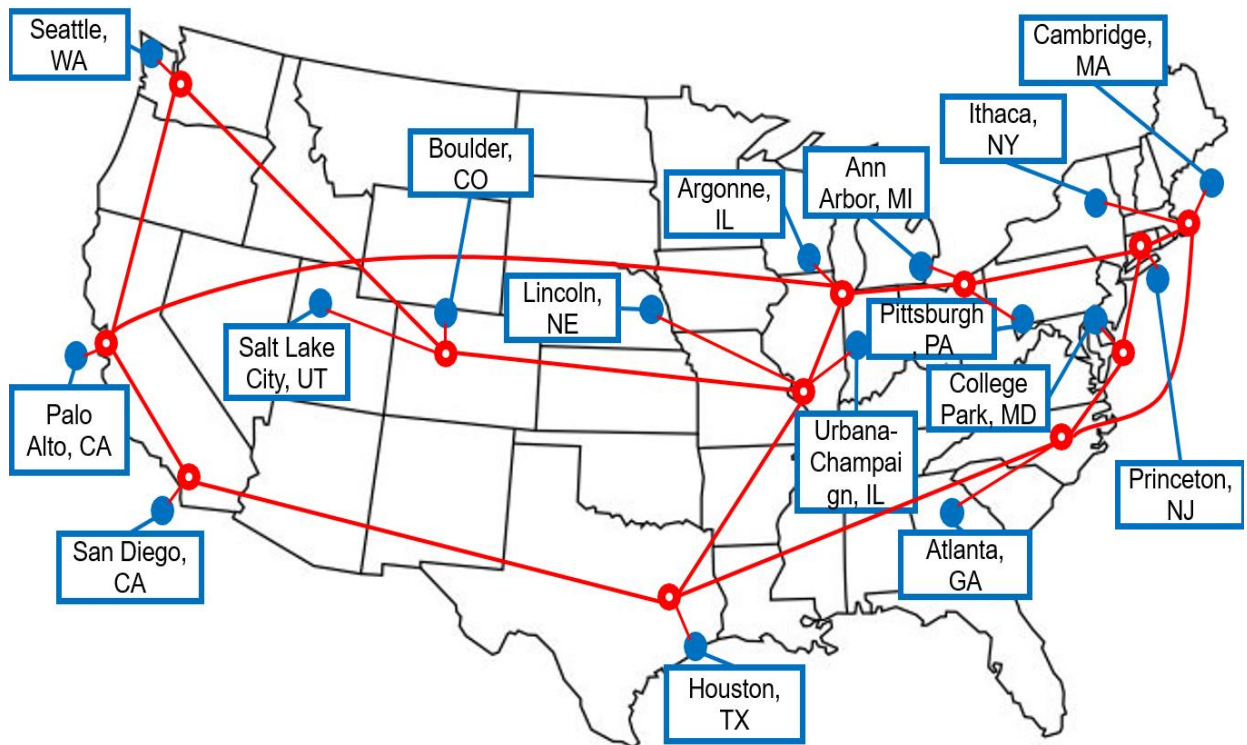


Figura 2.2 Red NSFNET en 1992

²⁶ NSFNET - National Science Foundation's Network: <https://internet10.com.mx/9-historia/internet/289-nsfnet-national-science-foundations-network>

Tras el cierre de **NSFNET**, varios proveedores de servicio de redes que tenían troncales centrales (*Backbone*) propias, decidieron conectarse entre ellas (*Gateway*) para así crear la red más grande: el Internet que usa el modelo **TCP/IP**.

2.2 Medios físicos para la transmisión de datos.

Para la transmisión de información son requeridos los medios de transmisión, que es el medio físico de conexión que existe entre un host origen y un host destino. Estos medios de transmisión se clasifican en (Figura 2.3)²⁷:

- Medios guiados: Transmisión a través de medios sólidos, presentan la ventaja de permitir un ancho de banda muy elevado y ser menos susceptibles a las interferencias [2.1] dependiendo de la distancia y si es de punto a punto o multipunto.
 - Medios Magnéticos.
 - Par Trenzado.
 - Cable Coaxial.
 - Líneas Eléctricas.
 - Fibra Óptica.

- Medios no guiados: Transmisión a través de espacio libre, disponen de las ondas electromagnéticas.
 - Radio.
 - Microondas.

Cada uno de ellos teniendo sus propias características como:

- Ondas electromagnéticas.
- Velocidad por el medio.
- Atenuación.
- Interferencias y ruido.
- Ancho de banda.

²⁷ [2.1] “Medios para la transmisión de datos”. Página 498

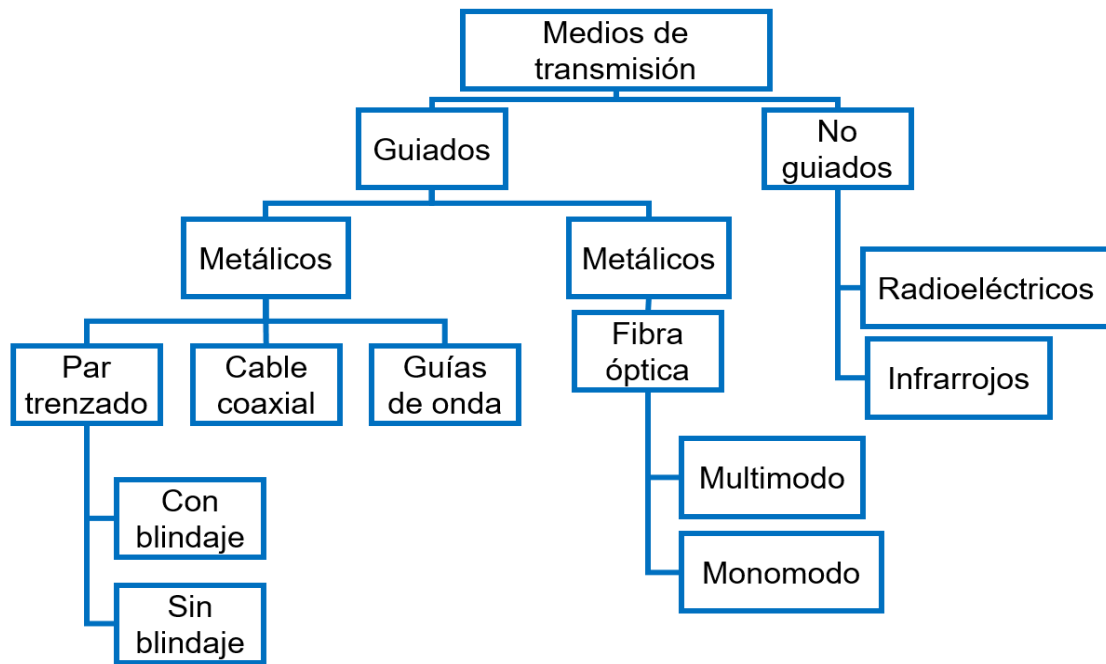


Figura 2.3 Medios de transmisión de datos.

2.2.1 Par trenzado

El par trenzado es el medio más barato y mayormente utilizado, está diseñado por un núcleo de hilos de cobre rodeados por un aislante. Los hilos se encuentran trenzados en espiral por pares, de forma que cada par se puede utilizar para la transmisión de datos [2.1]. El ejemplo más destacado de esto es el cableado de la red telefónica, casi todos los teléfonos se conectan a la central telefónica por este medio, también se utiliza para la transmisión de datos tanto analógicos como digitales.

Este medio de transmisión tiene sus limitantes en cuanto al ancho de banda, que depende del grosor del cable, distancia y velocidad de transmisión, por ello es ideal para redes tipo **LAN** (*Local Area Network*).

Existen variedad de categorías de cableado de par trenzado y se dividen en los siguientes grupos (Tabla 2.1)²⁸:

- **UTP** (Par Trenzado sin Blindaje o no apantallado): Únicamente diseñado por alambres y aislantes.
 - **Categoría 1:** Fue la categoría que se instaló para el servicio de telefonía durante 1983, transmite únicamente voz.
 - **Categoría 2:** Transmite hasta 4Mbps, tiene 4 pares trenzados con la siguiente configuración de colores:
 - Primer par: Blanco/Azul – Azul.
 - Segundo par: Blanco/Naranja – Naranja.

²⁸ [1.9] “Especificaciones IEEE 802.3” Página 252, [1.10] “Par trenzado” Página 84 y [2.1] “Categorías del par trenzado” Página 500.

- Tercer par: Blanco/Verde – Verde.
 - Cuarto par: Blanco/Marrón – Marrón.
- **Categoría 3:** Transmite hasta 10 Mbps y un ancho de banda de 16 MHz (dependiendo de la topología y tipo de red). Se puede utilizar para redes Token Ring a 4 Mbps y Ethernet 10Base-T a 10 Mbps. Utiliza la misma configuración de colores de la categoría 2 (configuración estándar) y cuenta con 10 rizados por metro de cable. La distancia estándar de aplicación es de 100 metros.
- **Categoría 4:** Transmite hasta 16 Mbps y un ancho de banda de 20 MHz (dependiendo de la topología y tipo de red). Pudiendo ser utilizado en redes Token Ring a 20 Mbps. Teniendo 4 pares trenzados con la configuración estándar. La distancia estándar de aplicación es de 100 metros.
- **Categoría 5:** Puede transmitir 100 Mbps y un ancho de banda de 100 MHz (dependiendo de la topología y tipo de red). Tiene 4 pares trenzados con la configuración estándar. La distancia estándar de aplicación es de 100 metros.
- **STP (Par Trenzado con Blindaje o Apantallado):** Este tipo de cable cuenta con una malla metálica que reduce las interferencias externas y el uso de distintos pasos de torsión entre los pares adyacentes reduce la diafonía [1.9].
 - **FTP:** Cable de par trenzado blindado.
 - **SFTP:** Cable laminado blindado individualmente (*Screened Foiled Twisted Pair*).
 - **SSTP:** Par trenzado de blindado individual (*Screened Shielded Twisted Pair*).

Categoría	Ancho de banda	Velocidad	Uso	Tipo
Categoría 1	-	-	Telefonía	UTP
Categoría 2	-	4 Mbps	En desuso	UTP
Categoría 3	16 MHz	10 a 16 Mbps	10Base-T / Ethernet 100Base-T4	UTP
Categoría 4	20 MHz	16 Mbps	Token Ring / Ethernet 10Base-T	UTP
Categoría 5	100 MHz	100 Mbps	10Base-T / Ethernet 100Base-TX	UTP
Categoría 5e	100 MHz	1 Gbps	100Base-TX / Ethernet 1000Base-T	UTP / FTP
Categoría 6	250 MHz	1Gbps	Ethernet 1000Base-T	FTP / STP / SFTP / SSTP
Categoría 6e	500 MHz	10 Gbps	Ethernet 10GBase-T	FTP / STP / SFTP / SSTP
Categoría 7	600 MHz	10 Gbps	Telefonía / Televisión / Ethernet 1000Base-T	FTP / STP / SFTP / SSTP
Categoría 7a	1000 MHz	10 Gbps	Telefonía / Televisión / Ethernet 1000Base-T	SFTP / SSTP
Categoría 8	2000 MHz	40 Gbps	Ethernet 40GBase-T / Telefonía / Televisión / Ethernet 1000Base-T	SFTP / SSTP

Tabla 2.1 Tipos de cables de par trenzado. [Elaboración propia]

En resumen el par trenzado es económico y fácil de instalar, el **UTP** funciona para telefonía de calidad y **STP** para transferencia de datos.

2.2.2 Cable Coaxial

El cable coaxial (Figura 2.4)²⁹ es otro medio físico de transmisión que a diferencia del par trenzado tiene mucho mayor ancho de banda (dependiendo de la calidad y longitud), velocidad de envío de datos y puede cubrir más distancia. Existen dos tipos de este medio físico:

- **50 ohms:** Para transmisión de datos en formato digital como el cable **RG-58** que es utilizado para Ethernet 10Base2.
- **75 ohms:** Para transmisión de datos en formato analógico y televisión por cable como por ejemplo **RG-11** utilizado para CATV (*Community Antenna Television*) o **RG-59** para banda ancha.

Al igual que el par trenzado el cable coaxial también cuenta con dos conductores pero diseñados para trabajar en un rango de frecuencias mayor y de forma concéntrica, contiene como núcleo un alambre de cobre cubierto por un material dieléctrico, que a su vez esta es rodeado por un conductor en forma de malla (conductor externo) y por ultimo estos tres son envueltos por una cubierta protectora de plástico, con diámetro de 1 cm y 2.5 cm.

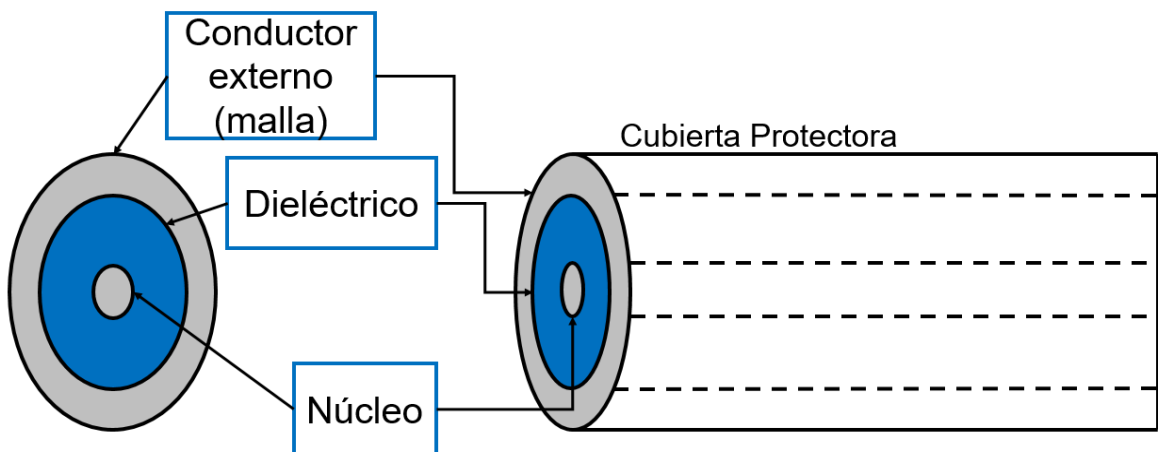


Figura 2.4 Cable coaxial.

Aún se utiliza para:

- Televisión por cable y acceso a Internet (Medio compartido).
- Redes de área metropolitana.
- Telefonía a larga distancia (Remplazado casi en su totalidad por fibra óptica).
- Redes **LAN**.

Una de sus características es el ancho de banda de este medio de transmisión físico, de 5 MHz a 1 GHz con una velocidad de envío de datos de 10 Mbps, con una distancia recomendable de 100 m. Las limitaciones (Tabla 2.2)³⁰ del cable coaxial son la

²⁹ [1.9] "Medios de transmisión guiados. Figura b. Cable coaxial". Página 98.

³⁰ [2.1] "Medios para la transmisión de datos". Página 503.

atenuación, el ruido de intermodulación (al usar varios canales y bandas de frecuencias a la vez) y ruido térmico.

Tipo de cable	Impedancia (Ω)	Diámetro Exterior (mm)	Atenuación (dB/Km)	Frecuencia (MHz)	Distancia (m)
CATV	75	19	1	5	305
		12.7	1.5	5	305
		10.5	2	4	302
RG11/U	75	10.3	4.5	4	305
			5.1	5	305
			0.66	10	30.5
RG59/U	75	6.25	8	5	305
			1.1	10	30.5
RG58A/U	50	5	6.2	150	30.5
			12.5	450	30.5
			20	900	30.5
RG213/U	50	10.3	2.3	150	30.5
			4.5	450	30.5
			7.6	900	30.5

Tabla 2.2 Características de transmisión del cable coaxial.

2.2.3 Líneas Eléctricas

Las líneas que suministran energía a los hogares también pueden ser medios compartidos de comunicación como la telefonía y la televisión pero a baja velocidad (Figura 2.5).

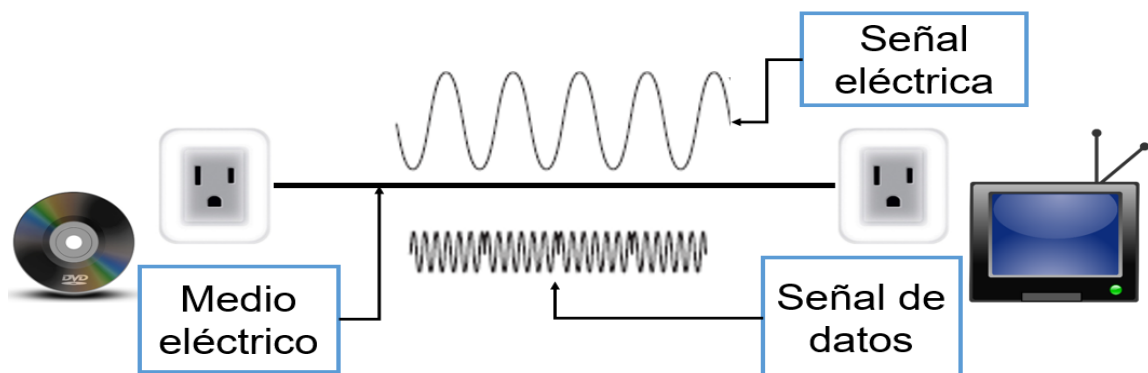


Figura 2.5 Medio compartido por cable eléctrico. [Elaboración propia]

Las señales de suministro de energía se transmiten a una frecuencia de 50 a 60 Hz, el cableado atenúa todas las señales de frecuencias más altas que son necesarias para la comunicación de datos de alta velocidad, sin embargo se puede enviar datos con una velocidad de 100 Mbps.

2.2.4 Fibra Óptica (FO)

La fibra óptica es un medio flexible y de muy pequeño espesor que conduce pulsos de luz, representando cada pulso un bit. La fibra óptica es inmune a las interferencias electromagnéticas, presenta una atenuación de la señal muy baja para distancias de hasta 100 kilómetros [2.6], por esto es el medio por excelencia para enlaces a largas distancias, pero también son ocupados para los troncales de Internet. Pero sus buenas características la elevan a costos muy altos.

Sus mayores ventajas con respecto a otros medios físicos de comunicación son:

- **Capacidad:** Un ancho de banda grande y derivado de ello una velocidad de transmisión considerable. Con unos cientos de Gbps en una docena de km.
- **Peso y tamaño:** Es visiblemente más delgada que un cable coaxial.
- **Atenuación:** Es bastante menor que el cable coaxial o el par trenzado, alcanzando mayores distancias.
- **Aislamiento electromagnético:** La comunicación no es afectada por los campos electromagnéticos.
- **Pérdidas:** Disminuyen considerablemente las pérdidas de información, dependiendo de la calidad de la fibra y los empalmes realizados en los dispositivos del sistema.

Un cable de fibra óptica está hecha por tres partes concéntricas (Figura 2.6)³¹:

- **Núcleo:** Hecho de vidrio y de plásticos con un diámetro de 50 y 300 μm . la elección de estas características depende al diseño del enlace.
- **Revestimiento:** Es la envoltura del núcleo siendo de vidrio o plástico, con diferentes propiedades ópticas a las del núcleo, haciendo que el haz de luz de datos sufra reflexiones a través de la fibra.
- **Cubierta:** Siendo el forro de la fibra hecha de plástico, y su función es proteger las fibras de humedad, compresiones y otros problemas.

³¹ [1.9] “Medios de transmisión guiados. Figura c. Fibra óptica”. Página 98.

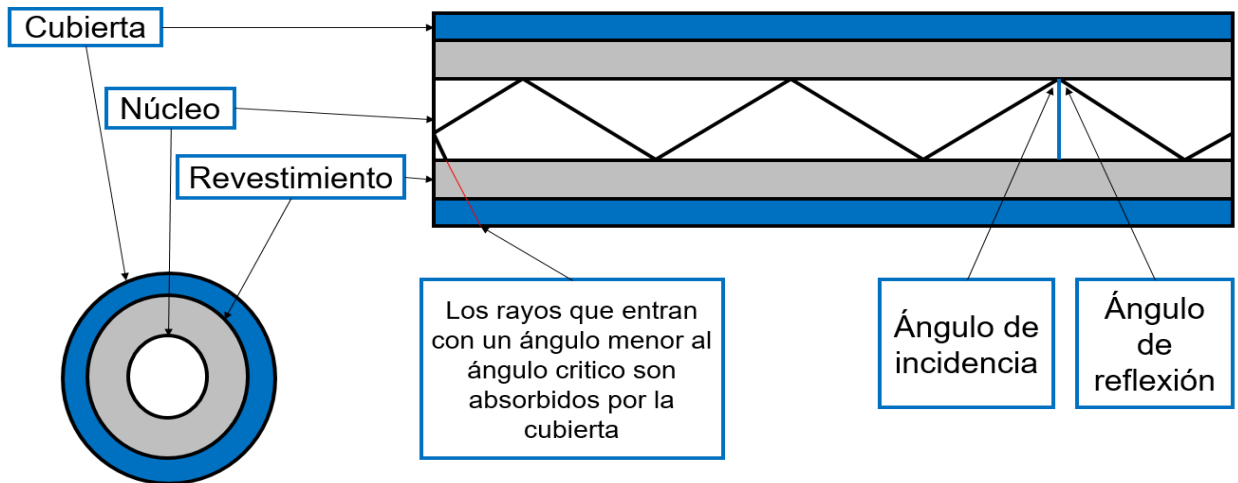


Figura 2.6 Fibra óptica.

El núcleo debe contar con cierto índice de refracción (es el cociente entre la velocidad de la luz en el vacío y la velocidad de propagación en el material) y el revestimiento debe contar con otro índice de refracción menor al del núcleo para poder mantener el haz de luz dentro de la fibra y este sufra reflexiones, permitiendo su propagación. Así, la propagación de la luz en la fibra se efectúa en formas o “modos” diferentes que dependen de la relación y variación de los índices de refracción tanto del núcleo como del revestimiento [2.1] y se dividen en:

- **Multimodo de índice escalonado** (Figura 2.7)³²: Dentro de la fibra viajan muchos rayos de luz con diferentes modos, los cuales se transmiten a la misma velocidad gracias a los diferentes índices de reflexión de los materiales del núcleo y la cubierta, pero con diferentes tiempos de llegada, ya que para ángulos más grandes el recorrido del rayo de luz es más grande.

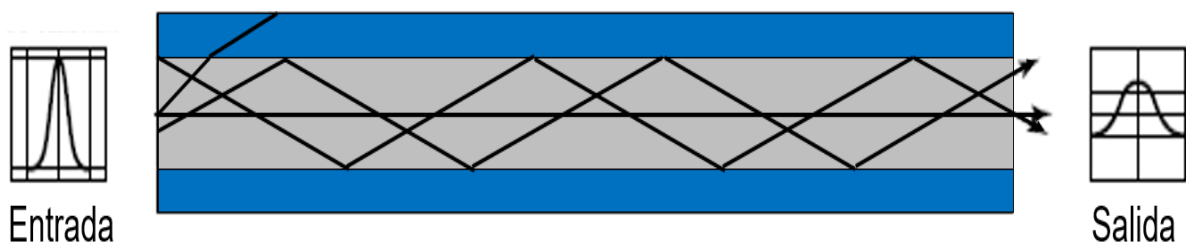


Figura 2.7 Fibra óptica - Multimodo.

- **Multimodo de índice gradual** (Figura 2.8)³³: Se consigue este modo de transmisión variando gradualmente el índice de refracción del núcleo. En lugar de sufrir las reflexiones en el límite entre núcleo y cubierta, el rayo de luz permanece muy cercano al centro del núcleo y describe curvas helicoidales,

³² [1.9] “Modos de transmisión en la fibras ópticas. (a) Multimodo de índice discreto”. Página 107.

³³ [1.9] “Modos de transmisión en la fibras ópticas. (b) Multimodo de índice gradual”. Página 107.

esto causado por las variaciones graduales del índice de refracción, reduciendo la longitud recorrida.

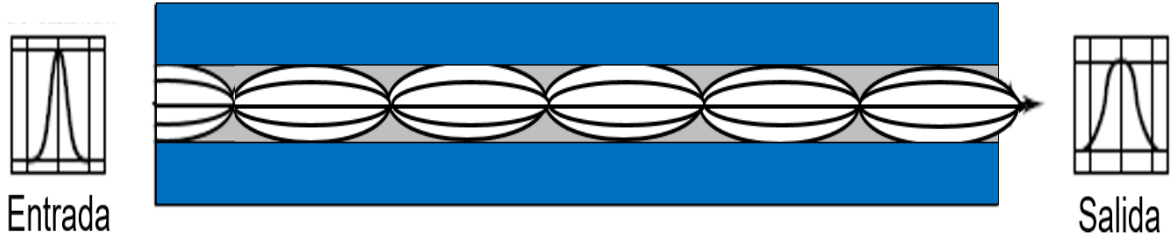


Figura 2.8 Fibra óptica – Multimodo de índice gradual.

- **Monomodo de índice escalonado** (Figura 2.9)³⁴: Un solo rayo de luz viaja por este tipo de fibra óptica, el cual se transmite de un solo modo por el núcleo.

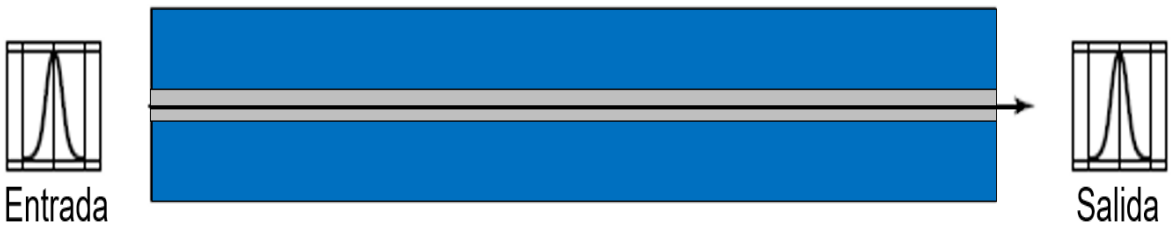


Figura 2.9 Figura #. Fibra óptica - Monomodo.

La atenuación en la fibra (Figura 2.10, tomada de la referencia [1.10])³⁵ tiene que ver de igual manera que en otros medios de transmisión, con la distancia y de las propiedades por el medio por el que se propaga, esta es la relación que hay entre la señal de entrada con la señal de salida. Utiliza solo tres ventanas de trabajo: la banda de 0.85 μm , 1.30 μm y 1.55 μm

³⁴ [1.9] "Modos de transmisión en la fibras ópticas. (c) Monomodo". Página 107.

³⁵ [1.10] "Atenuación de la luz dentro de una fibra en la región de infrarrojo.". Página 88.

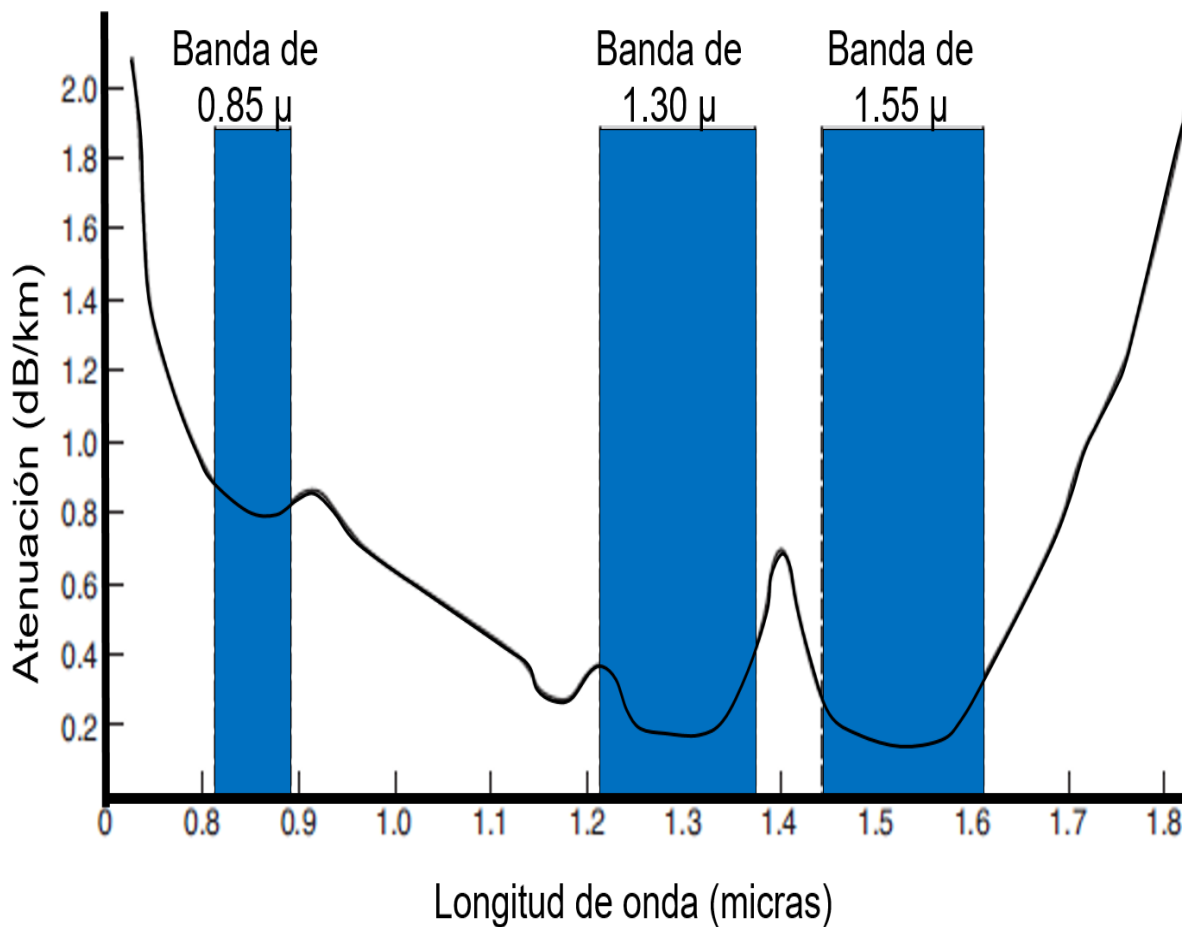


Figura 2.10 Atenuación de la fibra óptica.

2.2.5 Transmisión inalámbrica.

Este tipo de transmisión de información es por medios no guiados como en los casos anteriores (par trenzado, cable coaxial o fibra óptica), por lo que maneja las ondas de radio frecuencia de forma omnidireccional evitando el cableado físico.

Las señales de radio dependen de la frecuencia, por lo que a bajas frecuencias (30 a 300 kHz), la señal enviada tiene mayor probabilidad de cruzar los obstáculos en su trayectoria a diferencia de una señal de radio en altas frecuencias (3 – 30 MHz). Estas señales son susceptibles a la atenuación que sufre al traspasar los diversos obstáculos existentes entre el transmisor y el receptor, al igual por las diferentes reflexiones de la señal enviada.

Las ondas de radio en las bandas de frecuencia (Figura 2.11)³⁶ VLF³⁷ (3 – 30 kHz), LF³⁸ (30 – 300 kHz) y MF (300 – 3000 kHz), son capaces de transmitirse de forma paralela a la curvatura de la Tierra, pudiendo recorrer distancias considerables de

³⁶ [2.1] “En las bandas VLF, LF y MF, las ondas de radio siguen la curvatura de la Tierra. En la banda HF, rebotan en la ionosfera.” Página 95.

³⁷ Muy bajas frecuencias, Very Low Frequency.

³⁸ Bajas frecuencias, Low Frequency.

hasta 1,000 km. Un ejemplo de esto son las emisoras de radio AM (Amplitud Modulada, Amplitude Modulation), la cual se transmite en las bandas MF y la señal puede ser captada a distancias relativamente largas. Por otro lado, las ondas de radio en las bandas de frecuencia HF³⁹ (3 - 30 MHz) y VHF⁴⁰ (30 – 300 MHz) son refractadas por la ionosfera de la Tierra y son enviadas de nuevo a nuestro planeta.

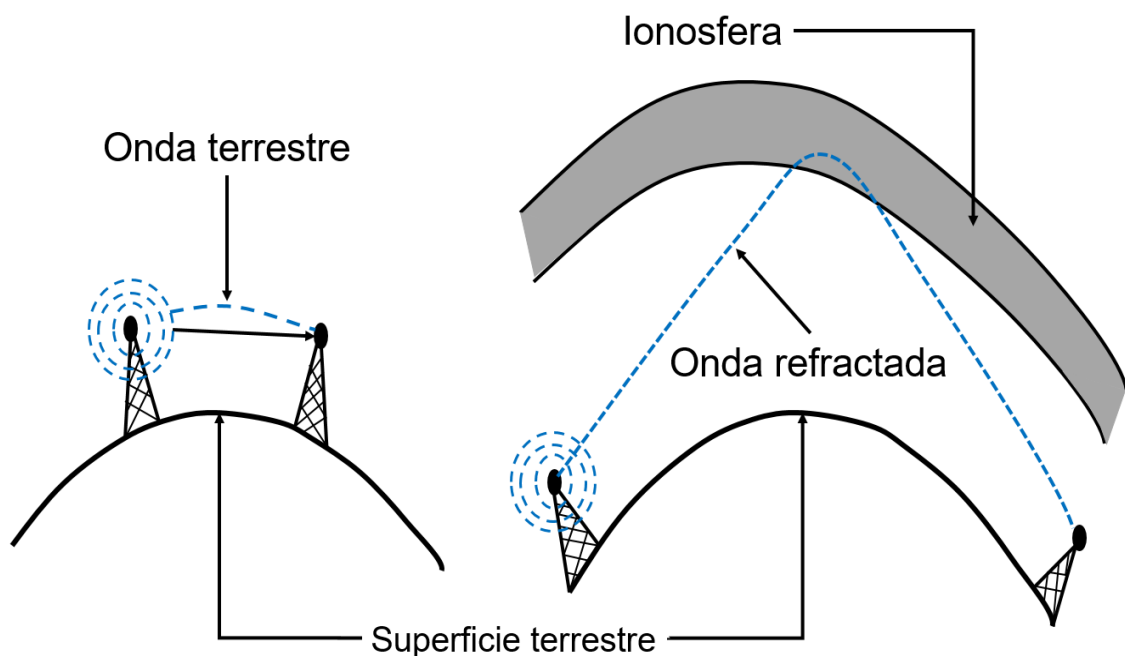


Figura 2.11. Señales de radio.

2.2.6 Espectro electromagnético

División del espectro electromagnético entre los diferentes medios de transmisión (Tabla 2.3)⁴¹:

Frecuencia	Longitud de onda	Designación	Medio	Aplicaciones
3Hz - 30KHz	10^8 - 10^4 m	Frecuencias muy bajas VLF	Conductores metálicos, Radio	Audio. Telefonía, Transmisión de datos, Radionavegación
30KHz - 300KHz	10^4 - 10^3 m	Frecuencias bajas LF	Conductores metálicos, Radio	Radioayudas, Radiofaros, Transmisión de portadora (PLC)

Tabla 2.3 Parte I. Uso del espectro electromagnético.

³⁹ Altas frecuencias, High frequency.

⁴⁰ Muy altas frecuencias, Very High Frequency.

⁴¹ [2.1] "El espectro electromagnético". Página 537.

Frecuencia	Longitud de onda	Designación	Medio	Aplicaciones
300KHz - 3MHz	10^3 - 10^2 m	Frecuencias Medias MF	Cable Coaxial, Radio de onda corta	Radiodifusión comercial, Defensa civil, Radioaficionados
3MHz - 30MHz	10^2 - 10 m	Frecuencia Alta HF	Cable Coaxial, Radio de onda corta	Radiodifusión comercial, Defensa civil, Comunicaciones militares
30MHz - 300MHz	0 - 1 m	Frecuencias Muy Altas VHF	Cable Coaxial, Radio de onda corta	Televisión VHF, Radio FM, Control de tránsito aéreo, Radiotaxis, Policía
300MHz - 3GHz	100 - 1 cm	Frecuencias Ultra Altas UHF	Radio de onda corta, Guías, Microondas	Televisión UHF, Telemetría espacial, Comunicaciones militares, Banda ciudadana
3GHz - 30GHz	10 - 1 cm	Frecuencias Súper Altas SHF	Guías de onda, Microondas	Radar, Comunicación por satélite, Radioenlaces de microondas
30GHz - 300GHz	< 1 cm	Frecuencias Extra Altas EHF	Guías de onda, Microondas, Fibra óptica	Radioastronomía, Servicio de ferrocarriles, Sistemas experimentales, Comunicaciones ópticas

Tabla 2.3 Parte II. Uso del espectro electromagnético.

2.3 Modelo de referencia OSI de ISO.

El Modelo de referencia **OSI** (Interconexión de Sistemas Abiertos, *Open Systems Interconnection*) desarrollado por la **ISO** (Organización Internacional de Normalización, *International Organization for Standardization*) en 1977 y publicada hasta 1984, es el modelo que se ocupa de las conexiones de sistemas abiertos a la comunicación con otros sistemas, las funciones de comunicación se distribuyen en un conjunto jerárquico de capas. Cada capa realiza un subconjunto de tareas relacionadas entre sí, las necesarias para llegar a comunicarse con otros sistemas [1.9]. Este modelo ofrece una explicación sencilla de la relación entre los componentes de hardware y el protocolo de red.

Los lineamientos que se abordaron para la creación de las siete capas son:

1. Debe existir un nivel de abstracción distinta en cada una de ellas.
2. Cada capa realizará una función definida y diferente, guiada por los protocolos de estandarización.
3. Delimitar las funciones de cada una de las capas de modo que se minimice el manejo de información y se evite la sobrecarga, tomando en cuenta que cada capa debe ser lo bastante pequeña para como poder manejar el sistema.

Sin tener éxito en su finalidad, es importante el conocimiento del modelo **OSI**, dado que se toma como referencia para otras arquitecturas.

Las tres primeras capas (Física, Enlace de Datos y Red) son las que permiten el tráfico a través de la red y las siguientes cuatro (Transporte, Sesión, Presentación y Aplicación) solo se usan en el host que envía y el host que recibe (Figura 2.11)⁴².

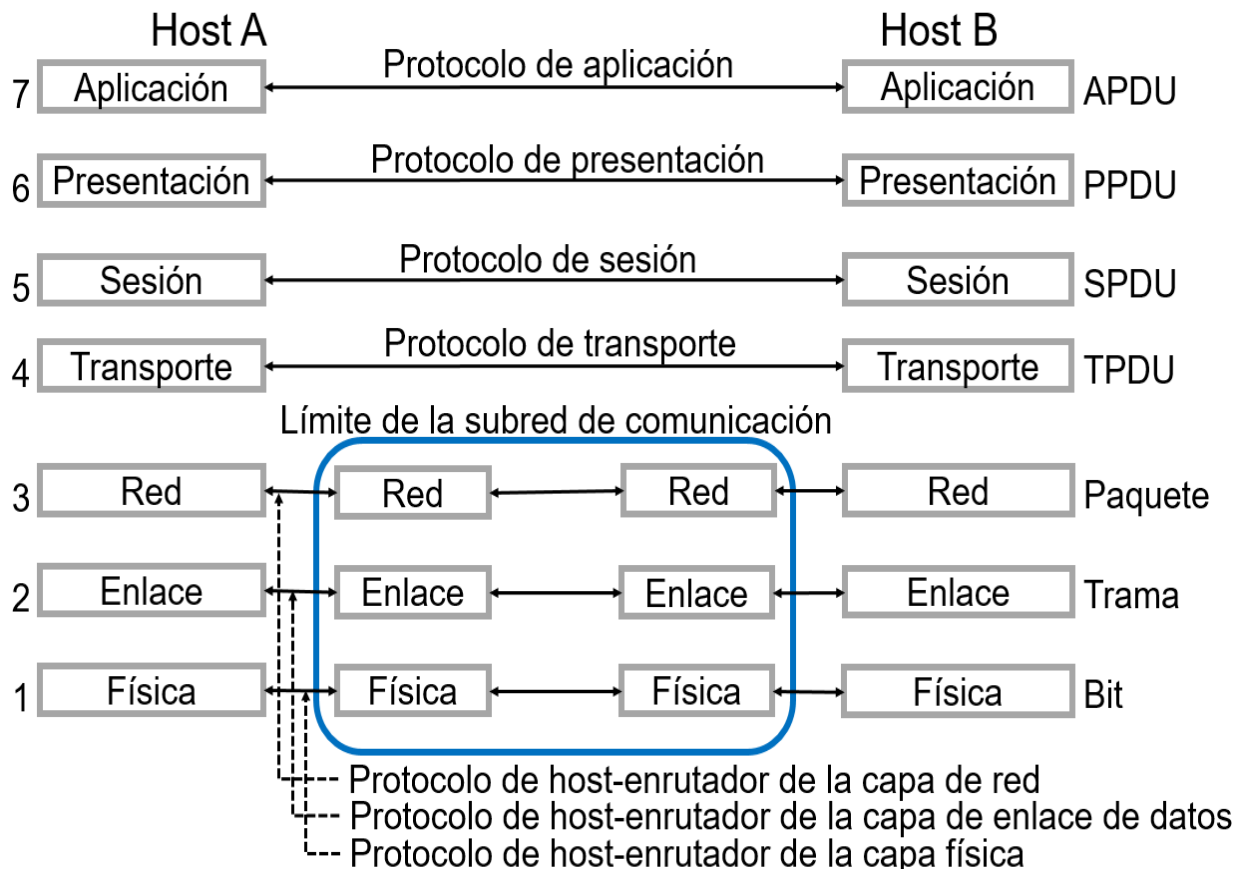


Figura 2.12 Modelo de referencia OSI.

⁴² [1.10] "El modelo de referencia OSI". Página 36.

2.3.1 Capa Física

Esta capa se encarga de la transmisión de bits a través de un canal de transmisión. La función de esta capa es la de asegurarse que cuando uno de los host del enlace envíe un bit 1 el otro host enlazado lo reciba como un bit 1 y no como un bit 0. Maneja las características eléctricas, mecánicas, funcionales y los procedimientos que se requieren para mandar los bits de datos en el enlace de comunicación.

También define los medios de transmisión tales como:

- **Medios guiados o físicos:** Par trenzado, cable coaxial, líneas eléctricas o fibra óptica.
- **Medios inalámbricos:** Radiotransmisión, microondas o infrarrojo.

Y por último asigna el tipo de modulación, filtrado y detección.

- **MDF o FDM** (Multiplexaje por División de Frecuencia).
- **MDT o TDM** (Multiplexaje por División de Tiempo).
- **MDC o CDM** (Multiplexaje por División de Código).

2.3.2 Capa de Enlace de datos

La tarea fundamental de esta capa es transformar un medio de transmisión en una línea que esté libre de errores en el traslado de los datos entre los equipos directamente conectados (Figura 2.12)⁴³. Envía bloques de datos llamados tramas (paquetes procedentes de la Capa de Red encapsulados) en forma secuencial llevando a cabo el enlace, el control de errores de transmisión y regular el flujo de datos (los dos últimos también se encuentran en capas superiores o en los protocolos). Cada trama contiene un encabezado, un campo de carga útil para almacenar el paquete y un terminador [1.10].

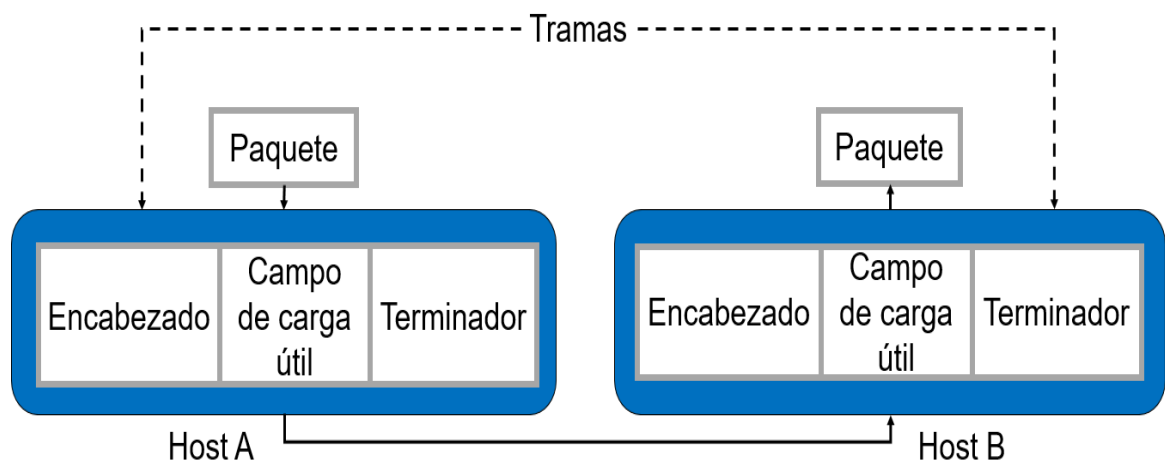


Figura 2.13 Relación entre paquetes y tramas.

⁴³ [1.10] "Relación entre paquetes y tramas". Página 168.

La Capa de Enlace de datos brinda varios servicios a la Capa de Red, como el envío de unos cuantos bits de los procesos realizados por la Capa de Red del host que envía al que recibe. De forma general esta capa prevé los siguientes servicios:

- **Sin conexión lógica ni confirmación** del receptor al transmisor de la llegada de datos. Ideal para sistemas con tasa de error muy pequeña y confiabilidad alta como Ethernet.
- **Sin conexión lógica con confirmación** del receptor al transmisor de la llegada de datos. El emisor siempre conoce el estado de los datos recibidos en el otro extremo, ya sea que los datos han llegado correctamente o la existencia de pérdidas en ellos, de esta forma el transmisor envía las tramas siguientes o repite las perdidas. Este servicio es ideal para sistemas con baja confiabilidad como en los medios inalámbricos.
- **Orientado a conexión con confirmación** del receptor al transmisor de la llegada de datos. Es el servicio más detallado dado que las tramas se enumeran individualmente y de la misma manera cada una de ellas tiene su confirmación de llegada, garantizando una conexión fiable. Es recomendable este servicio para sistemas con poca confiabilidad y con una longitud muy grande entre los host como un circuito de telefonía.

Se determina el servicio que brindará la Capa de Enlace de datos a la Capa de Red dependiendo del sistema de comunicación, pero para el control de errores es mayormente eficiente el servicio orientado a conexión con confirmación, ya que se puede monitorear el estado de cada trama de forma que si se pierde alguna el transmisor sabe específicamente cual debe volver a enviar, además de la existencia de un temporizador en los envíos de las tramas.

2.3.3 Capa de Red

Es responsable del establecimiento, mantenimiento y finalización de las conexiones que se realizan para la comunicación entre los host conectados a la red. Esta capa debe conocer la topología de la red, elegir las rutas apropiadas incluso para redes más grandes. También debe tener cuidado al escoger las rutas para no sobrecargar algunas de las líneas de comunicación y los enrutadores, y dejar inactivos a otros [1.10]. Y finalmente debe resolver los problemas presentes en una interconexión de equipos con topologías distintas.

Para el diseño de esta Capa de Red es necesario decretar la ruta de los paquetes desde el host origen hasta el destino. La existencia de excesivos paquetes en la subred en el mismo tiempo creará colisiones⁴⁴ entre los muchos paquetes y formarán cuellos de botella haciendo un sistema lento y con pérdidas, el manejo de esta congestión es

⁴⁴ Colisiones: Si dos o más tramas pertenecientes a la red transmiten de forma simultánea, las señales enviadas se traslapan en el tiempo y ambas alteran su contenido.

responsabilidad de la Capa de Red, trabajando en conjunto con las capas superiores las cuales manejan la carga de paquetes que envían en la red.

Esta capa tiene un algoritmo de enrutamiento, el cual se encarga de realizar tablas en donde es mayormente eficiente enviar los datos obtenidos en los routers. Así el algoritmo debe tener:

- **Exactitud:** Debe ser fiel a los datos asignados.
- **Sencillez:** Debe realizar el enrutamiento de la manera más fácil y rápida.
- **Robustez:** Debe resolver los problemas presentes en los cambios de topología y tráfico.
- **Estabilidad:** Debe coincidir con el enrutamiento establecido.
- **Equidad:** Se debe permitir el flujo total de toda la red, permitiendo la comunicación entre diferentes host.
- **Eficiencia:** Todas las interconexiones deben ser satisfactorias.

Para el problema de la congestión presente en la Capa de Red, tanto esta capa y la de Transporte deben resolver este conflicto en conjunto, haciendo que la Capa de Transporte no sature de datos a la red. El control de congestión se ocupa de asegurar que la red sea capaz de transportar el tráfico ofrecido. Es un asunto global, en el que interviene el comportamiento de todos los hosts y enrutadores [1.10].

Un último problema para resolver en la Capa de Red, es el buen funcionamiento que debe tener en la interconexión con los diferentes tipos de redes como **LAN** (*Local Area Network*), **MAN** (*Metropolitan Area Network*), **WAN** (*Wide Area Network*), Internet, **PAN** (*Personal Area Network*), **CAN** (*Campus Area Network*) y redes inalámbricas, las cuales manejan diferentes tipos de protocolos. La diferencia que existe entre estas diferentes redes se resume en los siguientes aspectos:

- Servicios ofrecidos.
- Enrutamiento.
- Difusión o multidifusión.
- Tamaño de los datos enviados.
- Calidad.
- Seguridad.
- Confiabilidad.
- Tiempos.

Una vez delimitadas las diferencias existentes entre los tipos de redes, es posible encontrar soluciones a este problema. En su momento existían dos, una de ellas sería colocar un dispositivo capaz de traducir la información obtenida en la interfaz de las redes conectadas, pero ello resultaría nada sencillo. La otra opción es agregar una capa más como interfaz de las redes, dicha capa tiene el **Protocolo IP** (Protocolo de Internet) el cual puede ser tratado como el protocolo de red casi universal pero existen otros como: **IPX**, **SNA** y **AppleTalk** aunque estos últimos no son muy utilizados. Los campos dirección origen y destino en la cabecera **IP** contienen cada uno una dirección internet global de 32 bits que, generalmente, consta de un identificador de red y un

identificador de computador [1.9], así es posible la comunicación directamente con el destinatario, aun cuando el paquete deba enviarse por los siguientes tipos de redes:

- **Clase A:** Pocas redes con muchas terminales.
- **Clase B:** Número medio de redes y cada una con un número medio de terminales.
- **Clase C:** Muchas redes y cada una con pocas terminales.

2.3.4 Capa de Transporte

Esta capa se ocupa de ampliar el servicio de entrega de la Capa de Red entre dos sistemas terminales a un servicio de entrega entre dos procesos de la Capa de Aplicación que se ejecutan en los sistemas terminales [2.6], en otras palabras, es proporcionar un servicio de transmisión de datos eficiente, confiable y económico a sus usuarios [1.10] utilizando los servicios que previamente brindó la Capa de Red a la Capa de Transporte.

Existen dos tipos de servicios que brinda la capa de transporte:

- Orientado a conexión
- Sin conexión

Ambas encargadas del establecimiento, mantenimiento y finalización de las conexiones al igual que la Capa de Red, sin embargo, la Capa de Transporte lo maneja a nivel del host del usuario a diferencia de la Capa de Red quien lo maneja a nivel de los enrutadores.

Una última función es controlar la velocidad de transmisión de las entidades (hardware y software de los procesos de la Capa de Transporte) encargadas de enviar los segmentos (datos) por la Capa de Transporte para evitar las congestiones.

Como la Capa de Transporte se sitúa a nivel de los host, estos entablan una comunicación lógica (Figura 2.13), la cual se traduce en una conexión directa aunque no sea realmente así, las terminales pueden estar separadas por una distancia considerablemente grande y conectados por otros equipos como switches y routers.

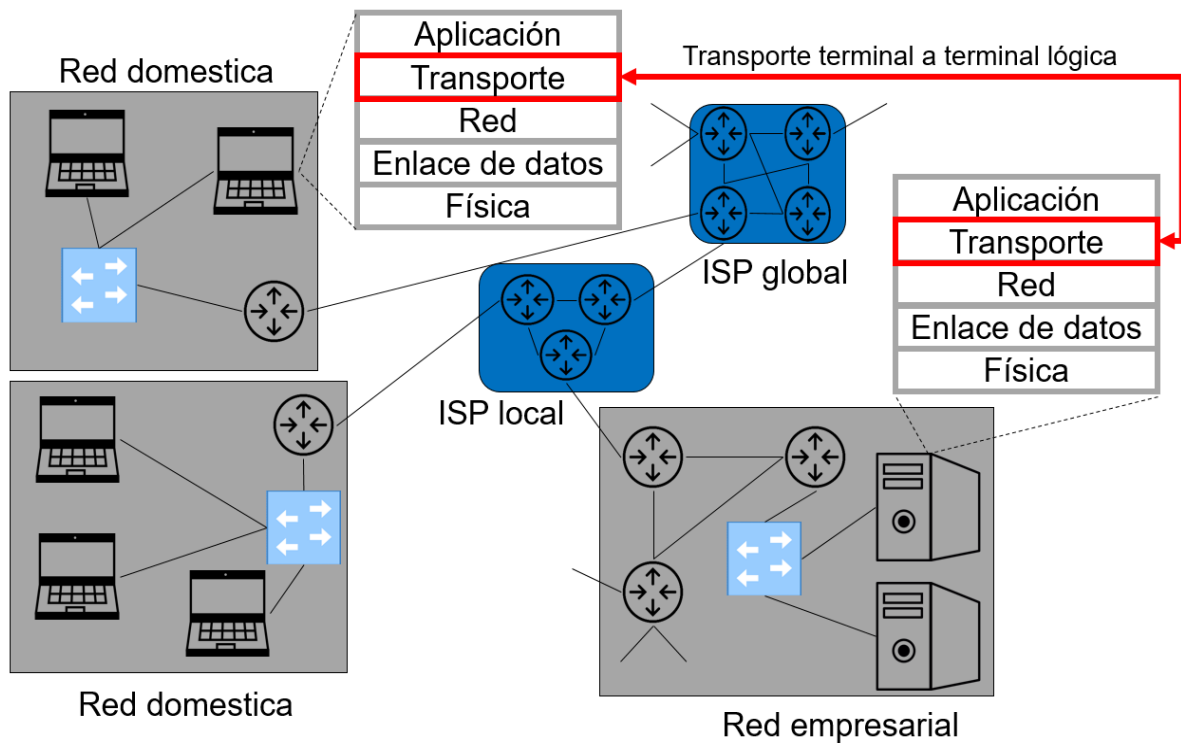


Figura 2.14 Comunicación lógica de la capa de transporte. [Elaboración propia]

A pesar de que la congestión se realiza en los enrutadores, en otras palabras a nivel de Capa de Red, también este problema se deriva por el tráfico producido que la Capa de Transporte envía por la red. Pero a diferencia de las soluciones que implementa la Capa de red, esta capa soluciona este problema regulando la velocidad con que los host envían los paquetes. El científico en computación Leonard Kleinrock propuso la métrica para la potencia, siendo la siguiente (Ecuación 2.1):

Ecuación de la potencia de envío de paquetes.

$$Potencia = \frac{Carga}{Retardo} \quad (\text{Ecuación 2.1})$$

- **Carga:** Carga ofrecida y son los paquetes enviados por segundo en la red.
- **Retardo:** Retardo de propagación a través de la red en segundos.

En cuanto el retardo sea mínimo la potencia logra su máximo y esto significa que las entidades de transporte logran la mejor eficiencia, aprovechando de la mejor manera el ancho de banda de la red.

El protocolo más conocido para esta capa es **TCP**. Dicho protocolo fue creado específicamente para el flujo efectivo entre redes que difieren en topologías, anchos de banda, retardos, tamaño de los paquetes entre otros parámetros. Su función es enviar los paquetes de datos con una velocidad máxima sin causar una congestión.

TCP se diseñó para adaptarse de manera dinámica a las propiedades de la interred y sobreponerse a muchos tipos de fallas [1.10].

2.3.5 Capa de Sesión

Esta capa permite a los host iniciar sesiones entre ellos, dichas sesiones ofrecen los servicios de:

- Control de transmisión.
- Manejo de operaciones simultáneas (**tokens**).
- Reanudación de transmisiones en caso de interrupciones.

La Capa de Sesión permite delimitar y sincronizar el intercambio de datos, incluyendo los medios para crear un punto de restauración y un esquema de recuperación [2.6].

2.3.6 Capa de Presentación

La Capa de Presentación se enfoca en la sintaxis de la información transmitida e interpretar el significado de los datos de intercambio.

Estos servicios incluyen la compresión y el cifrado de los datos, así como la descripción, haciendo que la aplicación no se preocupe por el formato de los datos representan y almacenan, datos que difieren entre máquinas [2.6].

Esta capa a diferencia de las demás que se centralizan primordialmente en mover los datos del origen al destino, da formato a los datos que son enviados y los presenta en una forma compatible para que lleguen al destinatario, comprimir los datos al ser enviados y descomprimirlos a su llegada y finalmente la encriptación y descifrado durante la conexión con el envío de datos.

2.3.7 Capa de Aplicación

Un proceso de aplicación es aquella parte de un sistema final que procesa información en un entorno de red **OSI** y que se invoca por el usuario [2.5].

Esta capa es la interfaz entre el usuario y los programas lógicos que se encarga de realizar una función en particular llamados elementos de servicios de aplicaciones genéricas y especializadas.

2.3.7.1 Generales:

- **ACSE** (Elemento de servicio de control de asociación).

Funciona como la conexión a nivel de aplicación, conocidas como asociaciones permitiendo el intercambio de datos.

- **RTSE** (Elemento de servicio de transferencia confiable).

Producido para la transferencia masiva de información pero actualmente en desuso.

- **ROSE** (Elemento de servicio de operaciones remotas).

Operaciones Cliente-Servidor.

2.3.7.2 Especializadas:

- **Web y HTTP.**

En el año de 1994 se puso en circulación la mayor aplicación para la red de Internet; **WWW** (*World Wide Web*), disponible para toda persona en el mundo, ya que antes la red de Internet solo era campo para investigadores y personal académico. Los sitios web han crecido en demasía y ciertas empresas tomaron popularidad en sus campos de función como: una librería (Amazon en 1994), subastas en línea (eBay en 1995), búsqueda (Google en 1998) y redes sociales (Facebook, 2004) [1.10] son solo algunos ejemplos en un inicio, dado que al transcurrir el tiempo las aplicaciones móviles empezaron a surgir como una galería de fotos en línea (Instagram en 2010) o mapas en la web (Google Maps en 2005).

Para poder introducirse entre el sinfín de páginas web existentes, se utiliza lo que se conoce como navegador web tales como: Chrome, Firefox, Internet Explorer, entre otros, y su protocolo para poder navegar en la web se le llama **HTTP** (Protocolo de Transferencia de HiperTexto, *HyperText Transfer Protocol*), el cual sirve para poder pasar de una página web a otra, se implementa mediante dos programas: un programa cliente y un programa servidor. Ambos programas, que se ejecutan en sistemas terminales diferentes, se comunican entre sí intercambiando mensajes **HTTP** [2.6] (Figura 2.14)⁴⁵.



Figura 2.15 Protocolo HTTP.

⁴⁵ [2.6] "Comportamiento solicitud-respuesta de HTTP". Página 82.

- Correo electrónico y mensajes instantáneos:

Ha existido desde los comienzos del Internet en 1990 e igual que el correo postal sirve como un medio de comunicación pero mucho más rápido, eficiente, fácil y barato. A través de los años ha sufrido infinidad de cambios por lo que en la actualidad es posible enviar archivos como fotografías, audio, videos, hipervínculos, entre otras cosas.

En este sistema de correo electrónico existen 3 integrantes que establecen esta comunicación (Figura 2.15): usuarios (host que envía y host que recibe), servidores de correo (infraestructura de correo electrónico) y el **SMTP** (Protocolo Simple de Transferencia de Correo, *Simple Mail Transfer Protocol*).

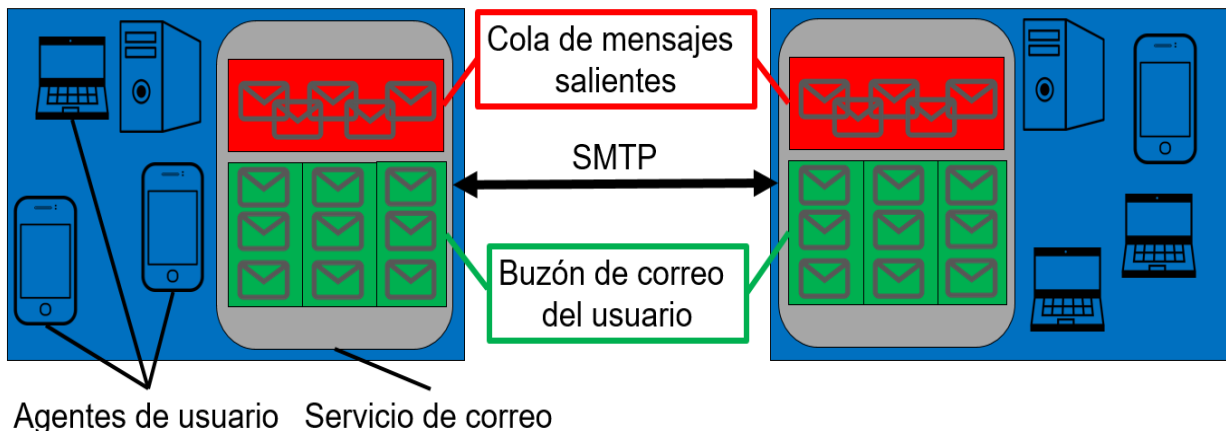


Figura 2.16 Arquitectura del servicio de correo electrónico. [Elaboración propia]

El protocolo **SMTP** es el corazón del correo electrónico por Internet, transfiere mensajes desde los servidores de correo de los emisores a los servidores de correo de los destinatarios [2.6]. Y en la actualidad sigue funcionando como por ejemplo: Gmail, Outlook, Yahoo, entre otros.

- **DNS** (Servicio de Resolución de Nombres, *Domain Name Service*):

Fue diseñado en 1983 y se encarga de separar la dirección **IP** y su nombre de algún sitio web para facilitar su búsqueda, de esta manera el navegador puede encontrar el sitio buscándola mediante su nombre y no su dirección **IP**. Esto facilita el motor de búsqueda ya que de no existir el usuario deberá recordar la dirección **IP** de la página web y descartar el problema si la página web cambia de dirección **IP**. La función básica del **DNS** es la invención de un esquema jerárquico de nombres basado en dominios y un sistema de base de datos distribuido para implementar este esquema de nombres. El **DNS** se usa principalmente para asociar los nombres de host con las direcciones **IP** [1.10].

- **Audio y video:**

Una de las principales prácticas para la red de Internet es el audio y video, por ejemplo proveer la voz por **IP** y el video por pedido como películas, videollamadas o bien videos pregrabados. Para ello fue necesario tener dispositivos con mejores

características y un ancho de banda en la red de internet prioritario para ello, aunque las llamadas telefónicas no requieren de estas características pero sí el tráfico de video. Las compañías de teléfono buscaron la forma más económica de transportar las llamadas mediante el uso de equipo de redes IP, los datos de voz se transportan a través de redes de Internet, a lo cual se le conoce como telefonía de Internet o **VoIP** (voz sobre IP) [1.10].

Existe un problema con la distribución de video por el protocolo **HTTP**, y es que todos los usuarios que soliciten un servicio de video igual tendrán la misma calidad (audio y video). Para ello se diseñó una nueva tecnología denominada como **DASH** (Flujos Dinámicos Adaptativos Sobre HTTP) en el cual el vídeo se codifica en varias versiones diferentes, teniendo cada versión una tasa de bits distinta, por tanto, un nivel de calidad diferente [2.6].

Algunos ejemplos de este servicio son: Skype (2003) o WhatsApp (2009) para llamada o video llamadas y Netflix (1997) o Youtube (2005) para video.

2.4 Protocolos TCP/IP (Transmission Control Protocol e Internet Protocol)

Estos protocolos son un grupo de reglas para los datos (Figura 2.16), que por medio de algunos procedimientos permita a las máquinas y los programas intercambiar recursos, es decir, cada host implicado en una comunicación con otros host debe seguir las reglas para que los datos enviados puedan ser interpretados por los host receptores. El **Protocolo IP** se encuentra a nivel de la Capa de Red (Capa de Interred en el modelo **TCP/IP**) y **TCP** se encuentra a nivel de la Capa Transporte.

El conjunto de protocolos **TCP/IP** puede interpretarse en términos de capas o niveles (Figura 2.17):

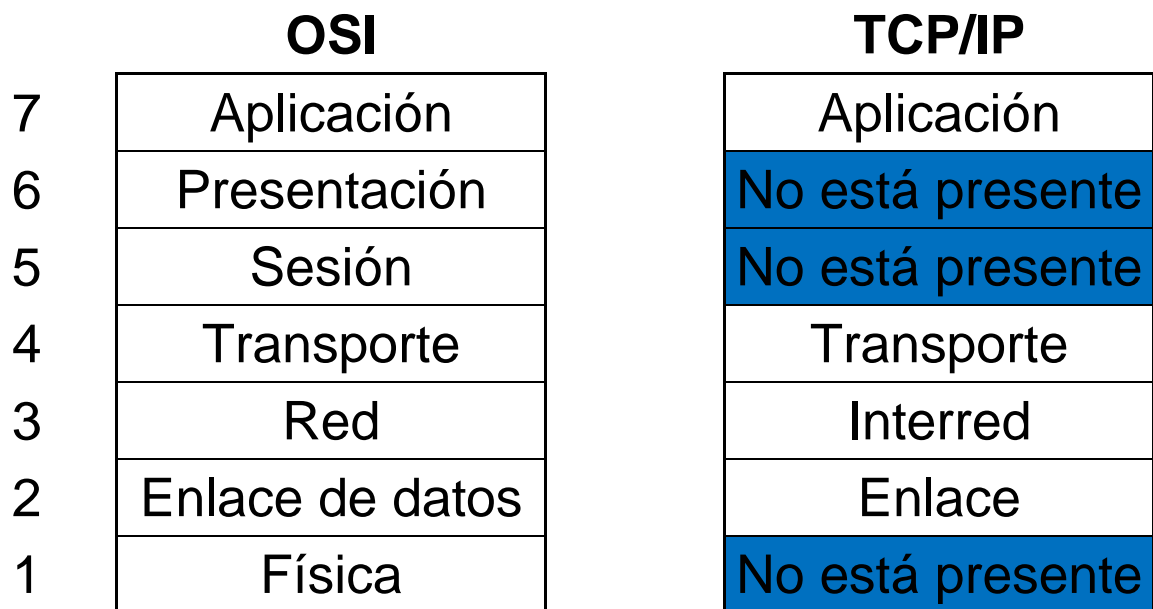


Figura 2.17 Capas del modelo TCP/IP. [Elaboración propia]

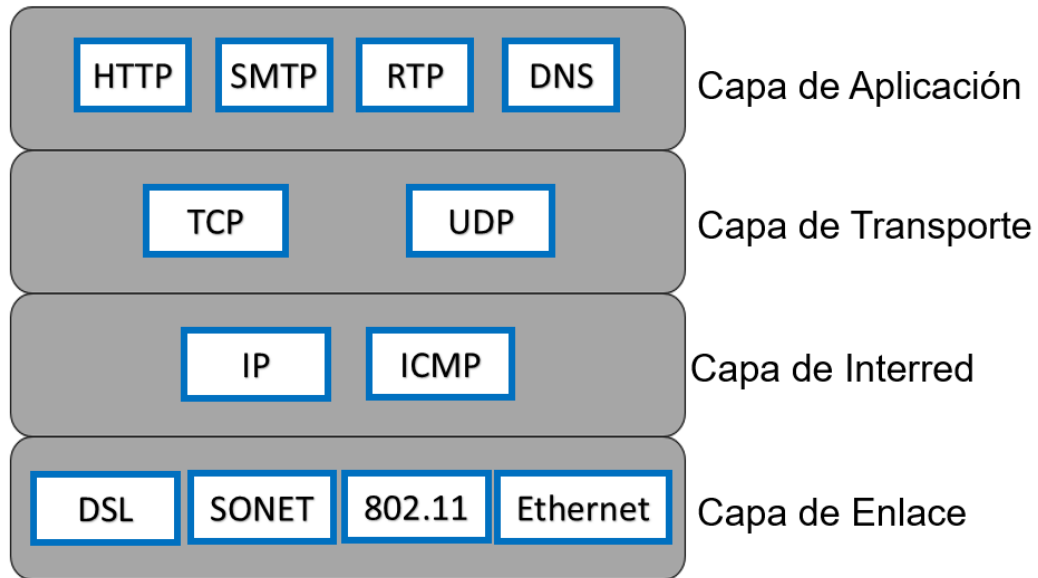


Figura 2.18 Protocolos por capa. [Elaboración propia]

2.4.1 La Capa de Enlace

Siendo la capa más baja de este modelo, ésta diseñada como una interfaz entre el host y los enlaces de transmisión, describiendo que tipo de conexiones se deben realizar para cumplir con las necesidades de la Capa de Interred.

2.4.2 La Capa de Interred

La Capa de Interred es la equivalente a la Capa de Red del modelo **OSI**. Su trabajo es permitir que los host inyecten paquetes en cualquier red y que viajen de manera independiente hacia el destino [1.10], es decir, esta capa se encarga de entregar los paquetes IP a su destinatario siendo el ruteo de paquetes el principal aspecto de interés. Es aquí donde aparece la mayor importancia del **Protocolo IP** además de otro llamado **ICMP** (Protocolo de Mensajes de Control de Internet, *Internet Control Message Protocol*).

El trabajo del **Protocolo IP** es proporcionar una guía por el medio de transmisión para transportar paquetes de la fuente al destino, sin importar si estas máquinas están en la misma red o si hay otras redes entre ellas (Figura 2.18)⁴⁶.

⁴⁶ [1.10] "Internet es una colección interconectada de muchas redes.". Página 375.

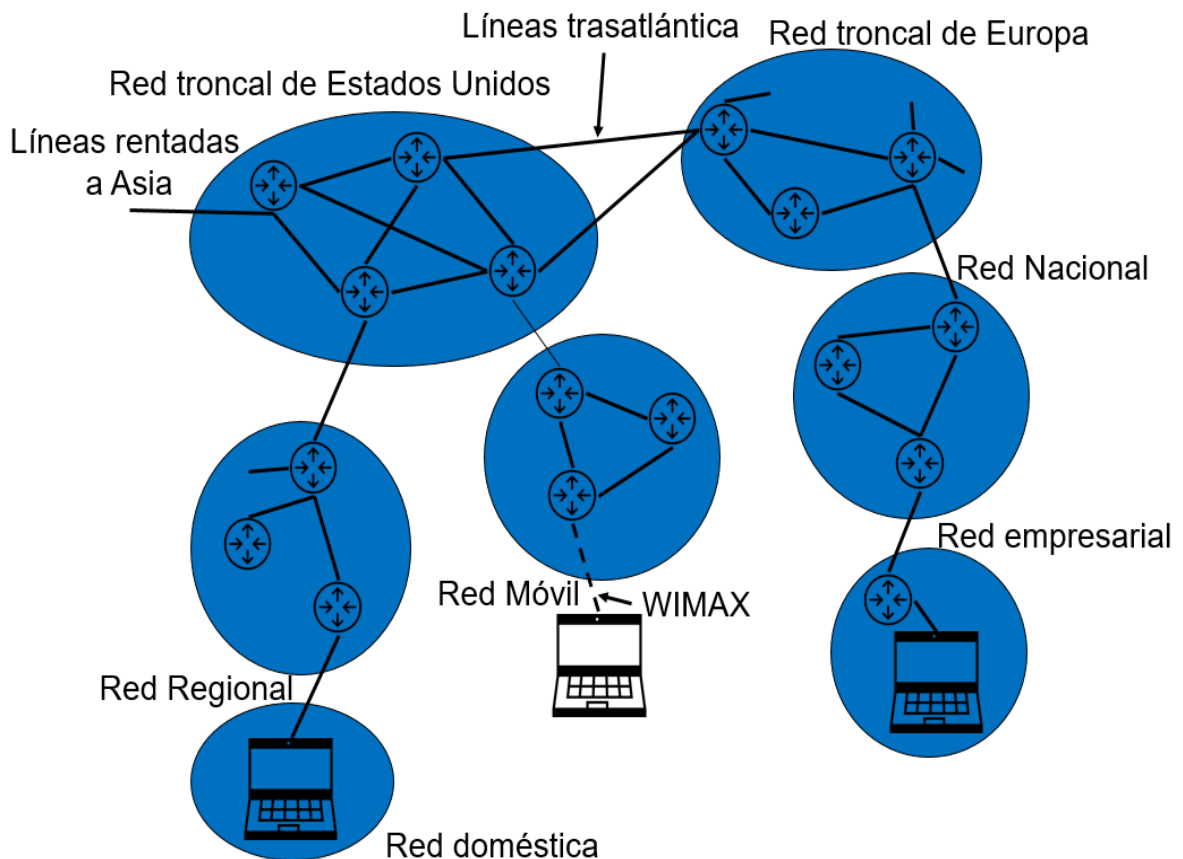


Figura 2.19 Protocolo IP.

Los datos son enviados a través de los enrutadores **IP** los cuales van guiando a los mensajes desde el host que envía hasta el que recibe.

2.4.3 La Capa de Transporte

En esta capa se definen dos protocolos de transporte para ambos extremos:

- **TCP**: Es el responsable de supervisar el estado de lo que se ha enviado y retransmite todo lo que no pudo ser recibido en otro extremo. Si un mensaje es grande para enviarse en un solo datagrama, **TCP** lo segmenta en varios datagramas y se asegura que lleguen correctamente al extremo receptor. Por otro lado, el **Protocolo IP** es el responsable del enrutamiento de los datagramas individuales [2.1]. En el momento que **TCP** segmenta el mensaje estos se vuelven independientes en su envío a través de la red, por lo que hay ocasiones en que el primer segmento enviado sea el último en llegar al receptor (Figura 2.19).

Provee los siguientes servicios:

- **Servicio orientado a conexión**: Causa el intercambio de datos de control entre el cliente y el servidor antes del envío de los datos de interés

a través de la Capa de Aplicación, previniendo a ambos lados que se preparen para el flujo de datos, establecen una conexión Full-Duplex y posteriormente **TCP** termina con la conexión.

- **Servicio de transferencia de datos fiable:** Se encarga de segmentar el mensaje al enviarlo por la red (de ser necesario), volver a unirlos en el receptor y ordenarlos correctamente.
- **Servicio de congestión:** Por medio del campo Ventana se encarga de no saturar de datos al receptor, el cual por medio del campo Ventana se encarga de hacerle saber al transmisor la cantidad de bits que puede soportar.

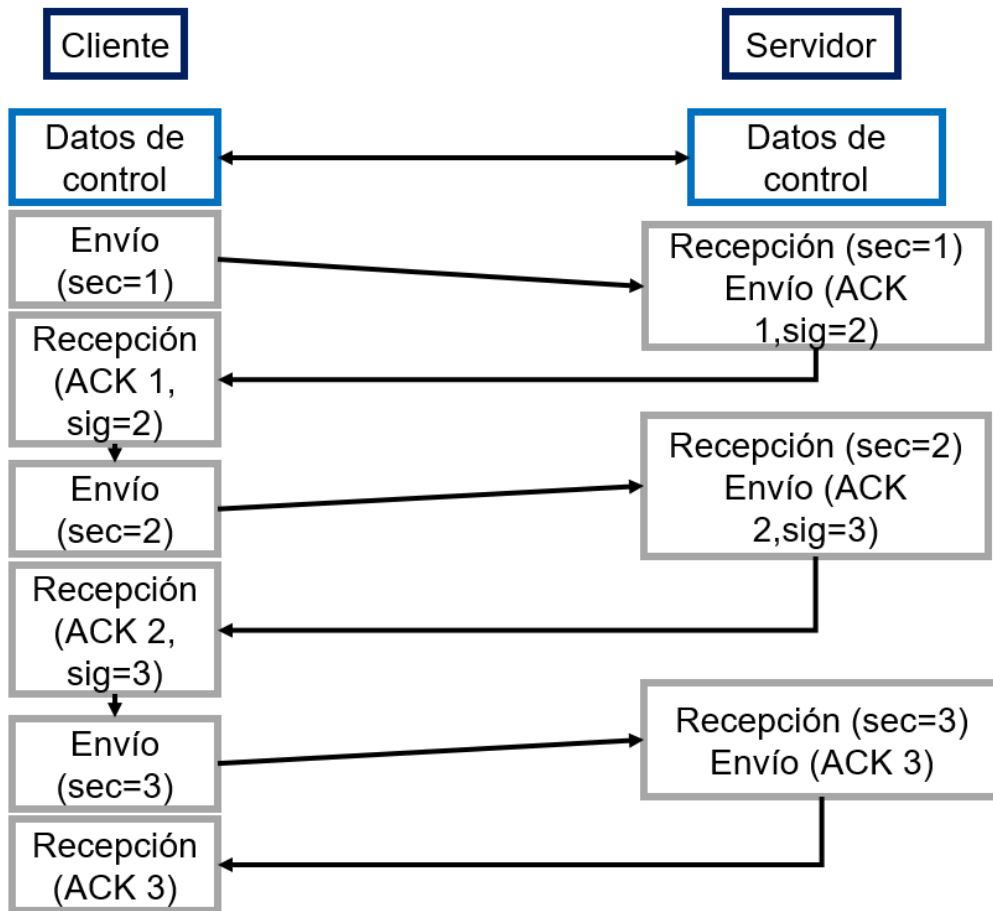


Figura 2.20 Protocolo TCP. Ventana = 1 paquete. [Elaboración propia]

- **UDP** (Protocolo de Datagrama de Usuario, *User Datagram Protocol*): Es un protocolo sin conexión y no confiable dado que no garantiza la entrega los datos, ni su orden correcto y tampoco a su protección contra duplicaciones (Figura 2.20).

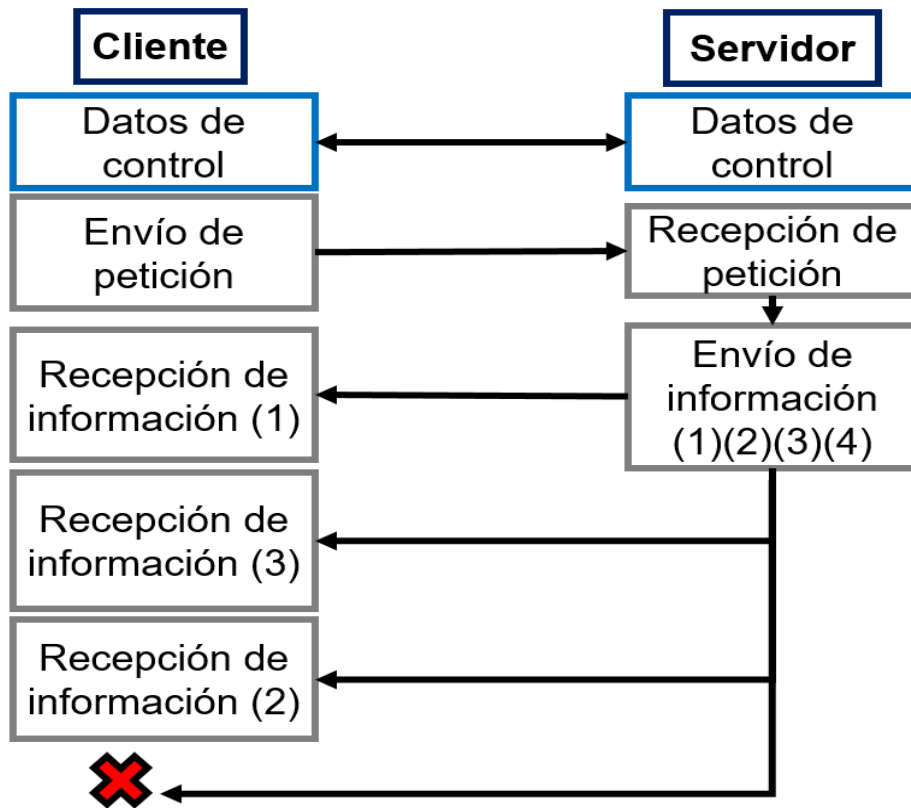


Figura 2.21 Protocolo UDP. [Elaboración propia]

2.4.4 La Capa de Aplicación

Esta capa tiene todos los protocolos de nivel más alto. Entre los que están:

- **TELNET** (Terminal virtual)
- **FTP** (Transferencia de archivos)
- **SMTP** (Correo electrónico)
- **DNS** (Sistema de nombres de dominio)
- **RTP** (Protocolo para transmitir medios en tiempo real)

Desempeña el mismo trabajo que realiza la Capa de Aplicación del modelo de referencia **OSI**.

2.5 Voz sobre IP (VoIP).

La transmisión de paquetes de voz en una red es similar a la de datos, por lo que la red trabaja de la misma manera particularmente con el **Protocolo IP** dando como resultado: **VoIP** (voz sobre IP, *Voice over Internet Protocol*). **VoIP** como el **DSP** (Procesador Digital de Señales) el cual se encarga de la compresión y la descompresión de los paquetes de datos, ambos son el pilar esencial de esta tecnología.

VoIP está diseñado para la transferencia de voz entre el host transmisor y el receptor en paquetes **IP** por medio de Internet, es decir, los paquetes de voz son previamente convertidos en datos para poder ser transmitidos por Internet por medio de **Protocolo IP**. Para su eficiencia debe cumplir los siguientes objetivos:

- Compatibilidad entre diferentes equipos.
- Supresión de silencios.
- Codificación de los paquetes de voz.
- Direccionamiento.

Para la digitalización de datos analógicos son necesarios codificadores y decodificadores (**Codec's**) y precisamente para los datos de voz se utiliza el **ITU-T G.711** también conocido como **PCM** (Modulación por Codificación de Pulsos, *Pulse Code Modulation*) en sus 2 variantes Ley A y Ley μ [1.8].

La transferencia establecida en **VoIP** consta de tres aspectos importantes (Figura 2.21): codificación o decodificación en donde la señal analógica es digitalizada en señales **PCM**, algoritmo de compresión o descompresión de la señal **PCM** y finalmente la segmentación o unión de la señal **PCM**.

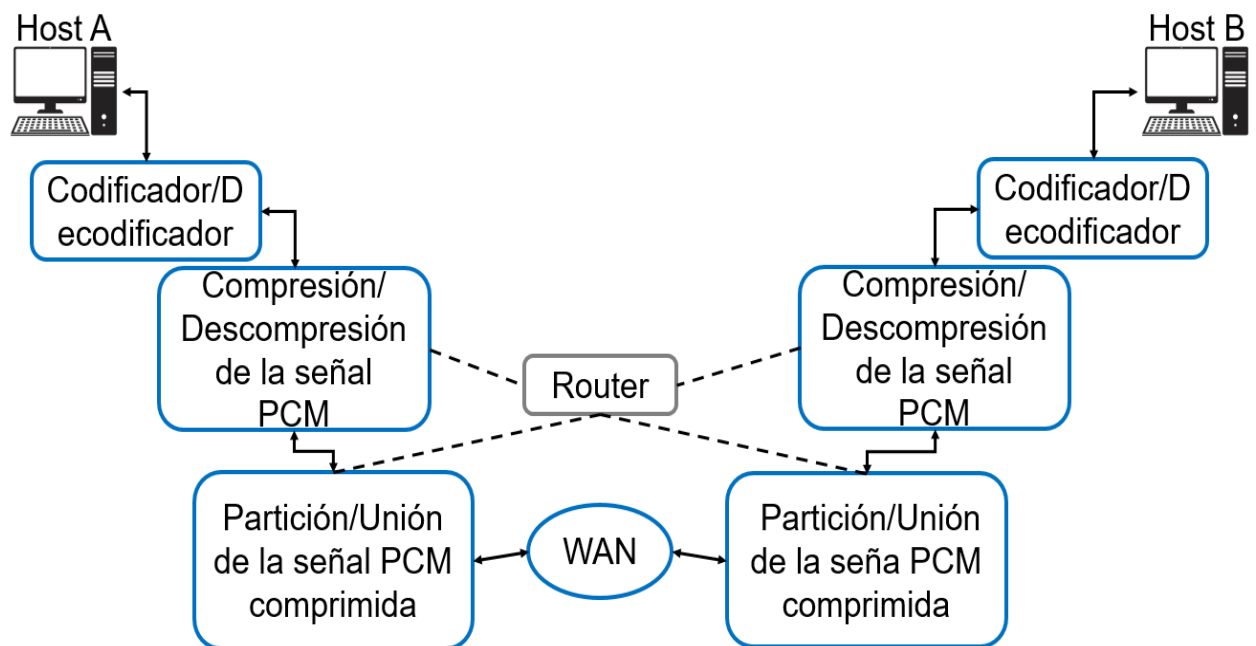


Figura 2.22 Flujo de paquetes de voz. [Elaboración propia]

El router o el Gateway son los encargados de los procesos realizados durante el flujo de paquetes de voz (Figura 2.22) pero también se puede implementar una **PBX** (Conmutador Telefónico Privado, Private Branch Exchange) para realizar los procesos de codificación.

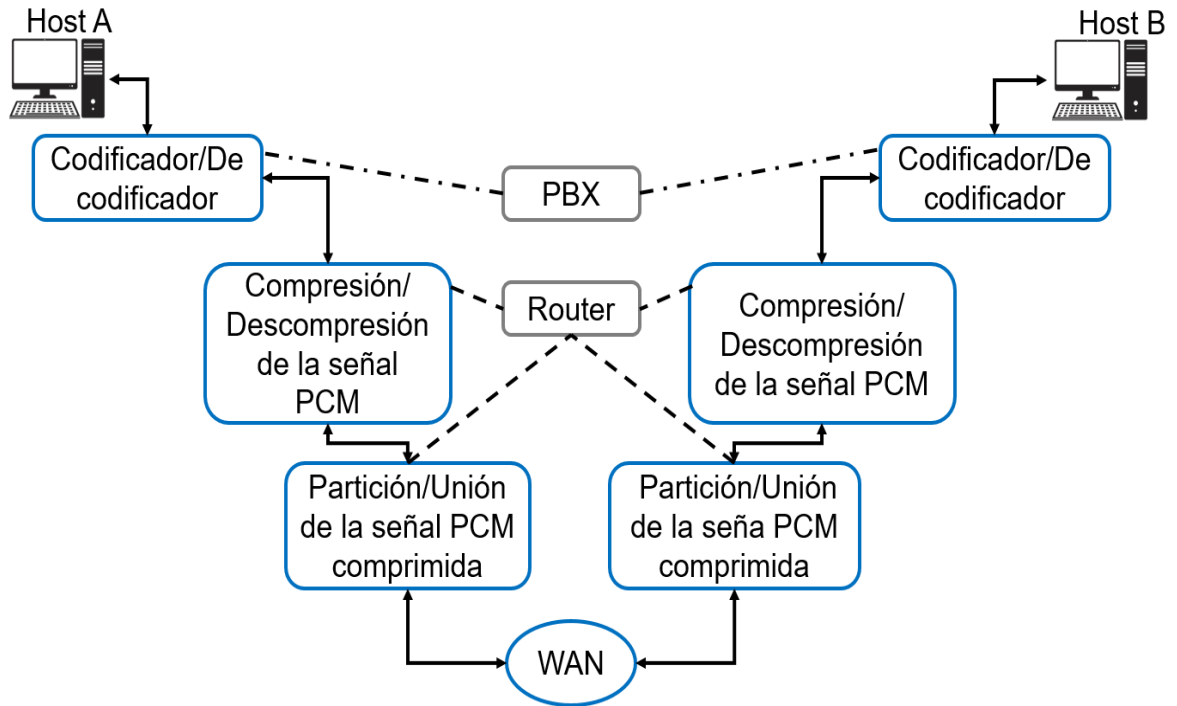


Figura 2.23 Flujo de paquetes de voz con una PBX. [Elaboración propia]

La diferencia entre **VoIP** y **ToIP** (Telefonía sobre IP), es que la primera es la tecnología propiamente empleada y la segunda es un servicio, en este caso de telefonía sobre IP.

2.5.1 Componentes de la red VoIP.

Los principales elementos que componen una red **VoIP** son los 3 siguientes:

2.5.1.1 Terminales IP.

- Telfonos fijos con **ATA** (adaptadores analógicos IP, *Analogic Telephony Adapter*).
- Inalámbricos.
- Móviles **IP**.
- Consolas.
- Softphones o programas de telefonía para computadoras o teléfonos móviles.
- IPphones.

2.5.1.2 Gatekeepers IP o PBX.

Centrales telefónicas para gestionar llamadas internas siendo un elemento no necesario pero muy útil para el control. Su función es conectar internamente (extensiones) las terminales **IP** de una red **LAN** y posiblemente conectarse con una red **PSTN** (Figura 2.23).

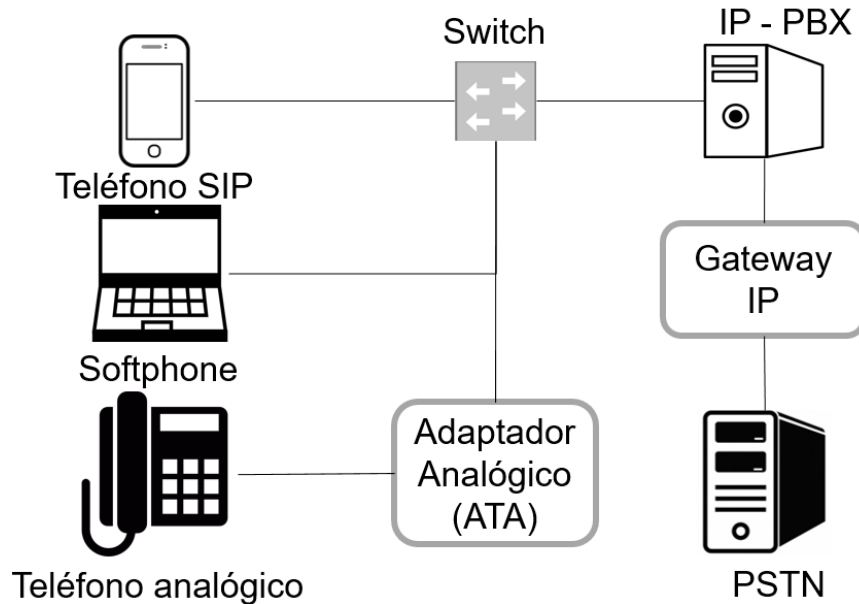


Figura 2.24 Función de una PBX. [Elaboración propia]

RBX realiza las siguientes tareas:

- Administración de ancho de banda.
- Traducción de direcciones.
- Asignación de extensiones.
- Administración de terminales y redes.
- Control de llamadas.
- Mensajería.
- Registro de llamadas y mensajes.
- Agenda.
- Historiales.
- Buzón de voz.
- Grabación y personalización de audios.
- Conferencias.
- Videollamdas.
- Desvío de llamadas.

2.5.1.3 Gateway IP.

Dispositivo que se encarga de la conexión entre 2 redes completamente diferentes, en este caso se encarga del enlace entre la red VoIP y la PSTN (Figura 2.23).

2.5.2 Protocolos VoIP.

Los protocolos VoIP existen de la misma manera para que en cualquier transferencia de datos en donde se realice la conexión, se mantenga y finalmente finalice la transferencia al igual que el modelo de referencia **OSI** o **TCP/IP**. Algunos de esos protocolos son:

2.5.2.1 H.323

Es el primer protocolo desarrollado para el envío de datos multimedia, en otras palabras, su función es proveer comunicaciones convergentes de audio, video y datos sobre redes **IP**. Publicada por la **ITU-T** como Sistemas y equipos terminales de telefonía visual de banda angosta, para redes de área local que proveen calidad de servicio no garantizada [2.7] trabajando con paquetes **UDP**, haciendo referencia a las terminales **IP**, equipos y servicios estableciendo la señalización y control desde el año de su creación en 1996.

De la misma manera que se nombraron los componentes para una red **VoIP**, los componentes o dispositivos de este protocolo son los mismos con la peculiaridad que son específicamente para el protocolo **H.323** (Figura 2.24)⁴⁷:

- **Terminales H.323**: Son los extremos finales de la red compatibles con **H.323**.
- **Gatekeeper**: Encargado de administrar los recursos utilizando el protocolo **H.225**, direccionando los paquetes multimedia y cantidad de flujo dentro de la red **H.323**.
- **Gateway**: Encargado de conexión entre la red **H.323** con la red **PSTN**.

Usa los siguientes protocolos de la **UTI-T** y de **IETF** (Grupo de Trabajo de Ingeniería de Internet, *Internet Engineering Task Force*).

- **DNS**: Traduce un identificador basado en texto (nombre) a una identificación numérica (dirección IP).
- **G.711**: Codificación digital de audio.
- **G.722**: Codificación de audio de 7 kHz dentro de 64 kbit/s.
- **G.723**: Codificación de voz de 24 y 40 kbit/s.
- **G.728**: Codificación de señales vocales a 16 kbit/s utilizando predicción lineal con excitación por código de bajo retardo.
- **G.729**: Codificación de la voz a 8 kbit/s mediante predicción lineal con excitación por código algebraico de estructura conjugada.
- **H.225**: Protocolo de señalización y se encarga del control de llamadas.
- **H.235**: Se encarga de la seguridad y el cifrado.

⁴⁷ [1.8] "Protocolos que soportan un sistema H.323". Página 55.

- **H.245**: Protocolo de señalización y se encarga de la especificación de los paquetes multimedia.
- **H.261**: Códec de vídeo para servicios audiovisuales de 64 kbit/s.
- **H.263**: Códec de vídeo similar al **H.261** mejorando la calidad de imagen.
- **H.320**: Ejecución de los paquetes multimedia.
- **H.450**: Descripción de los servicios suplementarios.
- **H.460**: Definición de las extensiones opcionales.
- **Q.931**: Protocolo de señalización e indica el inicio de llamada.
- **RAS**: Registro, Administración y Estado. Protocolo de direccionamiento y permite que una terminal **H.323** localice otra terminal del mismo tipo.
- **RTCP** (*Real Time Transport Control Protocol*): Protocolo de transmisión que se encarga de las congestiones existentes en la red.
- **RTP** (*Real Time Protocol*): Protocolo para transmisión de voz en tiempo real.
- **T.120**: Protocolo de datos para conferencias o multimedia.
- **UDP**: Protocolo de transporte y se utiliza por la eficiencia en el ancho de banda.

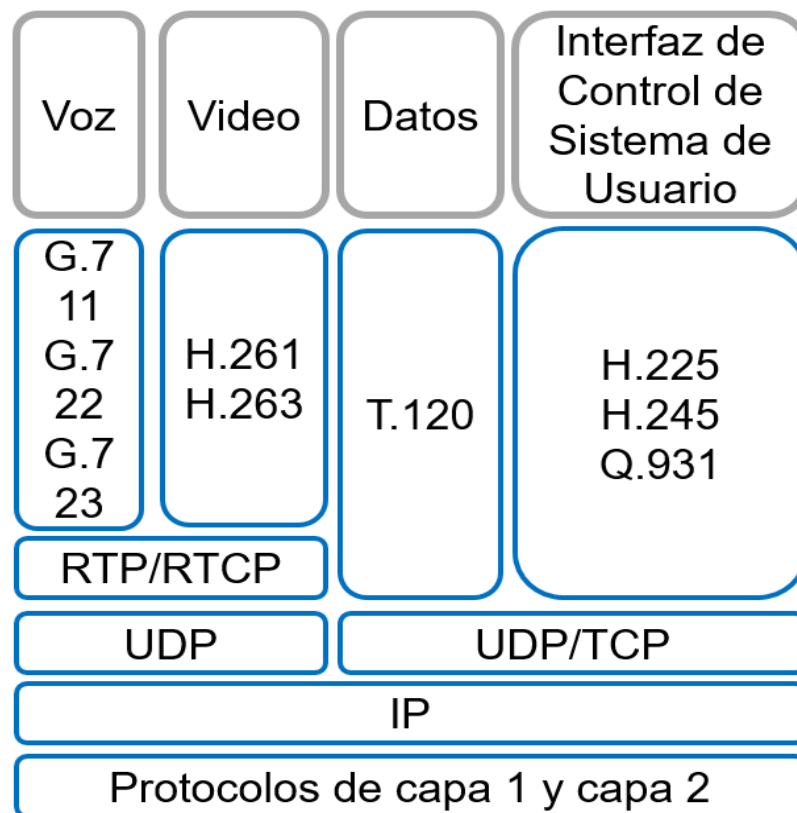


Figura 2.25 Protocolos en una red H.323

Al realizar una llamada utilizando esta arquitectura se puede observar mediante la siguiente figura (Figura 2.25)⁴⁸ la comunicación entre las terminales con el Gatekeeper:

⁴⁸ [1.8] "Llamada H.323". Página 58.

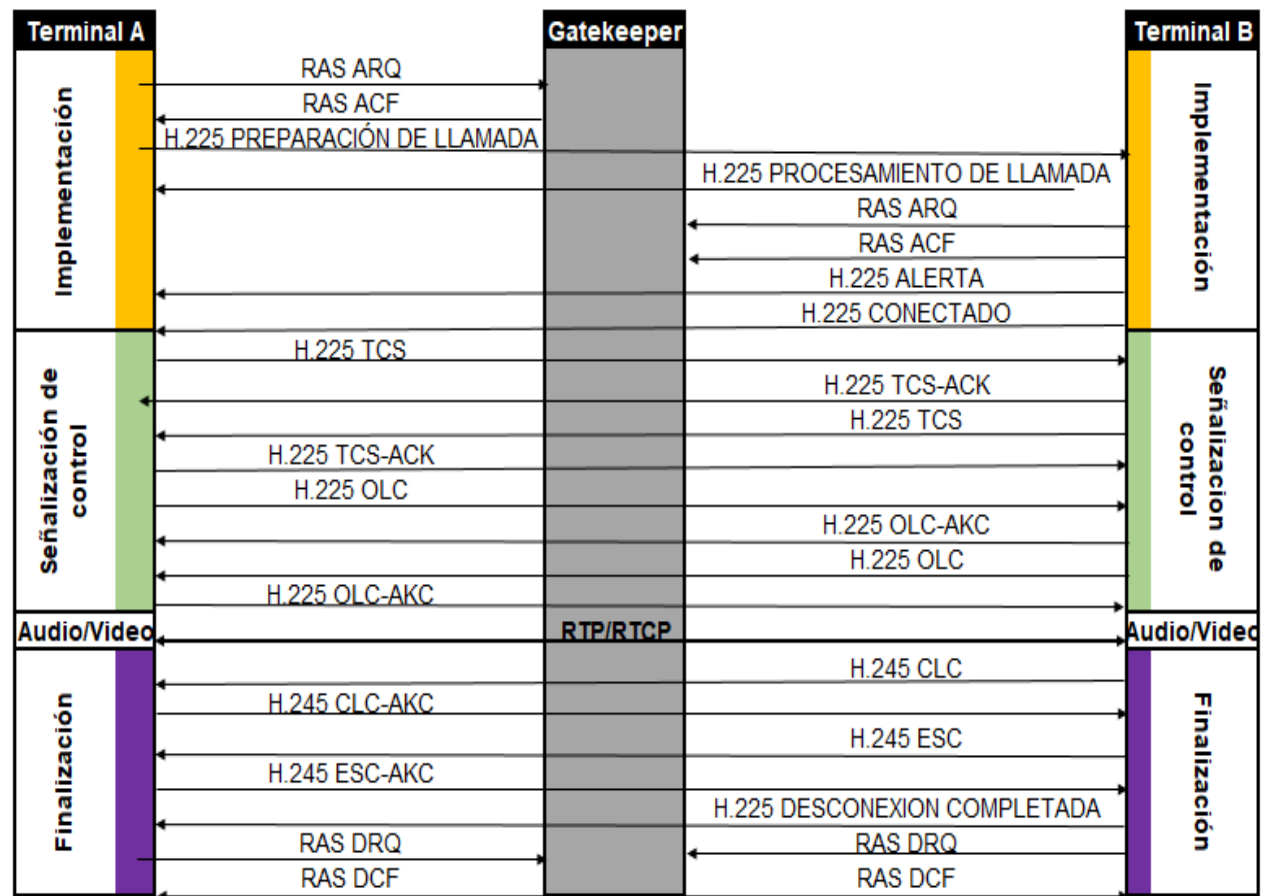


Figura 2.26 Proceso de llamada H.323

1. Implementación: Se realizan los procesos para iniciar la conexión entre dos terminales **H.323**.
 - La terminal que realiza la llamada (Terminal A) debe iniciar con el registro de la terminal **H.323** ante el Gatekeeper mediante el protocolo **RAS** por medio de un mensaje **ARQ** (solicitud de admisión) y por parte del Gatekeeper un mensaje **ACF** (confirmación de admisión).
 - La Terminal A debe realizar el envío de un mensaje “PREPARACIÓN DE LLAMADA” a la otra terminal (Terminal B) por medio del protocolo **H.323** indicando la dirección **IP** puerto y nombre de la terminal que realiza la llamada. En el otro extremo la Terminal B debe responder con un “PROCESAMIENTO DE LLAMADA”.
 - De la misma forma que la Terminal A, la Terminal B debe registrarse ante el Gatekeeper.
 - La Terminal B avisa a la Terminal A para la generación de tono con un mensaje de “ALERTA” y posteriormente un mensaje de “CONECTADO” para iniciar la llamada.
2. Señalización de control: Se realiza mediante el protocolo **H.245** en donde ambas terminales determinan cuál de ellas será el “MAESTRO” y el “ESCLAVO”, capacidades de ambas terminales y la codificación de los datos

multimedia que se enviarán, así, abren el canal de comunicación por la dirección **IP** y el puerto.

- Comunicación **TCS** (*Terminal Capability Set*) en donde se establecen las capacidades que soportan ambas terminales para realizar la llamada y se ejecuta mediante pregunta y respuesta entre ambas terminales.
 - Comunicación **OLC** (*Open Logical Channel*) en donde se abre el canal lógico que contiene la información para la transmisión y recepción de los datos (voz/video) y el tipo de codificación.
3. Comunicación entre las terminales: Da comienzo la transmisión de voz/video.
 - Se inicia la comunicación mediante el protocolo **RTP/RTCP**.
 4. Finalización: Desconexión de las terminales.
 - Se realiza mediante mensajes **CLC** (*Close Logical Channel*) de cualquiera de las terminales para cerrar el canal lógico de transmisión y **ESC** (*End Session Comand*) para cerrar las sesiones.
 - Mediante el protocolo **H.245** “DESCONEXIÓN COMPLETADA”.
 - Y finalmente ambas terminales liberan los registros en el Gatekeeper mediante el protocolo **RAS** por medio del mensaje **DRQ** (solicitud de desconexión) y **DCF** (confirmación de desconexión).

2.5.2.2 SIP (Session Initiation Protocol)

El protocolo **SIP** o Protocolo de Inicio de Sesión, es un estándar para la inicialización, modificación y finalización de sesiones interactivas de usuario donde intervienen elementos multimedia. Se emplea para la señalización **VoIP** y permite determinar la ubicación de los usuarios (posibilitando la movilidad) [2.4]. Desarrollado por el **IETF** (Grupo de Trabajo de Ingeniería de Internet, *Internet Engineering Task Force*), utiliza como protocolos de transporte **TCP**, **UDP** y **SCTP** (Stream Control Transmission Protocol para la seguridad en la pérdida de paquetes para TCP y latencia para UDP), además del protocolo **SDP** (Protocolo de Descripción de Sesión, *Session Description Protocol*) quien se encarga de la descripción de sesiones al momento de iniciarlos (no los negocia). También el protocolo **SIP** utiliza el puerto 5060 y es muy parecido a **HTTP** trabajando a nivel de aplicación (Capa 7 del modelo OSI) y los siguientes protocolos (Figura 2.26).

Aplicaciones de Audio y Video		Aplicaciones de Control de la Sesión	
Codificación/Decodificación		SDP	
RTP/RTCP		SIP	
UDP		TCP	SCTP
IP			

Figura 2.27 Protocolos usados en la red SIP. [Elaboración propia]

Brinda una variedad de servicios como:

1. Localización de terminales.
2. Disponibilidad de terminales.
3. Traducción de nombres en direcciones **IP**.
4. Gestionar las fases de las sesiones.
5. Gestiona el número de terminales en la sesión y contabilizarlas.

2.5.1.2.1 Mensajes SIP

El protocolo **SIP** ocupa mensajería de señalización para las solicitudes y respuestas que se realizan en un modelo Cliente-Servidor con una notación **ABNF** (*Augmented Backus-Naur Form*), el cual permite caracteres del código **ASCII** (Código Estándar Estadounidense para el Intercambio de Información, *American Standard Code for Information Interchange*). Las solicitudes están basadas en el protocolo **SDP** y las respuestas en el protocolo **HTTP**.

- **Métodos / Solicitudes / Peticiones.**

Las peticiones más comunes son:

- **INVITE**: Invitación para inicio de una sesión.
- **ACK**: Confirmación de llegada el mensaje **INVITE**.
- **BYE**: Terminar sesión.
- **CANCEL**: Cancelación sesión solicitada.
- **OPTIONS**: Consulta los detalladles con los servidores.
- **REGISTER**: Registro de la terminal ante en el servidor.
- **PRACK**: Confirmación provisional.
- **PUBLISH**: Publicación de Evento en el servidor.
- **NOTIFY**: Notificación de Evento.
- **SUBSCRIBE**: Suscribe a la terminal a un **NOTIFY**.
- **UPDATE**: Modifica el estado de la sesión en curso.
- **MESSAGE**: Mensaje instantáneo.
- **REFER**: Petición a la terminal de una solicitud **SIP**.
- **INFO**: Envío de información de la sesión en curso.

- **Respuestas / Código de estado.**

Existen 6 categorías posibles y se muestran en la siguiente tabla con las respuestas más comunes (Tabla 2.4):

Respuestas SIP		
Respuestas Informativas		
100	Intentando	100 - 199 (1XX)
180	Sonando	
182	En cola	
Respuestas Exitosas		
200	Ok, autorizado	200 - 299 (2XX)
202	Aceptado	
Respuestas de Redireccionamiento		
301	Movido permanentemente	300 - 399 (3XX)
302	Movido temporalmente	
305	Usar Proxy	
Respuestas a Errores del Cliente		
401	No autorizado	400 - 499 (4XX)
403	Prohibido	
404	No encuentro (usuario)	
405	No permitido	
407	Requiere autenticación	
415	Tipo de medio no admitido	
484	Direcciones no completas	
Respuestas a Errores del Servidor		
500	Error interno	500 - 599 (5XX)
503	Servicio no disponible	
Respuestas a Errores Globales		
600	Ocupados	600 - 699 (6XX)
604	No existe en ningún lugar	

Tabla 2.4. Respuestas SIP. [Elaboración propia]

2.5.1.2.2 Entidades SIP

La red **SIP** tiene entidades de red, mejor conocidos como Agentes, los cuales pueden ser Clientes **SIP** o Servidores **SIP**, se direccionan por el esquema **URI** (Identificador de Recursos Uniforme, *Uniform Resource Locator*) que sirve para el servicio de localización de entidades **SIP** como Cliente-Servidor.

1. Terminales SIP y Agentes de Usuarios (User Agent):

Son entidades lógicas dentro de la red **SIP** y pueden ser identificados por la acción que realizan como Servidor (*User Agent Server*) cuando recibe una petición o Clientes (*User Agent Client*), cuando realiza una petición mensajes del protocolo **SIP**, ambos pudiendo ser una terminal final (PC, Smartphone, Laptop, teléfonos **IP**, etc) con periféricos multimedia de audio/video.

Cuando se realiza una llamada, una de las terminales (Cliente) envía un mensaje de texto **SIP** (INVITE) al otro (Servidor) el cual acepta o rechaza la petición con otro mensaje de texto **SIP** (200 para aceptar o 500 para rechazar), es decir, para poder entablar una llamada con **SIP** bastan 2 mensajes.

2. Servidor Proxy (Proxy Server)

Esta terminal actúa como Cliente y Servidor (Figura 2.27) siendo el intermediario de las señalizaciones **SIP** haciendo las peticiones a nombre del Cliente que las solicite y a nombre del Servidor que responde. En general brinda los siguientes servicios:

- Enrutamiento de llamadas.
- Admisión.
- Autenticación.
- Permisos de llamadas.
- Interpreta y rescribe mensajes (de ser necesario)

El Servidor Proxy opera con el Registrar Server para localizar los recipientes de las llamadas en el dominio local y con el Redirect Server para localización en otros dominios **SIP**, donde hay otros Proxy Server [2.7] y existen dos tipos:

- **Stateful**: Tiene memoria y crea un estado de petición, lo mantiene hasta que la comunicación finalice, ideal para el desvío de llamadas y replicación de llamadas.
- **Stateless**: Sin memoria realizando el enrutamiento para posteriormente enviar los mensajes, ideal para redes **SIP** de alta demanda **UDP**.

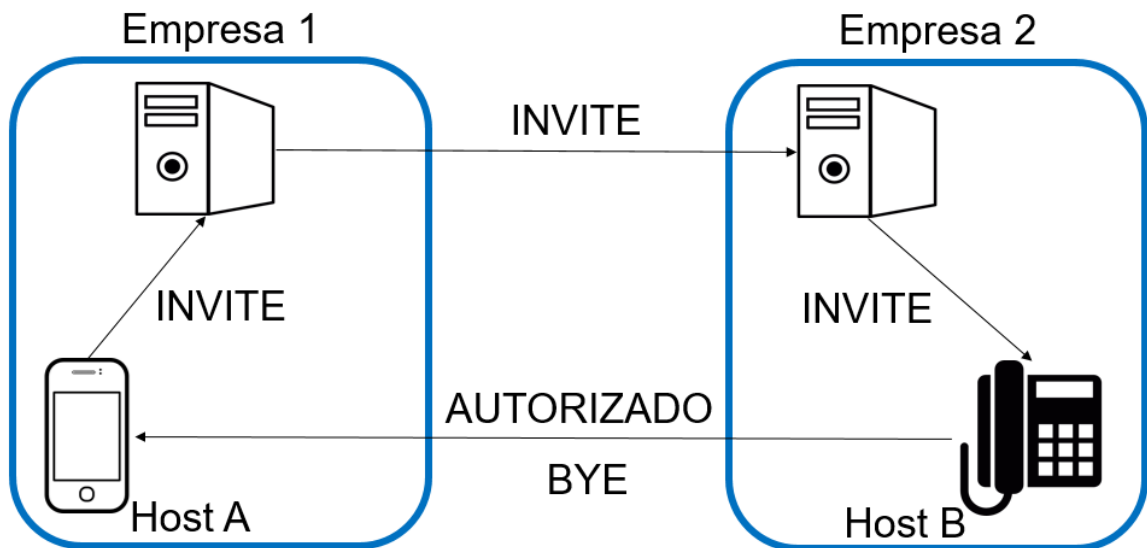


Figura 2.28 Función de Servidores proxy como Cliente y Servidor. [Elaboración propia]

Proceso de registro y solicitud de llamada por medio del Servidor Proxy (Figura 2.28).

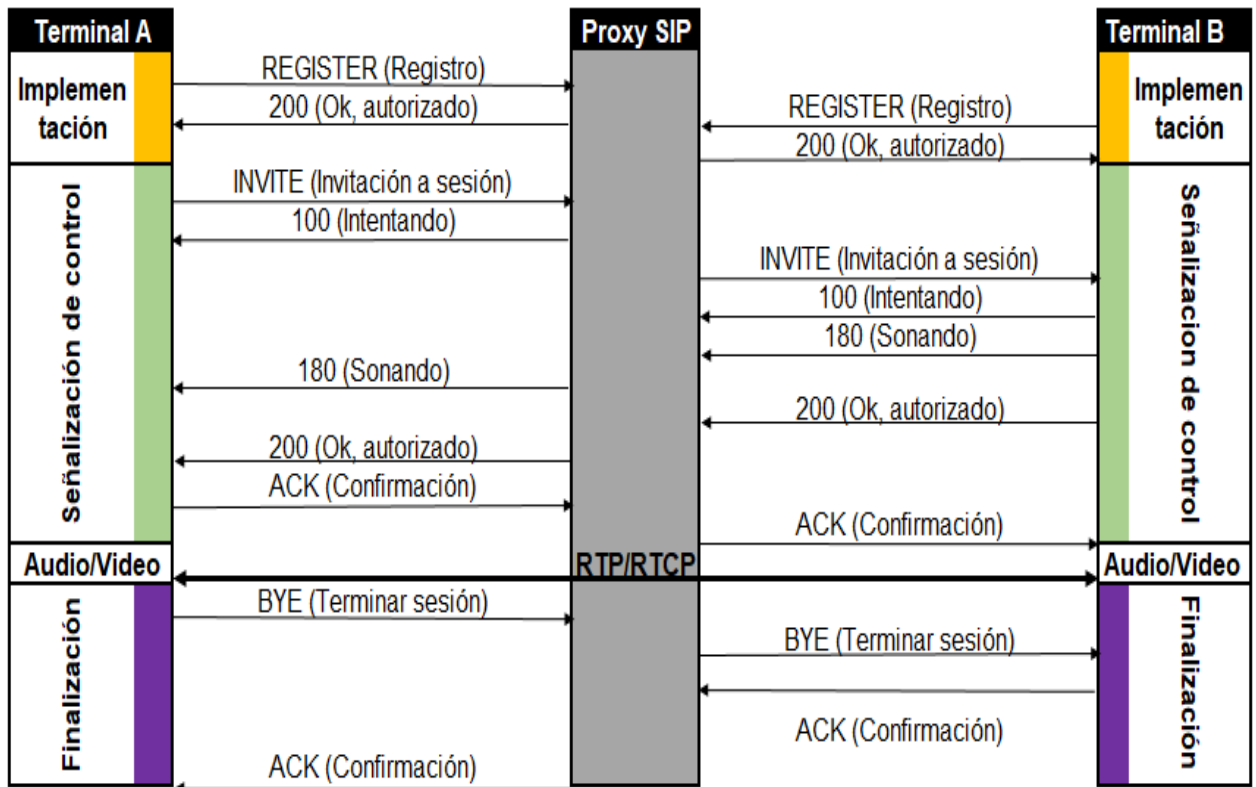


Figura 2.29 Servidor proxy como Cliente y Servidor. [Elaboración propia]

3. Servidor de Registro (Registrar Server)

Este servidor funciona para registrar a los usuarios almacenando su localización y asignándole un nombre **SIP**. El usuario tiene un **URI** (Identificador de Recursos Uniforme, *Uniform Resource Identifier*) independiente de su ubicación, el usuario

hace una petición REGISTER al servidor (Figura 2.29), para que este asocie la dirección lógica a una **IP** (*binding*) [2.4], comprendiendo los mensajes y autenticando la cuenta en la base de datos.

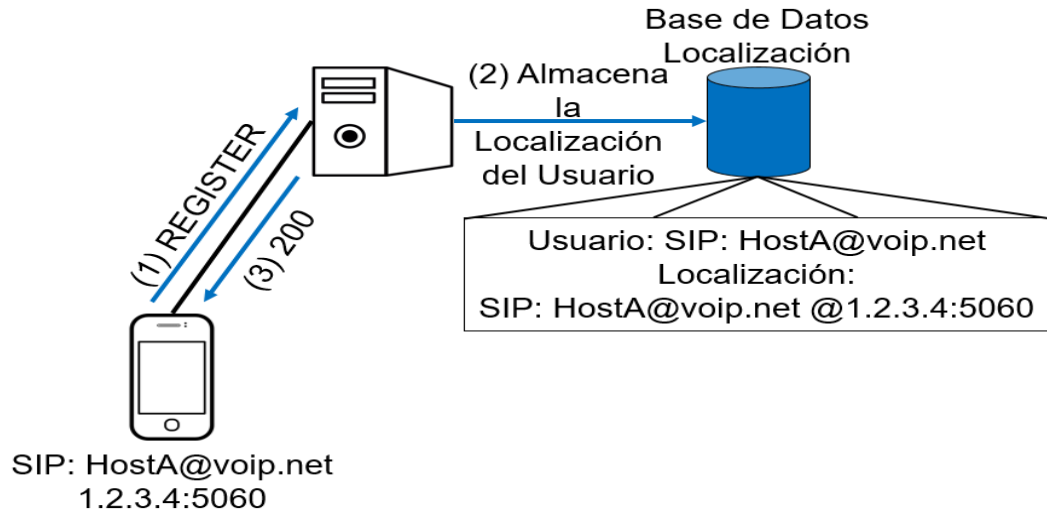


Figura 2.30 Servidor de Registro. [Elaboración propia]

4. Servidor de Redireccionamiento (Redirect Server)

Manejándose como solo Servidor a peticiones **SIP**, busca las direcciones solicitadas y entrega respuestas del tipo **3XX** y no reenvía las peticiones recibidas. Su única función es enviar la información de encaminamiento para una petición como respuesta al cliente, de manera tal que se quita del camino de los subsiguientes mensajes para una transacción⁴⁹, mientras ayuda en la localización de la petición [1.8] (Figura 2.30).

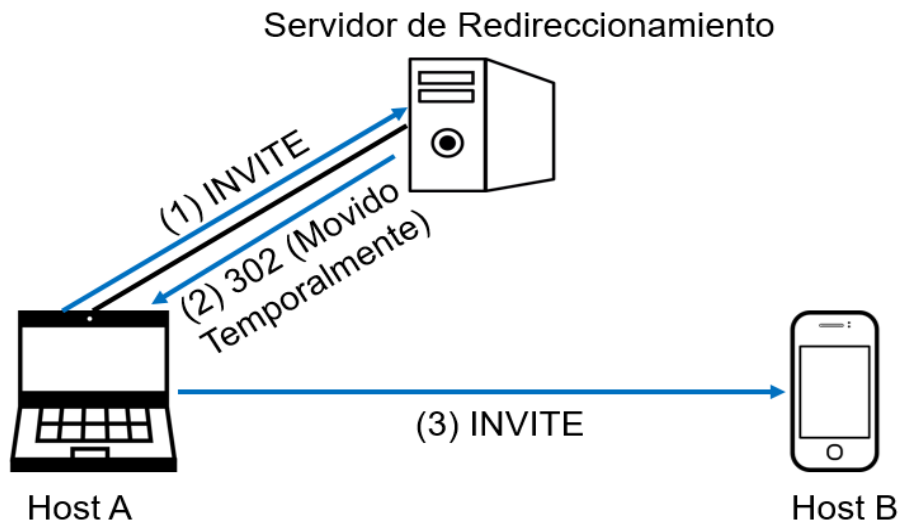


Figura 2.31 Servidor de Redireccionamiento. [Elaboración propia]

⁴⁹ Transacciones: Son todas las respuestas a peticiones (Serie de mensajes SIP).

Los diferentes Servidores (Proxy Server, Registrar Server y Redirect Server) son divisiones de forma conceptual, pudiendo estar todos trabajando en una misma terminal o en diferentes.

2.5.1.2.3 Servicios para PBX

Las herramientas **SIP** para una **PBX** funcionan básicamente como una central telefónica dentro de un servidor IP, ofreciendo servicios clásicos como bifurcación, transferencia, retención e identificación de llamadas, además de conferencias, buzón de voz, mensajería instantánea y videollamadas. También brindando la posibilidad de personalizar estos servicios. Para lograr estos servicios es necesario comprender 2 de sus protocolos más importantes que son:

- **Protocolo SDP.**

Diseñado para describir los detalles de las sesiones creadas durante algún evento multimedia para los dispositivos registrados. Proporciona información como:

- Propósito de la sesión creada.
- Nombre de la sesión y los participantes.
- Direcciones **IP** de los participantes.
- Números de puerto utilizados.
- Capacidades de los dispositivos conectados.
- Codificación/decodificación usada durante la sesión.
- Protocolo de transporte.
- Hora de inicio.
- Hora de finalización.
- Temporizador del evento.
- Tipo de multimedia.
- Medios que lo componen.

Esta información es fundamental ya que inicialmente **SDP** no negociaba las capacidades de los dispositivos, lo cual fue adicionalmente asignado a este protocolo.

- **RTP/RTCP.**

Cuando se tratan las problemáticas en la Capa de Transporte (Capa 4 del modelo OSI) es factible pensar en el protocolo **TCP/UDP** para el caso de transmisión de datos, sin embargo para la tecnología **VoIP** no es fiable implementarlo como tal ya que la variable a tratar en **VoIP** es que debe ser en tiempo real y adicionalmente el tema de datos como audio/video.

Para la solución a los problemas que no se pueden resolver mediante **TCP/UDP** se crearon los protocolos **RTP** y **RTCP**. **RTP** es utilizado para el intercambio de datos multimedia, mientras que **RTCP** se usa para el envío periódico de información de control asociada a un determinado flujo de datos [1.8]. Así, **RTP/RTCP** realizan funciones de:

- Codificación/Decodificación de datos.
- Identificación de usuarios.
- Relación temporal entre los datos recibidos.
- Sincroniza medios de diferentes tipos.
- Evitar la pérdida de paquetes multimedia.
- Límites de fotogramas.
- Tipo de compresión de datos.
- Solución a problemas de congestión.
- Diagnóstico de red.
- Tipo de carga útil.
- Marcas de tiempo.

Particularmente **RTCP** envía paquetes de información del estado de los datos transmitidos, definiendo 2 tipos de mensajes.

1. **Informe del receptor:** Indica la cantidad de paquetes recibidos y perdidos, estadísticas de las fluctuaciones y detalles de la capacidad del receptor.
2. **Informe de transmisor:** Comunicación entre el transmisor y receptor para obtener los detalles de latencia existentes en la red.

2.5.2.3 IAX (Protocolo Inter-Asterisk)

Diseñado para control y transmisión de paquetería multimedia, principalmente para **VoIP**, una de las principales características del protocolo **IAX** (*Inter-Asterisk Exchange*) es que utiliza un único puerto **UDP** (4569), para la señalización y el envío de paquetes multimedia de distintas comunicaciones a la misma vez y sin utilizar el protocolo **RTP**, esto se logra por medio de la introducción de campos numéricos en los paquetes **UDP** que identifican al transmisor y al receptor (*Source Call Number* y *Destination Call Number*), típicamente con sistemas Asterisk interconectados, pero el protocolo **IAX** es de código abierto que puede ser utilizado por cualquier equipo. La actual versión es **IAX2**, simplifica el proceso de llamadas por **VoIP** a través de Firewall (cortafuegos) y el flujo entre las redes que contienen dispositivos **NAT** (Network Address Translation).

El protocolo **IAX** utiliza frames (tramas IAX) de forma binaria para el flujo de información en la red, dichos frames se estructuran con las siguientes características:

1. **Bit F:** Indicador de si el frame transmitido se trata de un Full Frame, el cual funciona para enviar la señalización y audio/video, es decir, se encarga de la transmisión de los paquetes multimedia aunado a la señalización.
2. **Bit R:** Indicador que especifica si el frame está siendo retransmitido.
3. **Source Call Number:** Número de registro de la llamada que consta de 15 bits, ayuda que la llamada en curso no deba reutilizarse por otra, solo podrá utilizarse en otra llamada cuando el registro no esté activo.
4. **Destination Call Number:** Tiene el mismo número que el Source Call Number pero este identifica de forma única la llamada.
5. **Timestamp:** La marca de tiempo consta de 32 bits.
6. **OSeqno:** Consta de 8 bits y es el número de secuencias de entrada.
7. **Frame Type:** Identifica el tipo de frame (voz, video, texto, imagen) con 8 bits.

2.6 Consideraciones de calidad de servicio (QoS) para VoIP y ToIP.

La **QoS** (*Quality of Service*) tiene la función de darle control a los niveles de prioridad, implementado dentro de cualquier tipo de red, ya que garantiza el mínimo en calidad en una llamada y el tráfico de voz. El primer nivel prioritario es todos los datos que deben ser transmitidos en tiempo real como voz y video, dejando en un segundo término toda la transmisión de datos que no es fundamental en el tiempo de envío (datos).

Los parámetros de **QoS** en cualquier red son:

- **Ancho de banda** (Tabla 2.5)⁵⁰: Establece la máxima tasa de transferencia de paquetes dentro de una red. El ancho de banda se define mediante el tipo de codificación para **VoIP**.

⁵⁰ Calidad de servicio en VoIP: <http://bibing.us.es/proyectos/abreproy/12088/fichero/5+-+Calidad+de+Servicio.pdf>

Información del Códec				Cálculo del Ancho de Banda			
Códec	Tasa de bit (Kbps)	Tamaño de trama (bytes)	Tamaño de trama (ms)	Tamaño de datos (bytes)	Tamaño de datos (ms)	Paquetes por segundo	Ancho de banda (Kbps)
G.711	64	80	10	160	20	50	87.2
G.729	8	10	10	20	20	50	31.2
AMR	12.2	30.5	20	30.5	20	50	35.4
	20.2	25.5	20	25.5	20	50	33.4
	7.95	19.875	20	19.875	20	50	31.15
	7.4	18.5	20	18.5	20	50	30.6
	6.7	16.75	20	16.75	20	50	29.9
	5.9	14.75	20	14.75	20	50	29.1
	5.15	12.875	20	12.875	20	50	28.35
iLBC	4.75	11.875	20	11.875	20	50	27.95
	15.2	30	20	38	20	50	38.4
	13.33	50	30	50	30	33.3	28.8

Tabla 2.5 Anchos de banda dependiendo el Códec.

- **Retardo:** Es el tiempo que tarda un paquete en viajar a través de la red desde el host origen hasta el host destinatario.
- **Variación del retardo (*jitter*):** Es la variación que existe en el parámetro de retraso en la llegada de los paquetes en el receptor.
- **Pérdida de paquetes:** Es un acontecimiento que ocurre por la congestión que existe en la red, de forma que al no poder circular los nuevos paquetes debido a los que ya existen en la red se desechan y se pierde la información.

Pero en la tecnología de **VoIP** se tiene un enfoque diferente en la entrega de información en comparación con la redes de datos convencionales, **VoIP** debe entregar voz de manera mucho más confiable y con un menor tiempo de descarga de datos [1.8], creando 3 políticas de calidad: **Best Effort** (Servicio de mejor esfuerzo), **IntServ** (Servicios Integrados o ISA, *Integrated Service Architecture*) y **DiffServ** (Servicios Diferenciados o DS, *Differentiated Service*).

1. **Best Effort** (Servicio de mejor esfuerzo).

Se le denomina como servicio pero realmente no lo es, ya que puede ser cualquier red que no cuente con un modelo de calidad y la red funcione de manera natural haciendo que los datos intenten llegar a su destino sin ningún tipo de soporte, por ello no cuenta con ninguna garantía.

2. IntServ (Servicios Integrados).

Es un conjunto de estándares con la finalidad de proporcionar transporte con QoS basada en la reserva de recursos sobre interconexiones de redes basadas en IP. Utiliza el protocolo **RSVP** (Reserva de Recursos, *Resource Reservation Protocol*) en donde cada dispositivo participante en la red realice el protocolo de calidad y defina si la comunicación pueda realizarse o no. Brinda los siguientes tipos de servicios:

- **Garantizado:** Servicio más exigente dentro de los servicios integrados y consta de varios elementos clave:
 - Proporciona una tasa de datos fija (caudal).
 - Límite de retardo en estancia en cola aunado a la latencia obteniendo el retardo máximo.
 - Sin pérdidas en cola.
- **De carga controlada:** Servicio que brinda calidad en referencia a una red normal con el mínimo de carga y consta de los siguientes elementos:
 - Se adapta el comportamiento de las aplicaciones.
 - No especifica el retardo máximo permitido en colas, sin embargo, asegura que un gran porcentaje no tendrá afectaciones de retardos excedentes al mínimo.
 - Asegura a entrega de paquetes.

El protocolo **RSVP** brinda apoyo a los servicios **ISA** (Arquitectura Industrial Estándar, *Industry Standard Architecture*), soluciona una de las tareas fundamentales en la interconexión de redes que es la distribución de datos desde una fuente a uno o más destinos con la **QoS** solicitada, como puede ser el rendimiento, retardo, variación del retardo, etcétera [1.9]. Esto incrementa la dificultad cuando se vuelve una red de multidifusión ya que la cantidad de datos enviados genera un enorme tráfico dentro de la red a diferencia de una red de monodifusión.

El protocolo **RSVP** se utiliza para hacer reservaciones optimizando el ancho de banda y eliminando la congestión usando enrutamiento de difusión con árboles de expansión (Figura 2.31)⁵¹, en donde la interconexión de diferentes terminales se pueden traslapar sin ningún tipo de problema, haciendo que los receptores puedan apartar el ancho de banda necesario para el flujo de datos, evitando la congestión y logrando la calidad necesaria para la transmisión.

⁵¹ [1.10] "(a) El host 3 solicita un canal al host 1. (b) Después el host 3 solicita un segundo canal, al host 2. (c) El host 5 solicita un canal al host 1". Página 361.

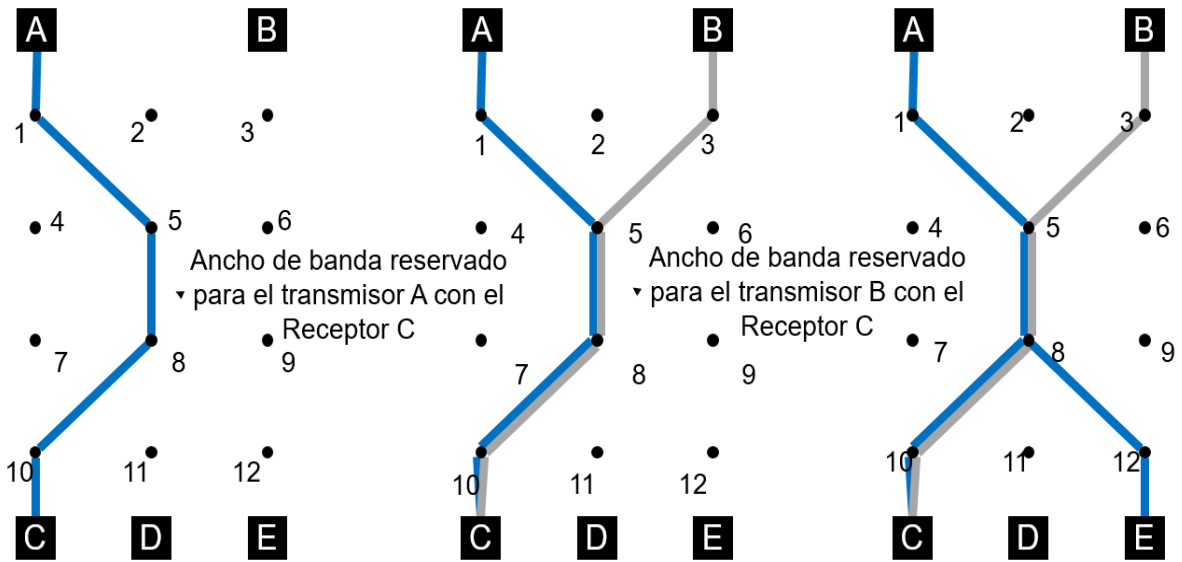


Figura 2.32 Reserva del ancho de banda.

3. DiffServ (Servicios Diferenciados).

La **ISA** y el protocolo **RSVP** son herramientas fundamentales para las redes privadas e Internet, pero su desventaja es que sus características son relativamente complejas de implementar y muy poco estables para una conexión de muchas terminales debido al tráfico que estos generan, por ello se crearon los **DS** (Servicios Diferenciados), los cuales están diseñados para proporcionar una herramienta simple, fácil de implementar y que suponga una escasa sobrecarga que ofrezca distintos servicios de red que estén diferenciados según su rendimiento [1.9], haciendo que los enrutadores se comporten de distinta manera dependiendo del tipo de servicio (comportamiento por salto).

Servicios Diferenciados consta de 5 características esenciales:

- **Clasificador:** Diferencia los paquetes enviados por la red basándose en el código **DS**, en la cabecera o de ser necesario en los datos del paquete.
- **Monitoreo:** Mide el tráfico enviado determinando si el flujo de paquetes cumple o excede el nivel de servicio.
- **Marcador:** Selecciona y clasifica los datos de mayor a menor relevancia asignándoles un servicio de envío prioritario o no.
- **Conformador:** Retarda los datos de menor relevancia para que no consuman parte del ancho de banda.
- **Descarte de paquetes:** Desecha paquetes cuando excede la tasa especificada en el perfil de clase.

Los **DS** brindan los siguientes servicios:

- **Servicio de reenvío urgente.** Su función es asegurar un servicio de calidad superior, asegurando una baja tasa de pérdidas, poco retardo y un ancho de banda eficiente.
- **Servicio de reenvío asegurado.** Funciona para redes que cuentan con el modelo del mejor esfuerzo de manera que no se necesite la reserva de recursos

Capítulo 3 Asterisk y Raspberry Pi

3. Características de un sistema de telefonía IP

3.1 Extensiones telefónicas por software

Las extensiones son caracteres alfanuméricos de identificación, asignados por el administrador del software al cliente o servicio durante el proceso de registro, al cual le da la capacidad de acceso a algunos o a todos los servicios que se brindan en la red de forma prioritaria dependiendo del tipo de protocolo.

El proceso que conlleva el registro y asignación de las extensiones depende de 2 factores que son:

- Servidor
- Terminales

Se puede apreciar en la Figura 3.1, que el servidor cuenta únicamente con 2 extensiones (101 y 102), la Terminal 1 y Terminal 2 deben contar con los datos de Usuario y Contraseña (previamente declaradas por el servidor) para que ambas puedan tomar las extensiones solicitadas. En cuanto a la Terminal 3, al solicitar una extensión inexistente es prácticamente imposible registrarla ante el servidor y brindarle los servicios de la red.

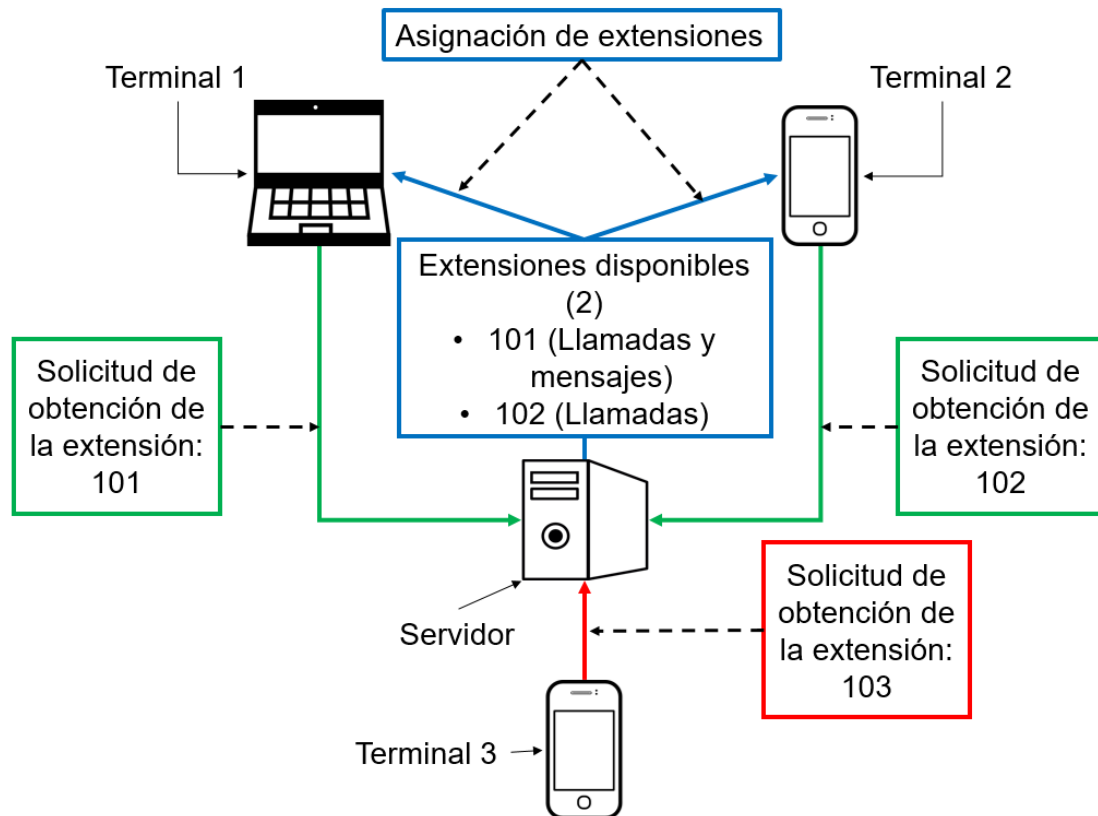


Figura 3.1 Registro de extensiones. [Elaboración propia]

3.1.1 Extensiones en servidor

Durante la declaración de las extensiones se debe tomar en cuenta los procesos por los que debe pasar un servicio disponible en la red (mensajes o llamadas), por ejemplo, para la asignación de una extensión a un cliente el cual realiza una llamada posterior al registro; se debe considerar la posibilidad de que la extensión a llamar no está disponible (ya sea porque está en uso o está desconectada de la red), por lo que debe hacerse saber a la terminal solicitante que la extensión marcada no se encuentra disponible y brindarle el servicio de Buzón de Voz, así al cliente que desea llamar se le reproducen una lista de acciones ejecutadas por el servidor. El siguiente ejemplo será en base a un servidor de telefonía IP: **Asterisk** (en la sección 3.2 de la presente tesis se estudia más afondo las características del software Asterisk), inicializando una familiarización con los procesos que se trabajan dentro de dicho servidor de telefonía sobre IP (Figura 3.2):

- Registro previo (Puerto y dirección IP)
- Recepción de llamada en el servidor
- Solicitud de llamada a la extensión solicitada
(exten => 101,1,Dial(SIP/101))
- Tono
- Aviso de la disponibilidad de la extensión llamada
(exten => 101,2,Dial(SIP/999))
- Buzón de voz
(exten => 101,102,Voicemail(101))
- Finalizar conexión
(exten => 101,103,Hangup)

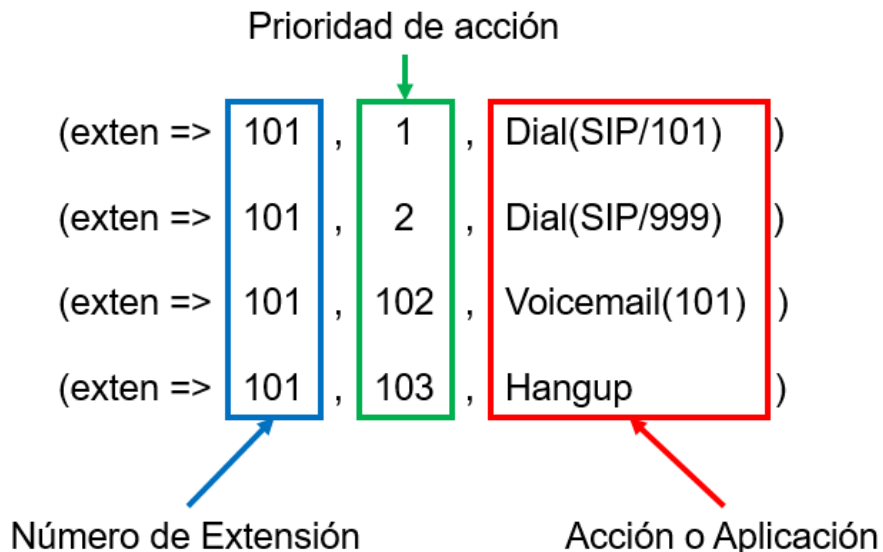


Figura 3.2 Extensiones (Asterisk). [Elaboración propia]

Por lo que dependiendo del registro, la acción en la lista lleva orden prioritario, de la misma manera funciona con los servicios brindados en la red, siendo así el encargado de darle guía al **Dialplan** (Plan de Marcado), esta guía se trata de

un lenguaje de comandos específicos y una de las formas principales de instruir al servidor sobre cómo comportarse, lo que permite enrutar y manipular los servicios de forma programada [3.2].

A continuación se presenta en la Tabla 3.1 algunas de las diferentes aplicaciones que se pueden enlistar dentro de las extensiones:

Comando	Acción o Aplicaciones
Answer()	Provoca un retardo en el momento de responder la llamada y recopila datos de los usuarios conectados.
Dial()	Realiza la conexión entre las diferentes extensiones.
Playback()	Carga y reproduce un mensaje a la terminal que realiza la llamada.
Hangup()	Finaliza el servicio en curso.
Bridge()	Realiza la conexión del canal de ejecución y el canal destino.
Directory()	Presenta a los usuarios un directorio de marcación por nombre.
Verbose()	Imprime en la interfaz el nombre del canal utilizado y su identificación única.
NoOp()	Imprime la duración de servicios en consola.
SMS()	Permite recibir o enviar mensajes a través de la PSTN.
VoiceMail()	Correo de voz.
Hint()	Mensaje instantáneo.
Goto()	Redirecciona el servicio a otro destino.
Busy()	Aviso de ocupado.
Background()	Reproducción de algún sonido antes del ingreso de la extensión.
Echo()	Prueba de latencia y calidad de la llamada.
MusicOnHold()	Reproduce un archivo de audio cuando se está en modo de espera durante una llamada.
Log()	Registra un audio personalizado desde el Plan de Marcado

Tabla 3.1 Acciones en las extensiones (Asterisk)⁵². [Elaboración propia]

Las acciones dependen de la versión instalada y de la misma manera si el software está configurado previamente para una correcta funcionalidad.

⁵² Aplicaciones Asterisk: <https://wiki.asterisk.org/wiki/display/AST/Applications>

3.1.2 Extensiones en terminales

Para la asignación de las extensiones para los usuarios se deben tomar en cuenta la lista antes presentada en el **Capítulo 2** en la **Sección 2.5.1.1 Terminales IP**, por el cual es posible enlazarse mediante el registro y extensión al dispositivo.

Existen comercialmente incontables aplicaciones con la tecnología **VoIP** para PC (Softphone) y Smartphones que pueden realizar la tarea de registrarse con la extensión asignada.

Un **Softphone** es un programa dedicado a funcionar o simular una terminal telefónica IP, estas aplicaciones al ser solo de software, son fáciles de instalar y actualizar y comúnmente tienen otras características que utilizan otros periféricos, como una cámara web para video-llamadas, o quizás la capacidad de cargar archivos multimedia desde su escritorio [3.8].

A continuación se presentan en la Tabla 3.2 las diferentes alternativas de programas y aplicaciones tipo Softphone:

Aplicación	Android (Play Store)	iOS (App Store)	PC (Softphone)	PC (WEB)	Servicios
Zoiper	Si	Si	Si	Si	Llamadas, mensajes, multimedia, video-conferencias y video-llamadas
Mobilevoip Cheap Calls	Si	Si	Si	No	Llamadas, mensajes
Sipnetic	Si	No	No	No	Llamadas, mensajes, video-llamadas
SIPSKY	Si	No	No	No	Llamadas, mensajes, video-llamadas
Bria mobile	Si	Si	No	No	Llamadas, mensajes, multimedia, video-conferencias y video-llamadas
Talkatone	Si	Si	Si	No	Llamadas, mensajes y multimedia
GnomeMeeting	No	No	Si	No	Llamadas y video-llamadas

Tabla 3.2 Parte I. Aplicaciones VoIP. [Elaboración propia]

Express Talk	No	No	Si	No	Llamadas y video-llamadas
---------------------	----	----	----	----	---------------------------

Tabla 3.2 Parte II. Aplicaciones VoIP. [Elaboración propia]

3.2 Asterisk

El software **Asterisk** (Figura 3.3) ha sido el software elegido para la implementación de un sistema de telefonía IP para el Laboratorio de Telecomunicaciones, debido a que es una plataforma de **Código Abierto** (Open Source) flexible que funciona de la misma manera que una **PBX** convencional pero olvidando lo analógico y la red de telefonía clásica, así, **Asterisk** pasa a ser una **IP-PBX** por sus comunicaciones digitales a través de la red de Internet.

Al ser solo de tipo software le confiere una robustez innata para desplegar servicios típicos de los sistemas tradicionales, pero aportando mucha más flexibilidad, control, creatividad y a muy bajo coste [3.1], de manera que es un conmutador híbrido al poder trabajar con la red telefónica convencional y la red de Internet.



Figura 3.3 Logo del programa Asterisk.

Asterisk es un software que se distribuía en 2 versiones:

- Versión **Open Source**
- Versión **Business Edition** (Edición de Negocios o Comercial)

En el año 2010 **Digium** (empresa diseñadora de Asterisk), dejó de brindar soporte a la versión comercial por parte de la misma (durante la contratación, Digium brindaban soporte personalmente durante 1 año) y dejó de distribuirla ya que la versión abierta fue igual o mayormente aceptada por las empresas. La versión abierta empezó a tener ventaja sobre la comercial y la única razón fue porque al ser de licencia libre (General Public Licence) el software podía compartirse con las mismas funciones que desempeñaban las de paga.

Además **Asterisk** es un sistema multiprotocolos, por lo que soporta diversos protocolos de señalización **VoIP** como:

- SIP
- H.323
- MGCP
- SCCP (Cisco Skinny)
- IAX

Codecs:

- ADPCM
- G.729 (through purchase of a commercial license)
- G.711 (alaw & μ law)
- G.722
- GSM
- G.723.1 (pass through)
- iLBC
- G.726
- Linear
- LPC-10
- Speex

3.2.1 Historia

El creador de **Asterisk** es Mark Spencer, quien lo diseñó debido a la necesidad de crear un servicio a clientes vía telefónica disponible durante el día y la noche, el servicio de telefonía era para su pequeña empresa relacionada con el soporte en temas del **Sistema Operativo GNU/Linux**, llamada **Linux Support Services** y fundada en 1999. Su problema radicaba en el costo de este servicio telefónico por lo que se propuso hacer una **PBX** del tipo software, derivado de ello nace el proyecto **Asterisk en Código Abierto**.

Un segundo problema surge durante el desarrollo de **Asterisk**, siendo este la interoperabilidad de la red telefónica convencional y la red **VoIP**, por lo que se realizó la adición del proyecto **Zapata Telephony** diseñado por Jim Dixon, permitiendo trabajar así con interfaces de la telefonía convencional.

Poco después de fundar **Linux Support Services**, Spencer notó que lo más valioso de su empresa no era el servicio de soporte sino el medio por el cual se brindaba el servicio, es decir, **Asterisk** era el producto mayormente subrayado, de forma que en el año 2001 **Linux Support Services** cambiaría su nombre a **Digium** (Figura 3.4).



Figura 3.4 Logo del programa Digium.

Asterisk fue inicialmente pensada para remplazar las **PBX** de la telefonía convencional, debido a ello puede existir la duda si **Asterisk** es una **PBX** o una **IP-PBX**, la respuesta es que puede ser ambas cosas debido a su incontables funciones, es decir, su funcionamiento depende de quien la programe dándole aún más sentido al **Open Source**. **Asterisk** es un conjunto de herramientas de comunicación. Tiene funciones que alguien necesita para convertirlas en algo: un distribuidor de llamadas automatizadas, un sistema de correo de voz, un sistema de correo por video, un **IP-PBX** o mucho más [3.4].

El 24 de agosto del año 2018 **Sangoma** (Figura 3.5)⁵³ anuncia la adquisición de todas las acciones de la empresa **Digium** siendo ahora el propietario de **Asterisk**.



Figura 3.5 Logo del programa Sangoma.

3.2.2 Sistemas Operativos donde se puede instalar Asterisk.

Asterisk al ser un software de código libre presenta una flexibilidad alta y puede ser instalado en cualquier plataforma o **Sistema Operativo** basado de forma nativa en **UNIX** como **GNU/Linux**, **BSD** (Distribución de Software Berkeley, Berkeley Software Distribution) y **MacOS-X**. Existe la solución de una simulación de los anteriores sistemas operativos en otros fuera de la complejidad **UNIX**, por ejemplo en Windows es posible la adquisición del emulador como **VirtualBox** o **VMware**, el simulado de sistemas operativos solo es recomendable si el diseño de la central telefónica es del tipo pasatiempo y posiblemente para una pequeña oficina o casa (dependiendo de las características de la PC con las que se cuenta), esto debido a la desviación de recursos y consumo de estos.

⁵³ Página Oficial Sangoma: <https://www.sangoma.com/press-releases/sangoma-announces-definitive-agreement-to-acquire-digium-inc/>

Es importante resaltar que el término **Linux** se utiliza para englobar la gran variedad de sistemas operativos de **Código Abierto** de diversas compañías, debido al uso del núcleo (Kernel) de **Linux**. A estas variantes de las conoce como distribuciones, todas y cada una de ellas comparten el mismo grupo de herramientas (comandos).

A continuación se presentan unos ejemplos de sistemas operativos donde **Asterisk** debería funcionar sin problema alguno (Tabla 3.3):

SO basados en UNIX		
Linux	MacOS-X	BSD
GNU/Linux	11.0 Big Sur	FreeBSD
Debian	10.15 Catalina	OpenBSD
Ubuntu	10.14 Mojave	NetBSD
Red Hat Enterprise Linux	10.13 High Sierra	MidnightBSD
openSUSE	10.12 Sierra	NomadBSD
CentOS	10.11 El capitan	DragonFly BSD
Fedora	10.10 Yosemite	

Tabla 3.3 Opciones de sistemas operativos para Asterisk. [Elaboración propia]

3.2.3 Requerimientos de Hardware.

En cuanto a los requerimientos **Asterisk** se asemeja mucho a cualquier aplicación que consume recursos del procesador y buses del sistema en tiempo real, por ello, una recomendación mayormente importante para sistemas complejos, es tener un servidor **Asterisk** con la única tarea de ejecutar las funciones dictadas por el mismo sistema de **Asterisk**, o minimizar las tareas que no son fundamentales para **Asterisk**, de forma que las prestaciones por el servidor no se vean afectadas en su rendimiento, de otra forma la probabilidad es grande de que existan problemas en la calidad del audio como son: la presencia de eco, la estática, la caída del servicio y demás.

La selección del hardware (Tabla 3.4)⁵⁴ es uno de los pilares en la instalación de un servicio de telefonía IP, pues a medida que aumenta el número de llamadas simultáneamente, el servicio está propenso a colapsar. En la siguiente tabla se dan las recomendaciones de acuerdo al tipo de propósito con el que se diseña y en donde se implementará el servidor.

⁵⁴[3.8] "Guía de requisitos para el sistema". Página 762.

3.2.3.1 Problemas en el desempeño.

Para comenzar con el proyecto lo primero se debe preguntar es: ¿Qué tan poderoso debe ser el sistema? La pregunta anterior debe segmentarse en otras preguntas, denotando el consumo de recursos pues una definición de la potencia del sistema se responde con la referencia de las necesidades.

Las siguientes preguntas facilitan el poder responder la pregunta inicial:

- ¿Qué características y servicios requieren los usuarios?
 - Llamadas
 - Conferencias
 - Mensajes
 - Interconectividad con la PSTN
 - Video-llamadas
 - Multimedia
- ¿Cuántas llamadas simultáneas habrá, en promedio y como máximo?

No solo con referencia a las llamadas, si no a la cantidad de mensajes, conferencias, video-llamadas y mensajes multimedia, por lo que cada conexión aumentará la carga de trabajo.

- ¿Qué porcentaje del procesador utilizarán los **Codec's**?

Dependiendo del servicio a brindar es el porcentaje del procesador por los Codecs, siendo matemáticamente un conjunto de reglas para digitalizar señales analógicas, por lo tanto, consume mayores recursos al momento de comprimir dichas señales.

- ¿Qué secuencia de acciones del **Dialplan** necesita?

Todas las acciones enlistadas para las extensiones conllevan una carga de trabajo, por lo cual es recordable tener esto muy en cuenta al momento de la elección del hardware.

- ¿Cuántos usuarios habrá inicialmente? ¿Cuántos usuarios habrá en un futuro?

Es importante saber el número de usuarios que serán beneficiados con el sistema instalado, aunado a ello es relevante saber que el servicio pueda satisfacer a los usuarios futuros que se añadan al servicio.

- ¿Cuántas extensiones habrá?

Tomando como base que no todas las extensiones deben vincularse a una terminal, cualquier extensión debe realizar una acción la cual nuevamente aumenta una carga laboral.

3.2.3.2 Procesador.

Una carga importante al procesador del servidor de Asterisk, es la demanda asombrosa de cálculos matemáticos complejos de la **CPU**. La eficiencia con la que se lleven a cabo estas tareas vendrá determinada por la potencia de la **FPU**⁵⁵ dentro del procesador [3.9].

El procesador elegido deberá ser mayormente potente para que cumpla con los requisitos mínimos con los que el sistema será diseñado, es decir, no es necesario instalarle al servidor un procesador de última generación si el servidor se ejecutará en una oficina pequeña.

3.2.3.3 Placa base.

La selección de una buena placa base está en referencia a la estabilidad y alto rendimiento:

- Proporciona la mínima latencia posible.
- Alta flexibilidad en los ajustes de **BIOS** (Sistema Básico de Entrada y Salida, Basic Input Output System).
- Placa base para servidor o estación de trabajo con o sin puertos **PCI** (Interconexión de Componentes Periféricos, Peripheral Component Interconnect) para una expansión y configuración del sistema.
- Múltiples núcleos beneficiará significativamente al sistema.

3.2.4 Tarjetas especializadas.

Las tarjetas especializadas también conocidas como tarjetas de interfaz, pueden ser tanto analógicas como digitales, del tipo **PCI** o **PCIe**⁵⁶, estas tarjetas funcionan para conectar los dispositivos al servidor **Asterisk**.

Las tarjetas de trabajo para el software de **Asterisk** más relevantes en el mercado son proporcionadas por la misma empresa **Sangoma** desde el año 2001, diseñadas para trabajar en conjunto con un servidor PC y expandir las funcionalidades del software de telefonía IP.

- **Analógica** (Tabla 3.5)⁵⁷.

Son creadas para conectar la red **PSTN** a través de una PC. Proporcionan soporte para líneas y teléfonos de la PSTN utilizando los módulos **FXS**⁵⁸ (Foreign eXchange

⁵⁵ FPU (Unidad de Punto Flotante del inglés Floating Point Unit) o también conocido como coprocesador matemático y funciona como la unidad encargada de los cálculos matemáticos de punto flotante.

⁵⁶ PCI y PCIe: Son buses que ayudan a la interconexión de hardware con la placa base.

⁵⁷ Sangoma. Las tarjetas de telefonía digital más fiables y flexibles del mercado:

<https://www.sangoma.com/telephony-cards/digital/>

⁵⁸ FXS: Interfaz brindada por el servicio de la PSTN para la conexión del cliente o usuario y ofrece los servicios como el tono de marcado, voltaje de alimentación y de timbre.

Subscriber), **FXO**⁵⁹ (Foreign eXchange Office) y **DSP** (Digital Signal Processor, para la cancelación de eco en las líneas) la cual cuenta con 4, 8 y 24 puertos para ranuras **PCI** y/o **PCIe**.

Características / Modelo	A4A	A4B	A8A	A8B	TDM2400	AEX2400
Puertos	4	4	8	8	24	24
	RJ11	RJ11	RJ11	RJ11	50-pin RJ21	50-pin RJ21
Tipo de Bus	PCI 2.2+	PCI-E 1.0+	PCI 2.2+	PCI-E 1.0+	PCI 2.2+	PCI-E 1.0+
Conector	3.3/5 V	X1	3.3/5 V	X1	3.3/5 V	X1
Dimensiones (cm)	16 x 6.3	16 x 6.3	15.7 x 10.6	15.7 x 10.6	31.19 x 10.67	31.19 x 10.67
Cancelación de eco	VPM032	VPM032	VPM032	VPM032	VPMOCT0 32	VPMOCT0 32

Tabla 3.5 Tarjetas analógicas Digium.

- **Digital** (Tabla 3.6)⁶⁰.

Proporcionan 1, 2, 4 y 8 puertos para ranuras **PCI** y/o **PCIe** con un módulo **DSP** por hardware para tarjetas digitales **T1 / E1 / PRI**.

Características / Modelo	TE1 33	TE134	TE205	TE2 01	TE235	TE4 05	TE410	TE436	TE820
Puertos	1		2		4			8	
Tipo de Bus	PCI-E 1.0+	PCI 2.2+	PCI 2.2 +	PCI 2.2+	PCI-E 1.0+	PCI 2.2+	PCI 2.2+	PCI-E 1.0+	PCI-E 1.0+
Conector	X1	3.3/5 V	5 V	3.3 V	X1	5 V	3.3 V	X1	X1
Dimensiones (cm)	12.7 x 5.4		12.7 x 9.53		12.7 x 6.7	12.7 x 9.53		12.7 x 6.7	16.8 x 9.5
Cancelación de eco	Integrado		VPMOCT 064	VOM0 64	VPMOCTT1 28	VPM1 28	VPM1 28	VPMO CT25 6	

Tabla 3.6 Tarjetas digitales Digium.

⁵⁹ FXO: Interfaz brindada por el servicio de la terminal del cliente que recibe el servicio de la PSTN y brinda la señalización de descolgado y colgado.

⁶⁰ Sangoma. Las tarjetas de telefonía digital más fiables y flexibles del mercado:

<https://www.sangoma.com/telephony-cards/digital/>

3.3 Opciones Comerciales de sistemas de telefonía IP. Opciones de tipo Open Source. Comparación.

Elastix - 3CX⁶¹

La **PBX Elastix** (Figura 3.7) es una distribución precompilada de **Asterisk** basada en el protocolo **SIP**, dispuesta a trabajar con el **Sistema Operativo CentOS** y con interfaz web de **FreePBX**.



Figura 3.7 Logo del software Elastix.

Actualmente **Elastix** ha dejado ser un producto de **Código Abierto** o basado en **Asterisk** y **FreePBX** para ser remplazado por **3CX** (Figura 3.8), funcionando así para **Linux**, **Windows** y **Raspberry**, siendo así ahora un software de paga dependiendo de la cantidad de usuarios y llamadas activas.



Figura 3.8 Logo del software 3CX.

eZuce - sipXcom⁶²

Es un software de **Código Abierto** desarrollado por **SIPfoundry** en 2004 basado en el protocolo **SIP** y compatible con un **Sistema Operativo Linux: CentOS 7**.

La versión **sipXecs** (Figura 3.9) se trata de un software diseñado como una plataforma para universidades y escuelas, enfocadas a la comunicación, siendo un software de paga contando con programas de apoyo y reducción de costos.

⁶¹ Página Web 3CX: <https://www.3cx.es/>

⁶² Página Web sipXcom: <http://sipxcom.org/>



Figura 3.9 Logo del software sipXcom.

FreeSWITCH⁶³

El software **FreeSWITCH** (Figura 3.10), es un servidor **SIP** de **Código Abierto** versátil y ejecutable en cualquier hardware básico, encargado de la transformación digital de medios en terminales PC, diseñado para trabajar en el **Sistema Operativo Debia 10, Buster y CentOS 7**.



Figura 3.10 Logo del software FreeSWITCH.

OpenSIPs⁶⁴

OpenSIPs (Figura 3.11) es un software de **Código Abierto** del tipo servidor bajo el protocolo **SIP** de **Código Abierto** para multimedia de alto rendimiento, facilidad de uso, flexibilidad e integración.



Figura 3.11 Logo del software OpenSIPs.

Kamailio⁶⁵

Kamailio (Figura 3.12) es un servidor **SIP** flexible de **Código Abierto** capaz de manejar miles de las llamadas activas que se realizan por el mismo software, con la opción de funcionalidad de ser una **PBX** al igual que las antes mencionadas.

⁶³ Página oficial de FreeSWITCH: <https://freeswitch.com/>

⁶⁴ Página oficial de OpenSIPs: <https://opensips.org/>

⁶⁵ Página oficial de Kamailio: <https://www.kamailio.org/>



Figura 3.12 Logo del software Kamailio.

Issabel⁶⁶

Issabel (Figura 3.13) es un software de **Código Abierto** gratuito basado en **Asterisk**, que permite crear herramientas de comunicación para empresas. Así permitiendo al usuario convertir este software en lo que quiera y necesite.



Figura 3.13 Logo del software Issabel.

A continuación se presenta en la Tabla 3.7 las diferentes características que ofrece cada uno de los anteriores softwares:

Características / Software	Asterisk	3CX	sipXecs	Free SWITCH	Open SIPs	Kamailio	Issabel
Año de inicio	1999	2005	2004	2009	2005	2001	2016
Adquisición	Gratis	Paga	Paga	Paga	Gratis	Gratis	Gratis
SO	Linux, MacOS-X y BSD	Linux y Windows	CentOS 7	Linux: Debian 10 Buster y CentOS 7	Linux	Linux	Linux
Número de extensiones	Ilimitado	Limitado	Limitado	Limitado	Ilimitado	Ilimitado	Ilimitado

Tabla 3.7 Parte I. Comparación entre IP PBX comerciales. [Elaboración propia]

⁶⁶ Página oficial de Issabel: <https://www.issabel.org/>

Características / Software	Asterisk	3CX	Sip Xecs	Free SWITCH	Open SIPs	Kamailio	Issabel
Nube	No	Amazon, Google y Azure	No	Signal Wire	No	No	No
Aplicación y software propios	No	Si (Windows, Web Client, iOS y Android)	No	No	No	No	No
Protocolos	SIP, H.323, MGCP, SCCP y IAX	SIP	SIP	SIP, H.323, SCCP Y IAX	SIP	SIP	SIP y AIX
Codecs	ADPCM, G.729, G.711, G.722, G.723.1, G.726, GSM, iLBC, Linear, LPC-10 Y Speex	G.711, G.119, G.722, G.723.1, G.726, G.728, G.729, GSM, iLBC, Speex, LPC10, DoD, CELP, VP8, H.264 Y AVC	G.729	G.711, G.722, G.722.1 Y G.729	G.711 y G.729	G.711, G.722, G.732.1, G.729, GSM, iLBC, Opus y AMR	G.729 y Opus

Tabla 3.7 Parte II. Comparación entre IP PBX comerciales. [Elaboración propia]

3.4 Raspberry Pi 3.

La **Raspberry Pi (RBP)** (Figura 3.14) es una microcomputadora de bajo costo del tamaño de una tarjeta de crédito, relacionado al concepto de **Código Abierto** y destinada a trabajar de la misma manera que una PC de escritorio (navegación en internet, elaboración de documentos de texto, reproducción multimedia e incluso videojuegos), a la exploración de la informática y la programación en diferentes lenguajes como **Scratch** y **Python**.

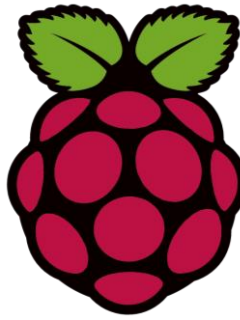


Figura 3.14 Logo de la Raspberry Pi.

La gama de modelos de la **Raspberry Pi** consta de cinco principales: **RBP Model A+**, **RBP Model B+**, **RBP 2**, **RBP 3**, **RBP 4** y **RBP Zero**, en donde algunas como el **Modelo 3** y **Zero** cuenta con mejoras y versiones **Plus** como es la **RBP Model 3 A+**, **RBP Model 3 B+** y **RBP Model Zero W**.

Las primeras tarjetas de este microordenador eran el **Modelo A** y **B**, las cuales ya no se fabrican, sin embargo el software que se diseñan para los otros modelos recientes como al **Modelo 3 B+** y **4** aún suelen ser compatibles con estos modelos. Los **Modelos A+** y **B+** no cambiaron del **SoC** (Sistema en Chip, System on Chip) **BCM2835**, por ello no hay una diferencia relevante en el rendimiento con los modelos anteriores (**A** y **B**).

A continuación se presenta en la Tabla 3.8 las características de cada modelo de la tarjeta **RBP**:

Modelo / Características	SoC	CPU	RAM	USB	Video	Audio	Almacenamiento	Wifi	Red	Precio M/N
A	BCM2835	ARM 1176 JZFS a 700 MHz	256 MB	1	RC A y HD MI	3.5 mm	SD	No	No	Descontinuado
B	BCM2835	ARM 1176 JZFS a 700 MHz	256 MB o 512 MB	2	RC A y HD MI	3.5 mm	SD	No	Fast Ethernet	Descontinuado
A+	BCM2835	ARM 1176 JZFS a 700 MHz	256 MB o 512 MB	1	HD MI	3.5 mm	micro SD	No	No	407.47

Tabla 3.8 Parte I. Características Raspberry Pi. [Elaboración propia]

Modelo / Características	SoC	CPU	RAM	USB	Video	Audio	Almacenamiento	Wifi	Red	Precio M/N
B+	BCM2835	ARM 1176J ZF-S a 700 MHz	510 MB	4	HDMI	3.5 mm	micro SD	No	Fast Ethernet	509.37
2B	BCM2836	Cortex-A7 de cuatro núcleos a 900 MHz	1 GB	4	HDMI	3.5 mm	micro SD	No	Fast Ethernet	713.11
3B	BCM2837	4 núcleos de 64 bits a 1.2 MHz	1 GB	4	HDMI	3.5 mm	micro SD	Si	Fast Ethernet	713.11
3A+	BCM2837B0	Cortex-A53 SoC de 64 bits a 1,4 GHz	512 MB	1	HDMI	3.5 mm	micro SD	Si	No	509.37
3B+	BCM2837B0	Cortex-A53 SoC de 64 bits a 1,4 GHz	1 GB	4	HDMI	3.5 mm	micro SD	Si	Gigabit Ethernet	713.11
4B	BCM2711	Arm Cortex-A72 de cuatro núcleos y 1,5 GHz de 64 bits	2 GB, 4GB y 8GB	4	2 micro HDMI (4K)	3.5 mm	micro SD	Si	Gigabit Ethernet	731.11/1120.6/1528.1

Tabla 3.8 Parte II. Características Raspberry Pi. [Elaboración propia]

Modelo / Características	SoC	CPU	RAM	USB	Video	Audio	Almacenamiento	Wifi	Red	Precio M/N
Zero W	BCM 2835	1 núcleo de 1 GHz	512 MB	1 micro USB (datos)	micro HDMI	No	micro SD	Si	No	203.75
400	BCM 2711	Arm Cortex-A72 de cuatro núcleos y 1,5 GHz de 64 bits	4 GB	3	2 micro HDMI (4K)	No	micro SD	Si	Giga bit Ethernet	1424.41

Tabla 3.8 Parte III. Características Raspberry Pi. [Elaboración propia]

3.4.1 Arquitectura de Hardware.

La **Raspberry Pi** utiliza un chip **SoC** que contienen módulos de la **CPU** (Unidad Central de Procesamiento, Central Processing Unit), **GPU** (Unidad de Procesamiento Gráfico, Graphics Processing Unit), **DSP**, **RAM** (Memoria de Acceso Aleatorio, Random Access Memory) y puertos **USB**, no cuenta con disco duro por lo que tiene una ranura para la inserción de una tarjeta **SD**.

Este microordenador consiste de un microprocesador y sus diversas conexiones externas hacia la tarjeta tales como (dependiendo del modelo de la Raspberry Pi) (Figura 3.15):

- **Broadcom con arquitectura ARM** (Advanced RISC Machine)
- **Ranura para tarjeta SD** (Secure Digital), **micro SD**, **SDHC** (Secure Digital High Capacity) o **MMA** (MultiMediaCard): Funciona como el disco duro con el **Sistema Operativo** previamente instalado y el resto de la memoria se usa como almacenamiento para el usuario.
- **Conector HDMI** (High-Definition Multimedia Interface): Salida de video y audio de la placa.
- **Conector de Ethernet**: Para realizar la conexión a la red por medio de un cable **Ethernet RJ45**.
- **Puertos USB 2.0**: Para conectar dispositivos vía **USB** como memorias, teclado, mouse, discos duros externos, entre otros.
- **Salida de audio 3.5 mm**: Salida analógica de audio.
- **Conector RCA** (Radio Corporation of America): Salida de video opcional al **HDMI**.

- **Conector DSI** (Display Serial Interface): Para la conexión de una pantalla de cristal líquido.
- **Conector CSI** (Camera Serial Interface): Puerto de 15 pines para la instalación de una cámara.
- **Encabezado GPIO** (Entrada y Salida de Propósito General, General Purpose Input/Output): Serie de conexiones genéricas, pensadas para la comunicación con diferentes dispositivos para el ingreso de datos. Son un conjunto de pines que permiten comunicar el procesador con el exterior, por lo tanto, se pueden programar mediante software tanto señales de entrada como de salida [3.3].
- **Conector de alimentación micro USB**: Suministro de voltaje para el funcionamiento de la **RBP** de 5 V y al menos 500 mA para los modelos A y de 700 mA para los **Modelos B**.

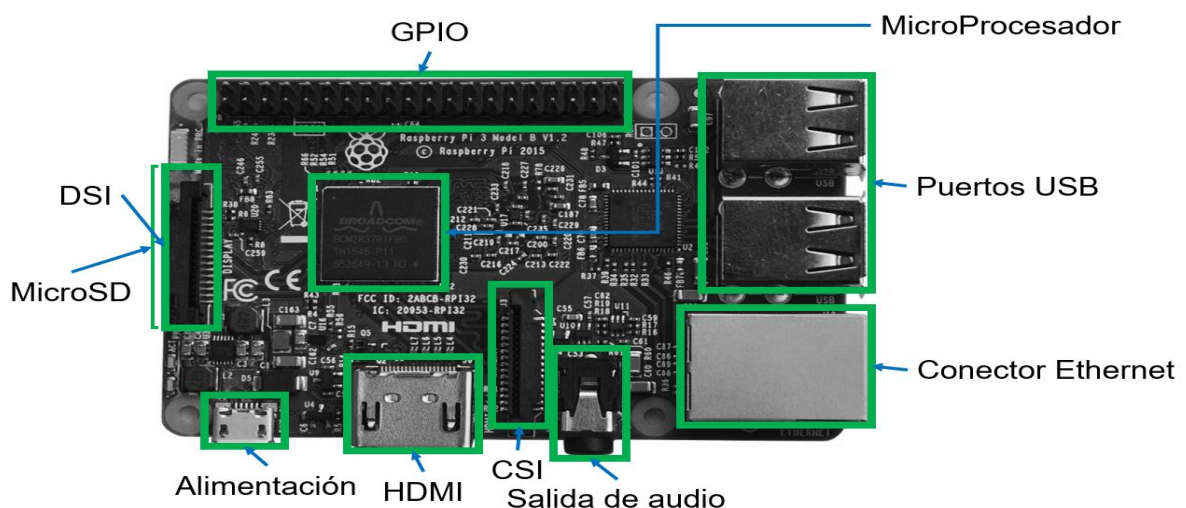


Figura 3.15 Raspberry Pi 3 modelo B. [Elaboración propia]

3.4.2 Arquitectura ARM.

Las **Raspberry Pi** utilizan un **ISA** (Repertorio de Instrucciones, Instruction Set Architecture) denominado como **ARM**, el cual fue desarrollado por Acorn Computers en 1980. Esta arquitectura es del tipo **RISC** (Computador con Repertorio de Instrucciones Reducidas, Reduced Instruction Set Computing) con un bajo consumo de energía.

La familia **SoC BCM283X** es la parte fundamental de las **RBP** por el hecho de estar basadas en la arquitectura **ARM**, la cual no puede trabajar con el software existente en las PC convencionales ya que están diseñados con una arquitectura **x86** con un **ISA** del tipo **CISC** (Computador con Repertorio de Instrucciones Complejas, Complex Instruction Set Computer).

La **SoC BCM2835** que se encuentra en el **Model A**, **Model B**, **Model A +**, **Model B +**, **Pi Zero** utiliza la arquitectura **ARM11**, la **BCM2836** que se encuentra en **RBP 2** utiliza la **ARMv7**, lo que le otorga un mayor rendimiento y compatibilidad con una gama más

amplia de sistemas operativos, la **BCM2837** (Figura 3.16) que se encuentra en la **RBP 3** que utiliza **ARMv8** de 64 bits, lo que aumenta aún más el rendimiento [3.7] y la **BCM2711** funciona con la **RBP 4** con **ARM A72**.



Figura 3.16 BCM2837 de Raspberry Pi.

3.4.3 Sistemas Operativos soportados.

El **Sistema Operativo** es un conjunto de reglas, órdenes y programas con la finalidad de la ejecución de varias tareas en el procesador, sirviendo así como el intermediario usuario/PC inicializando el hardware, controlando los dispositivos y administrando tareas y archivos. También tiene la misión de administrar los recursos de la máquina de proceso para evitar que los programas entren en conflicto y facilitar la ejecución de los programas que el usuario desee [3.3].

Como la tarjeta **RBP** es un microordenador necesita un **Sistema Operativo** por el que pueda realizar lo antes mencionado. Por lo que se hará mención de algunos sistemas operativos más utilizados.

3.4.3.1 Raspberry Pi OS

Anteriormente conocido como **Raspbian** (Figura 3.17), es un **Sistema Operativo** gratuito de Linux desarrollado especialmente para las tarjetas **RBP** basado en **Debian**, siendo uno de los más populares y utilizados debido a su enfoque educativo de programación en **Python** y de **Código Abierto**.

Es el **Sistema Operativo** primario del este microordenador, fue creado por un grupo de desarrolladores no afiliado a la tarjeta **RBP** pero que si son aficionados de esta.



Figura 3.17 Logo del Sistema Operativo Raspbian.

Actualmente cuenta con 2 versiones disponibles para instalar:

- **Raspberry Pi OS Lite** de **32-bit** sin escritorio.
- **Raspberry Pi OS Full** de **32-bit** con escritorio y aplicaciones recomendadas.

3.4.3.2 Ubuntu-Mate

Ubuntu-Mate (Figura 3.18) es otro **Sistema Operativo** multitarea basado en Ubuntu por parte de la empresa programadora **Canonical**, estable y fácil de usar por sus iconos y escritorio configurable.

Compatible con los modelos de **Raspberry Pi 2 B**, **3**, **3+** y **4**, debido a que sus requisitos de hardware son de bajos a moderados, permitiendo así una interacción dinámica y fluida con este **Sistema Operativo**.

Cuenta con un grupo variado de aplicaciones tales como: Calculadora, diccionario, LibreOffice, navegador Firefox, reproductor de multimedia Celuloide, webcamoid y una guía de iniciación del sistema en donde se citan las recomendaciones que se adecúen al usuario.



Figura 3.18 Logo del Sistema Operativo Ubuntu Mate.

Las diferentes versiones disponibles dependen del equipo en donde se cargara este

Sistema Operativo:

- **Raspberry Pi de 32 bits**
 - 20.04.1 LTS Focal Fossa
 - 20.10 Groovy Gorilla
- **Raspberry Pi de 64 bits**
 - 20.04.1 LTS Focal Fossa
 - 20.10 Groovy Gorilla

3.4.3.3 Ubuntu

Otro **Sistema Operativo** es **Ubuntu** (Figura 3.19), el cual es de **Código Abierto** y que se usa en todo el mundo, con todas las herramientas en apoyo a investigadores, estudiantes, docentes, empresarios y excéntricos del mundo de la programación e informática para el hogar, escuela o trabajo.



Figura 3.19 Logo del Sistema Operativo Ubuntu.

Cuenta con una diversidad de versiones dependiendo del dispositivo a instalársele:

- **Ubuntu con escritorio 20.10** para **RBP 4/400**.
- **Ubuntu Server 20.10** de **32 bits** para **RBP 2, 3, 4 y 400**.
- **Ubuntu Server 20.10** de **64 bits** para **RBP 3, 4 y 400**.
- **Ubuntu Server 20.04.1 LTS** de 32 bits para **RBP 2, 3 y 4**.
- **Ubuntu Server 20.04.1 LTS** de 64 bits para **RBP 3 y 4**.
- **Ubuntu Core 18** de **32 bits** para **RBP 2, 3 y 4**.
- **Ubuntu Core 64** de **32 bits** para **RBP 3 y 4**.

3.4.3.4 Risc OS

Es otro **Sistema Operativo** de **Código Abierto** con su propio núcleo es **Risc OS** (Figura 3.20), el cual, no tiene ninguna relación con **Linux** o **Windows**, diseñado para procesadores **ARM** rápido y compacto.

Muchos de los programas precargados en este **Sistema Operativo** no requieren de un procesador rápido, debido a ello utiliza muy pocos recursos aún siendo del tipo multitarea.

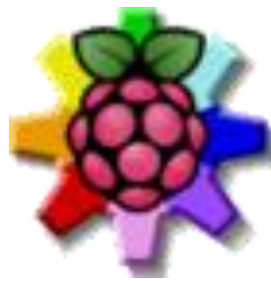


Figura 3.20 Logo del Sistema Operativo Risc OS.

Sus versiones son las siguientes:

- **RISC OS (5.28)** para los **Modelos RBP Zero, Zero W, 1A+, 1B+, 2B, 3A+, 3B+ y 4B.**
- **RISC OS (5.28)** para el **Modelo RBP 400.**

3.4.3.5 LibreElec

Siendo una distribución más de **Linux**, **LibreElec** (Figura 3.21) se trata de un **Sistema Operativo** de **Código Abierto** y gratuito diseñado específicamente para dispositivos embebidos enfocados para el entretenimiento, para lo anterior, se implementa la aplicación **Kodi** que permite al dispositivo ser un centro multimedia en el que se pueden reproducir videos, música o fotografías.



Figura 3.21 Logo del Sistema Operativo LibreElec.

Sus versiones son:

- Dispositivos con cualquier distribución de **Linux** de **32 bits.**
- Dispositivos con cualquier distribución de **Linux** de **64 bits.**

3.4.3.6 Windows 10 IoT Core

Este **Sistema Operativo** es una versión de **Windows 10** (Figura 3.22) gratuito y optimizada para dispositivos con arquitectura **ARM** y **x86 / x64** con o sin pantalla diseñada inicialmente para la **RBP 2**. Enfocado en el **IoT** (Internet de las Cosas,

Internet of Things) trabaja con pocos recursos y con el mismo núcleo de su versión para PC.



Figura 3.22 Logo del Sistema Operativo Windows 10 IoT Core.

3.4.3.7 ARCH LINUX ARM

El **Sistema Operativo** ArchLinux ARM (Figura 3.23) es otra distribución de Linux independiente de propósito general, desarrollado para equipos con arquitectura **ARM** y **x68 / x64**, su instalación es el sistema base (sin escritorio u otra aplicación), por lo que el usuario deberá ir agregando el contenido deseado para el fin del dispositivo.



Figura 3.23 Logo del Sistema Operativo Arch Linux ARM.

3.4.4 Puertos utilizados para conectar un servidor de telefonía.

A continuación se presenta un listado del mínimo requerido de dispositivos conectados (Figura 3.24) para que la tarjeta **RBP** y **Asterisk** funcionen de la mejor manera, por lo que algunos son obligatorios y otros opcionales dependiendo el **Sistema Operativo** y de la tarjeta **RBP**.

Ranura MicroSD: Este puerto funciona para la instalación dispositivo de almacenamiento micro SD con el cual el **Sistema Operativo** trabaja para ejecutar el servidor de telefonía.

Puerto de alimentación: Como todo dispositivo electrónico debe estar alimentado para que éste realice su trabajo. Para ello hará falta un alimentador de 5 V y al menos 500 mA o 700 mA como antes se ha mencionado.

Puerto HDMI o DSI: Servirá para conectar el dispositivo de salida que muestra los datos y la información al usuario, la pantalla/monitor servirá como interfaz para descargar, configurar, modificar, monitorear y reproducir la **PBX**.

Conector Ethernet: Fungiendo como el conector que adhiere al dispositivo (Raspberry Pi) a la red, en donde funcionará como servidor de telefonía por medio del cable Ethernet. Este puerto puede ser opcional, debido a que las tarjetas actuales cuentan con conexión inalámbrica, pero se recomienda el uso del puerto por las diversas

desventajas que existen en una conexión inalámbrica (retardo, desconexión, menor velocidad, entre otras)

Puertos USB: En estos puertos se puede conectar otros dispositivos de entrada y normalmente son ocupados para la conexión del mouse, teclado y memorias USB. Para la **PBX** el dispositivo de entrada por obligación es el teclado por el cual se ingresan los caracteres para la configuración del software, el mouse puede ser inservible si el **Sistema Operativo** funciona con un escritorio e iconos, de lo contrario todo el programa puede ser trabajado mediante el teclado.

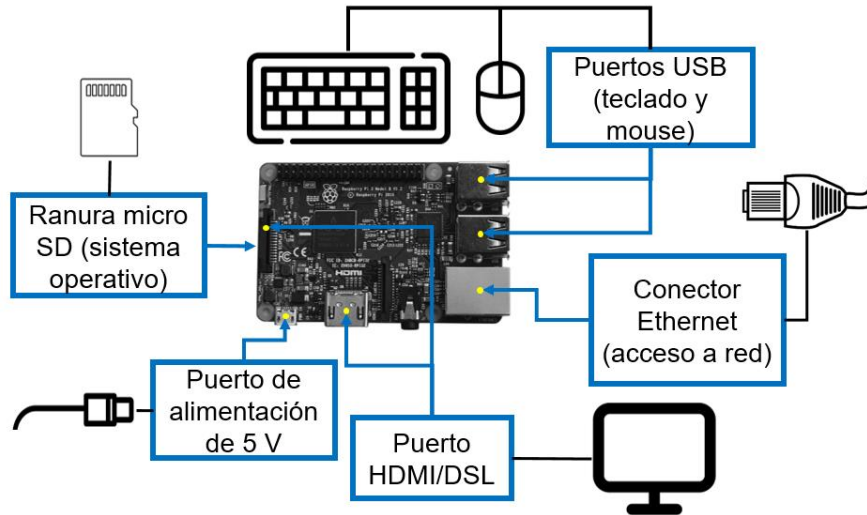


Figura 3.24 Periféricos usados por un PBX. [Elaboración propia]

Capítulo 4. Implementación de un sistema de telefonía IP con Raspberry Pi y Asterisk

4 Implementación.

4.1 Descarga e instalación del sistema operativo.

Se debe descargar el **Sistema Operativo** de su página oficial: **Ubuntu-Mate**⁶⁷. De igual manera es necesario descargar el software: **Win32 Disk Imager**⁶⁸.

Es necesario contar con un micro ordenador **Raspberry Pi 3 B**, al cual se le debe instalar el sistema operativo **Ubuntu Mate 18.04.2** (Figura 4.1) en una memoria **micro SD** de **16 GB** por medio del software **Win32 Disk Imager** (Figura 4.2, Figura 4.3 y Figura 4.4).



Figura 4.1 Ubuntu MATE.

⁶⁷ Descarga del Sistema Operativo: <https://ubuntu-mate.org/download/>

⁶⁸ Descarga del software instalador del Sistema Operativo: <https://win32diskimager.download/>

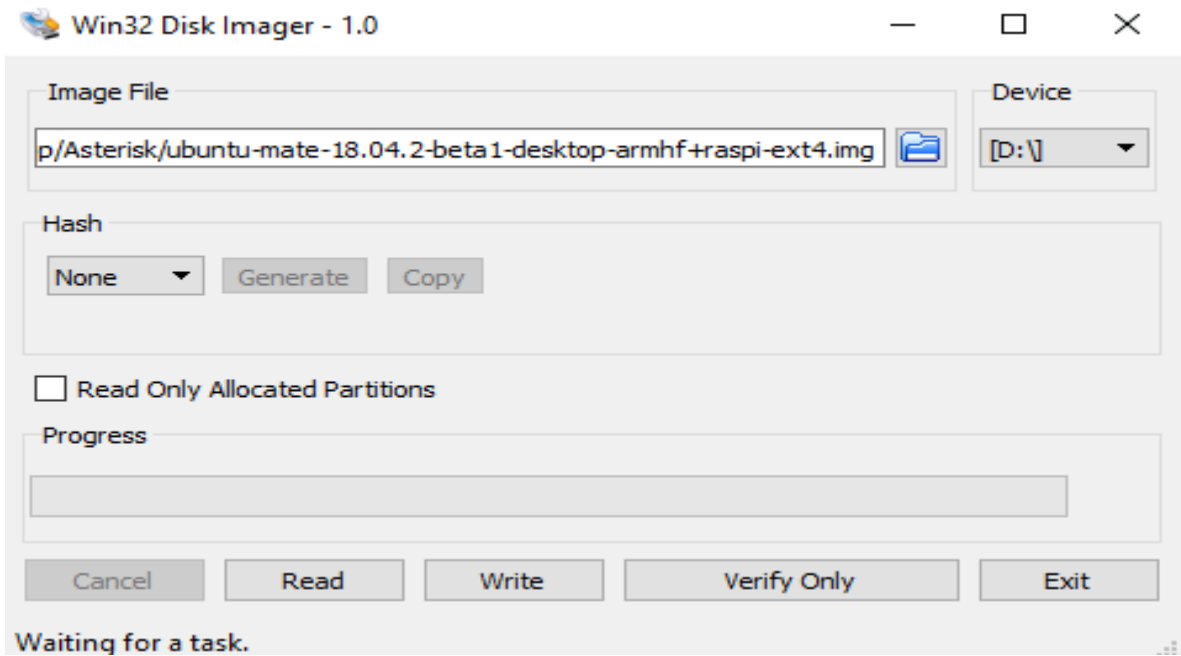


Figura 4.2 Inicio de instalación del Sistema Operativo 0%.

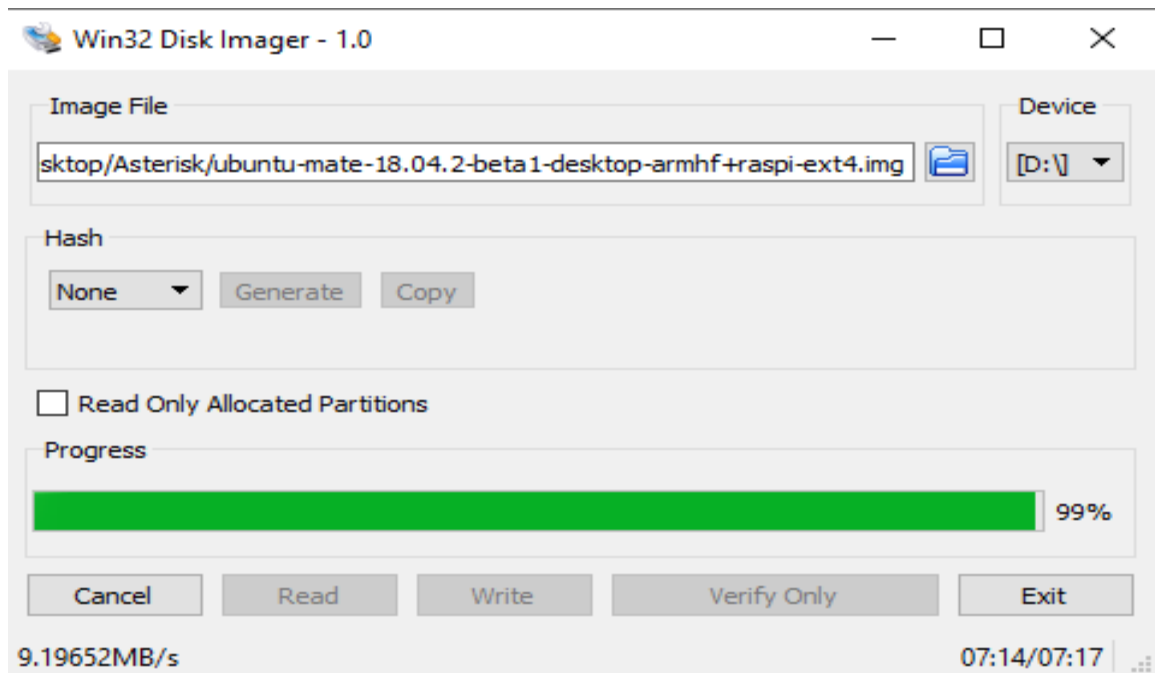


Figura 4.3 Inicio de instalación del Sistema Operativo 99%.

Una vez instalado el **Sistema Operativo** en la memoria **micro SD** la memoria debe ser insertada en la ranura de la **Raspberry Pi3** y lista para iniciar el sistema.

4.2 Configuración del Firewall y servicio SSH.

Ya instalado y configurado el **Sistema Operativo**, es necesario habilitar el servicio **SSH** en la Raspberry Pi 3, así, el servidor es accesible de forma remota en otra terminal, por lo que en un inicio se obtiene la información de los puertos activos, buscando específicamente el puerto 22 para el SSH por medio del siguiente comando (Figura 4.5):

4.2.1 Sudo -i (Comando 4.1)

Accede al modo Administrador (SuperUsuario o Root).

4.2.2 iptables -L (Comando 4.2)

Muestra el estado del firewall (Figura 4.5), supervisa los canales de entrada y salida, permitiendo solo el tráfico selectivo al servidor.

```
Comando 4.2
root@Asterisk-ISET:~# iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source                destination
Chain FORWARD (policy ACCEPT)
target    prot opt source                destination
Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination
root@Asterisk-ISET:~#
```

No existe ningún puerto en las reglas de entrada

Figura 4.4 Comando iptables -L.

Si no está habilitado el puerto 22 en el canal de entrada, se debe habilitar el puerto que corresponde al **SSH** por medio de la siguiente línea de comando (Figura 4.6):

4.2.3 iptables -A INPUT -p tcp --dport 22 -j ACCEPT (Comando 4.3)

Habilita el puerto descrito como canal de entrada al servidor.

Verificar nuevamente con el comando **iptables -L** que el puerto 22 ha sido activado.

```
root@Asterisk-ISET:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
root@Asterisk-ISET:~# iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source                destination
ACCEPT    tcp  --  anywhere              anywhere             tcp dpt:ssh
Chain FORWARD (policy ACCEPT)
target    prot opt source                destination
Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination
root@Asterisk-ISET:~#
```

Comando 4.3

Comando 4.2

Confirmación del puerto 22 (SSH)

Figura 4.5 Activación y verificación del puerto 22.

Posteriormente se genera una contraseña por default al servicio **SSH**, la cual es fundamental para poder ingresar al servidor de forma remota:

4.2.4 /usr/bin/ssh-keygen -A (Comando 4.4)

Crea una contraseña automática que ofrece seguridad de nivel medio (Figura 4.7).

Comando 4.4

```
root@Asterisk-ISET:~# /usr/bin/ssh-keygen -A
ssh-keygen: generating new host keys: RSA DSA ECDSA ED25519
root@Asterisk-ISET:~#
```

Contraseña

Figura 4.6 Generar contraseña.

Se inicia el servicio **SSH** (Figura 4.8):

4.2.5 service ssh start (Comando 4.5)

Sirve para inicializar el servicio **SSH**.

Se verifica el estado del **SSH**:

4.2.6 service ssh status (Comando 4.6)

Sirve para verifica el estado del servicio SSH (Active (Running)).

Comando 4.5

```
root@Asterisk-ISET:~# service ssh start
root@Asterisk-ISET:~# service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; disabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-04-27 12:36:09 CDT; 1s ago
     Process: 32564 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 32565 (sshd)
       Tasks: 1 (limit: 2154)
      CGroup: /system.slice/ssh.service
             └─32565 /usr/sbin/sshd -D

abr 27 12:36:09 Asterisk-ISET systemd[1]: Starting OpenBSD Secure Shell server...
abr 27 12:36:09 Asterisk-ISET sshd[32565]: Server listening on 0.0.0.0 port 22.
abr 27 12:36:09 Asterisk-ISET sshd[32565]: Server listening on :: port 22.
abr 27 12:36:09 Asterisk-ISET systemd[1]: Started OpenBSD Secure Shell server.
root@Asterisk-ISET:~#
```

Servicio SSH Activo

Comando 4.6

Figura 4.7 Estado del servicio SSH.

Se confirma de la anterior imagen que el servicio está activo por el puerto 22 por lo que es posible interactuar con el servicio SSH.

En otra terminal que no sea la **Raspberry Pi 3** se inicia el programa **Putty**⁶⁹ (Figura 4.9) para comenzar la configuración del servicio de Asterisk.

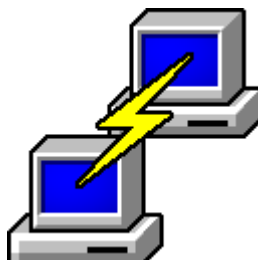


Figura 4.8 Logo del software Putty.

Para saber la **dirección IP** de la **RBP 3** es necesario el siguiente comando:

4.2.7 ifconfig

(Comando 4.7)

Asignación y configuración de la dirección IP del dispositivo.

Una vez que se sabe la **dirección IP** de la **RBP 3** es momento de ingresarla en el programa **Putty**, el **número de puerto** (22) y el tipo de conexión (SSH) (Figura 4.10).

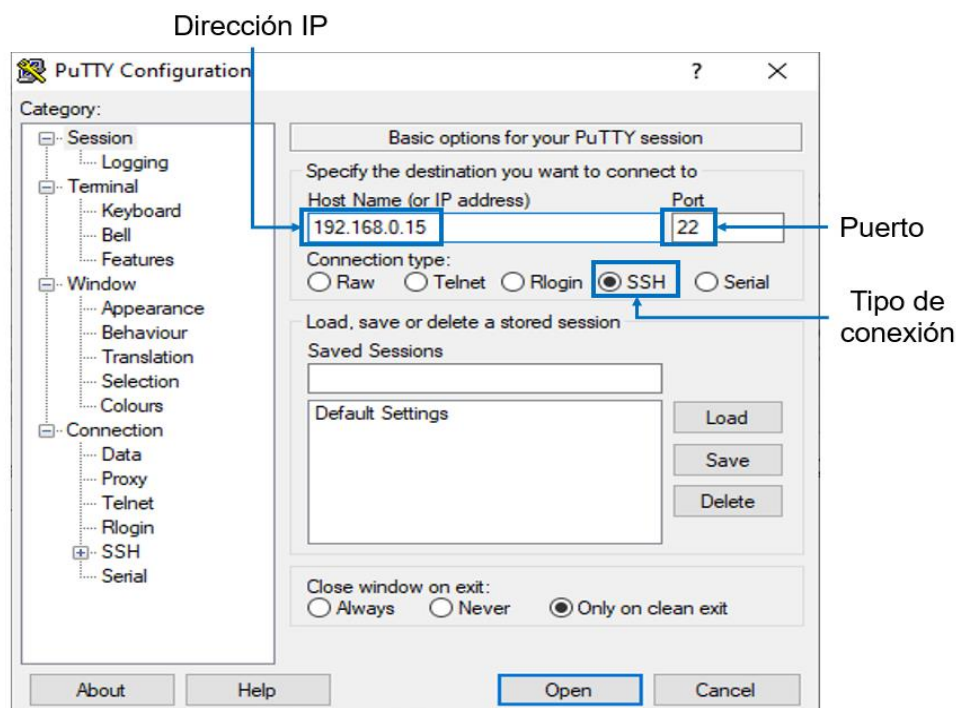


Figura 4.9 Ingreso inicial a la RBP 3 con dirección IP y puerto por medio de Putty.

⁶⁹ Descarga del software Putty: <https://www.putty.org/>

Al abrir sesión con **Usuario** registrado al iniciar el **Sistema Operativo** y la **Contraseña** otorgada al ingresar el **Comando 4.3** (por motivos de seguridad del sistema no aparecerán los caracteres ni asteriscos) (Figura 4.11).

```

      Usuario
      ↓
login as: uacm-iset
uacm-iset@192.168.0.15's password: ██████████ ← Contraseña
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-raspi2 armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Pueden actualizarse 0 paquetes.
0 actualizaciones son de seguridad.

Last login: Mon Apr 27 13:20:43 2020 from 192.168.0.13
uacm-iset@Asterisk-ISET:~$ █
```

Figura 4.10 Ingreso a la RBP 3 con Usuario y contraseña por medio de Putty.

4.3 Instalación y configuración del Servicio Asterisk

Al iniciar nuevamente como **Administrador**, pero ahora de forma remota por medio del Comando 4.1, se instaló el software **Asterisk**.

4.3.1 sudo -i

4.3.2 apt-get install asterisk (Comando 4.8)

Instala de forma automática el software **Asterisk** que más se adapta al **Sistema Operativo** en donde se ejecutará.

Verificando la versión de **Asterisk 13.18.3** instalada (Figura 4.12).

4.3.3 asterisk -V (Comando 4.9)

Informa de la versión instalada de **Asterisk**.

```
Comando 4.9
root@Asterisk-ISET:~# asterisk -V
Asterisk 13.18.3~dfsg-lubuntu4
root@Asterisk-ISET:~#
```

Versión de Asterisk

Figura 4.11 Versión de Asterisk.

Se activa el servicio Asterisk:

4.3.4 service asterisk start (Comando 4.10)

Inicia el servicio de **Asterisk** como **PBX**.

Asterisk puede contener las siguientes paqueterías⁷⁰.

4.3.5 apt-cache search asterisk (Comando 4.11)

- asterisk - Centralita privada Open Source (PBX)
- asterisk-config - Archivos de configuración para Asterisk
- asterisk-dahdi - DAHDI devices support for the Asterisk PBX
- asterisk-dev - Archivos de desarrollo de Asterisk
- asterisk-doc - Documentación del código fuente para Asterisk
- asterisk-espeak - Modulo eSpeak para Asterisk
- asterisk-flite - Módulo flite para Asterisk
- asterisk-mobile - Bluetooth phone support for the Asterisk PBX
- asterisk-modules - loadable modules for the Asterisk PBX
- asterisk-moh-opsound-g722 - asterisk extra sound files - English/g722
- asterisk-moh-opsound-gsm - asterisk extra sound files - English/gsm
- asterisk-moh-opsound-wav - asterisk extra sound files - English/wav
- asterisk-mp3 - MP3 playback support for the Asterisk PBX
- asterisk-mysql - MySQL database protocol support for the Asterisk PBX
- asterisk-ooH323 - H.323 protocol support for the Asterisk PBX - ooH323c
- asterisk-prompt-de - Mensajes en alemán para Asterisk PBX
- asterisk-prompt-fr-armelle - French voice prompts for Asterisk by Armelle Desjardins
- asterisk-prompt-fr-proformatique - French voice prompts for Asterisk
- asterisk-prompt-it - paquete vacío de transición
- asterisk-tests - internal test modules of the Asterisk PBX
- asterisk-voicemail - simple voicemail support for the Asterisk PBX
- asterisk-voicemail-imapstorage - IMAP voicemail storage support for the Asterisk PBX

⁷⁰ Las paqueterías enlistadas son mostradas por Asterisk al ejecutar el comando 4.11.

- asterisk-voicemail-odbcstorage - ODBC voicemail storage support for the Asterisk PBX
- asterisk-vpb - VoiceTronix devices support for the Asterisk PBX
- dahdi - utilities for using the DAHDI kernel modules
- dahdi-dkms - DAHDI telephony interface (dkms kernel driver)
- dahdi-linux - DAHDI telephony interface - Linux userspace parts
- libsipwitch1 - secure peer-to-peer SIP VoIP server - shared libraries
- libsipwitch1-dbg - secure peer-to-peer SIP VoIP server - debug symbols
- python-panoramisk-doc - asyncio based library to play with asterisk (doc)
- asterisk-core-sounds-en - asterisk PBX sound files - US English
- asterisk-core-sounds-en-g722 - asterisk PBX sound files - en-us/g722
- asterisk-core-sounds-en-gsm - asterisk PBX sound files - en-us/gsm
- asterisk-core-sounds-en-wav - asterisk PBX sound files - en-us/wav
- asterisk-core-sounds-es - asterisk PBX sound files – Spanish
- asterisk-core-sounds-es-g722 - asterisk PBX sound files - es-mx/g722
- asterisk-core-sounds-es-gsm - asterisk PBX sound files - es-mx/gsm
- asterisk-core-sounds-es-wav - asterisk PBX sound files - es-mx/wav
- asterisk-core-sounds-fr - asterisk PBX sound files - Canadian French
- asterisk-core-sounds-fr-g722 - asterisk PBX sound files - fr-ca/g722
- asterisk-core-sounds-fr-gsm - asterisk PBX sound files - fr-ca/gsm
- asterisk-core-sounds-fr-wav - asterisk PBX sound files - fr-ca/wav
- asterisk-core-sounds-it - asterisk PBX sound files – Italian
- asterisk-core-sounds-it-g722 - asterisk PBX sound files - it-it/g722
- asterisk-core-sounds-it-gsm - asterisk PBX sound files - it-it/gsm
- asterisk-core-sounds-it-wav - asterisk PBX sound files - it-it/wav
- asterisk-core-sounds-ru - asterisk PBX sound files – Russian
- asterisk-core-sounds-ru-g722 - asterisk PBX sound files - ru-ru/g722
- asterisk-core-sounds-ru-gsm - asterisk PBX sound files - ru-ru/gsm
- asterisk-core-sounds-ru-wav - asterisk PBX sound files - ru-ru/wav
- asterisk-opus - opus module for Asterisk
- asterisk-prompt-es-co - Colombian Spanish voice prompts for Asterisk
- asterisk-prompt-it-menardi - asterisk PBX Italian sound files
- asterisk-prompt-it-menardi-alaw - asterisk PBX Italian sound files - a-law prompts
- asterisk-prompt-it-menardi-gsm - asterisk PBX Italian sound files - gsm prompts
- asterisk-prompt-it-menardi-wav - asterisk PBX Italian sound files - wav prompts
- asterisk-testsuite - test suite for the Asterisk PBX
- asttest - test runner for the Asterisk test suite - dummy package
- braillegraph - simple histogram tool producing text dot-matrix graphs
- dahdi-source - DAHDI telephony interface - source code for kernel driver
- gastman - GUI tool for Asterisk administration and monitoring
- iaxmodem - software modem with IAX2 connectivity
- libasterisk-agi-perl - Collections of Perl modules to be used with Asterisk PBX AGI
- libnetsds-perl - Service Delivery Suite framework

- libopenr2-3 - MFC/R2 (telephony) call setup library
- libsipwitch-dev - secure peer-to-peer SIP VoIP server - development files
- libss7-2.0 - Signalling System 7 (ss7) library
- libss7-dev - Signalling System 7 (ss7) development files
- panoramisk - asyncio based library to play with asterisk (Python 3 binary)
- python-asterisk - Asterisk Manager API interface module for Python
- python-pyst - Python module for interacting with the Asterisk PBX
- python-starpy - Asterisk (AMI) protocols for Twisted Python
- python3-panoramisk - asyncio based library to play with asterisk (Python 3 library)
- ruby-ami - Ruby client library for the Asterisk Management Interface
- sipwitch - secure peer-to-peer VoIP server for the SIP protocol
- sipwitch-cgi - secure peer-to-peer SIP VoIP server - CGI XML-RPC interface
- asterisk-prompt-es - Español para Asterisk PBX
- dahdi-firmware-nonfree - firmware no libre DAHDI

De las cuales se instalan las siguientes, lo cuales son sonidos estándar en varios formatos, ya que de forma automática están en idioma inglés:

4.3.6 apt-get install asterisk-core-sounds-es (Comando 4.12)

Descarga e instala los sonidos en idioma español.

4.3.7 apt-get install asterisk-core-sounds-es-g722 (Comando 4.13)

Descarga e instala un conjunto de sonidos G.722.

4.3.8 apt-get install asterisk-core-sounds-es-gsm (Comando 4.14)

Descarga e instala un conjunto de sonidos GSM.

4.3.9 apt-get install asterisk-core-sounds-es-wav (Comando 4.15)

Descarga e instala un conjunto de sonidos WAV.

4.3.10 apt-get install asterisk-prompt-es (Comando 4.16)

Descarga e instala de audios con indicaciones en español para Asterisk PBX

Confirmando las paqueterías instaladas (Figura 4.13):

4.3.11 dpkg -l asterisk* (Comando 4.17)

Informa de las paqueterías instaladas por el usuario o de forma automática por el servidor como son las paqueterías en idioma inglés.

```

root@Asterisk-ISET:~# dpkg -l asterisk*
Deseado=desconocido(U)/Instalar/eliminar/Purgar/retener(H)
| Estado=No/Inst/ficheros-Conf/desempaquetado/medio-conf/medio-inst(H)/espera-disparo(W)/pendiente-disparo
|/ Err?=(ninguno)/requiere-Reinst (Estado,Err: mayúsc.=malo)
|/ Nombre Versión Arquitectura Descripción
+++
ii asterisk 1:13.18.3~dfsg-lubuntu4 armhf Open Source Private Branch Exchange (PBX)
un asterisk-lfb7f5c06d7a2052e38d021b3d8 <ninguna> <ninguna> (no hay ninguna descripción disponible)
ii asterisk-config 1:13.18.3~dfsg-lubuntu4 all Configuration files for Asterisk
un asterisk-config-custom <ninguna> <ninguna> (no hay ninguna descripción disponible)
ii asterisk-core-sounds-en 1.4.27-1 all asterisk PBX sound files - US English
un asterisk-core-sounds-en-g722 <ninguna> <ninguna> (no hay ninguna descripción disponible)
ii asterisk-core-sounds-en-gsm 1.4.27-1 all asterisk PBX sound files - en-us/gsm
un asterisk-core-sounds-en-wav <ninguna> <ninguna> (no hay ninguna descripción disponible)
ii asterisk-core-sounds-es 1.4.27-1 all asterisk PBX sound files - Spanish
ii asterisk-core-sounds-es-g722 1.4.27-1 all asterisk PBX sound files - es-mx/g722
ii asterisk-core-sounds-es-gsm 1.4.27-1 all asterisk PBX sound files - es-mx/gsm
ii asterisk-core-sounds-es-wav 1.4.27-1 all asterisk PBX sound files - es-mx/wav
un asterisk-dahdi <ninguna> <ninguna> (no hay ninguna descripción disponible)
un asterisk-dev <ninguna> <ninguna> (no hay ninguna descripción disponible)
un asterisk-doc <ninguna> <ninguna> (no hay ninguna descripción disponible)
ii asterisk-modules 1:13.18.3~dfsg-lubuntu4 armhf loadable modules for the Asterisk PBX
ii asterisk-moh-opsound-gsm 2.03-1 all asterisk extra sound files - English/gsm
un asterisk-oo323 <ninguna> <ninguna> (no hay ninguna descripción disponible)
un asterisk-opus <ninguna> <ninguna> (no hay ninguna descripción disponible)
un asterisk-prompt-en <ninguna> <ninguna> (no hay ninguna descripción disponible)
un asterisk-prompt-en-us <ninguna> <ninguna> (no hay ninguna descripción disponible)
ii asterisk-prompt-es 1.4-1 all Spanish prompts for the Asterisk PBX
un asterisk-prompt-es-mx <ninguna> <ninguna> (no hay ninguna descripción disponible)
un asterisk-sounds-main <ninguna> <ninguna> (no hay ninguna descripción disponible)
ii asterisk-voicemail 1:13.18.3~dfsg-lubuntu4 armhf simple voicemail support for the Asterisk PBX
un asterisk-voicemail-storage <ninguna> <ninguna> (no hay ninguna descripción disponible)
un asterisk-vpb <ninguna> <ninguna> (no hay ninguna descripción disponible)
root@Asterisk-ISET:~#

```

Figura 4.12 Paqueterías de Asterisk a instalar.

Se debe reiniciar el servicio de **Asterisk**

4.3.12 service asterisk restart

(Comando 4.18)

Se observa el estado del servicio de **Asterisk** (Figura 4.14).

4.3.13 service asterisk status

(Comando 4.19)

```

asterisk.service - Asterisk PBX
Loaded: loaded (/lib/systemd/system/asterisk.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2020-04-02 13:14:49 CST; 10mn ago
Docs: man:asterisk(8)
Main PID: 1153 (asterisk)
Tasks: 71 (limit: 2154)
CGroup: /system.slice/asterisk.service
└─1153 /usr/sbin/asterisk -g -f -U asterisk

abr 02 13:14:49 Aczel-Asterisk asterisk[1153]: [Apr  2 13:14:49] WARNING[1153]: cel_pgsql.c:460 process_my_load_module: CEL pgsql config file missing global section.
abr 02 13:14:49 Aczel-Asterisk asterisk[1153]: [Apr  2 13:14:49] NOTICE[1153]: cel_custom.c:97 load_config: No mappings found in cel_custom.conf. Not logging CEL to custom CSVs.
abr 02 13:14:49 Aczel-Asterisk asterisk[1153]: [Apr  2 13:14:49] NOTICE[1153]: cel_tds.c:452 tds_load_module: cel_tds has no global category, nothing to configure.
abr 02 13:14:49 Aczel-Asterisk asterisk[1153]: [Apr  2 13:14:49] WARNING[1153]: cel_tds.c:557 load_module: cel_tds module had config problems; declining load
abr 02 13:14:49 Aczel-Asterisk asterisk[1153]: radcli: rc_read_config: rc_read_config: can't open /etc/radiusclient-ng/radiusclient.conf: No such file or directory
abr 02 13:14:49 Aczel-Asterisk asterisk[1153]: [Apr  2 13:14:49] NOTICE[1153]: cdr_radius.c:278 load_module: Cannot load radiusclient-ng configuration file /etc/radiusclient-ng/radiusclient.conf.
abr 02 13:14:49 Aczel-Asterisk asterisk[1153]: [Apr  2 13:14:49] WARNING[1153]: channel.c:570 ast_channel_register: Already have a handler for type 'Console'
abr 02 13:14:49 Aczel-Asterisk asterisk[1153]: [Apr  2 13:14:49] ERROR[1153]: chan_oss.c:1515 load_module: Unable to register channel type 'OSS'
abr 02 13:14:49 Aczel-Asterisk systemd[1]: Started Asterisk PBX.
abr 02 13:15:06 Aczel-Asterisk systemd[1]: asterisk.service: Got notification message from PID 1298, but reception only permitted for main PID 1153

```

Error al encontrar el archivo: **radcli** en la dirección descrita

Figura 4.13 Error de ubicación del archivo radiusclient.conf.

4.4 Configuración Radius.

El sistema no es capaz de encontrar el archivo **radcli** en el apartado **radiusclient-ng** por lo que se debe buscar y enrutar la dirección en donde se encuentra, dentro de los archivos **cel.conf** y **cdr.conf**:

El comando **nano** es un editor de texto el cual permitirá la modificación de cada línea de texto en el archivo **cel.conf**:

4.4.1 nano /etc/asterisk/cel.conf (Comando 4.20)

Editor de texto básico y de uso sencillo.

Se elimina la dirección incorrecta (**radiuscfg=> /etc/radiusclient-ng/radiusclient.conf**) y se ingresa la siguiente línea de código:

4.4.1.1 radiuscfg => /etc/radcli/radiusclient.conf (Comando 4.21)

De la misma forma se debe ingresar al archivo **cdr.conf**:

4.4.2 nano /etc/asterisk/cdr.conf (Comando 4.22)

Nuevamente se elimina la dirección incorrecta (**radiuscfg=> /etc/radiusclient-ng/radiusclient.conf**) con la diferencia que se le agrega inicialmente el comando de (Figura 4.15):

4.4.2.1 [radius]

(Comando 4.23)

Seguido de la corrección de la ruta del archivo radiusclient.conf con el **Comando 4.21**.

4.4.2.2 radiuscfg => /etc/radcli/radiusclient.conf

```
[radius]
radiuscfg => /etc/radcli/radiusclient.conf
```

Figura 4.14 Enrutamiento correcto del archivo radiusclient.conf.

Reiniciar y observar el estado nuevamente del servicio **Asterisk**:

4.4.3 service asterisk restart

4.4.4 service asterisk status

Se observa que el servicio se encuentra de forma activo y sin ningún conflicto en el servidor (Figura 4.16), por lo que es posible registrar las extensiones de Asterisk.

```
● asterisk.service - Asterisk PBX
   Loaded: loaded (/lib/systemd/system/asterisk.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-04-27 14:42:45 CDT; 2s ago
     Docs: man:asterisk(8)
  Main PID: 2591 (asterisk)
    Tasks: 85 (limit: 2154)
   CGroup: /system.slice/asterisk.service
           └─2591 /usr/sbin/asterisk -g -f -U asterisk
             └─2659 /usr/bin/pulseaudio --start --log-target=syslog

abr 27 14:42:41 Asterisk-ISET asterisk[2591]: JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
abr 27 14:42:45 Asterisk-ISET asterisk[2591]: [Apr 27 14:42:45] NOTICE[2591]: confbridge/conf_config_parser.c:2095 verify_default_profiles: Adding default_menu menu to
abr 27 14:42:45 Asterisk-ISET asterisk[2591]: [Apr 27 14:42:45] NOTICE[2591]: cdr_pgsql.c:538 config_module: cdr_pgsql configuration contains no global section, skipping
abr 27 14:42:45 Asterisk-ISET asterisk[2591]: [Apr 27 14:42:45] WARNING[2591]: cel_pgsql.c:460 process_my_load_module: CEL pgsql config file missing global section.
abr 27 14:42:45 Asterisk-ISET asterisk[2591]: [Apr 27 14:42:45] NOTICE[2591]: cel_custom.c:97 load_config: No mappings found in cel_custom.conf. Not logging CEL to cust
abr 27 14:42:45 Asterisk-ISET asterisk[2591]: [Apr 27 14:42:45] NOTICE[2591]: cel_tds.c:452 tds_load_module: cel_tds has no global category, nothing to configure.
abr 27 14:42:45 Asterisk-ISET asterisk[2591]: [Apr 27 14:42:45] WARNING[2591]: cel_tds.c:557 load_module: cel_tds module had config problems; declining load
abr 27 14:42:45 Asterisk-ISET asterisk[2591]: [Apr 27 14:42:45] WARNING[2591]: channel.c:570 ast_channel_register: Already have a handler for type 'Console'
abr 27 14:42:45 Asterisk-ISET asterisk[2591]: [Apr 27 14:42:45] ERROR[2591]: chan_oss.c:1515 load_module: Unable to register channel type 'OSS'
abr 27 14:42:45 Asterisk-ISET systemd[1]: Started Asterisk PBX.
```

Figura 4.15 Estado activo y correcto de Asterisk.

4.5 Extensiones de llamadas y mensajería.

A continuación se presentan los directorios creados por Asterisk (Figura 4.17).

```
root@Asterisk-ISET:~# ls /etc/asterisk/
acl.conf          cdr_custom.conf  cli_aliases.conf  features.conf     motif.conf        res_corosync.conf  sorcery.conf
adsi.conf         cdr_manager.conf cli.conf          festival.conf     musiconhold.conf  res_curl.conf      ss7.timers
agents.conf       cdr_mysql.conf   cli_permissions.conf followme.conf     muted.conf        res_fax.conf       stasis.conf
alarmreceiver.conf cdr_odbc.conf    codecs.conf       func_odbc.conf   ooh323.conf       res_ldap.conf      statsd.conf
alsa.conf         cdr_pgsql.conf   confbridge.conf   hep.conf          osp.conf          res_odbc.conf      telcordia-l.adsi
amd.conf          cdr_sqlite3_custom.conf config_test.conf  http.conf         oss.conf          res_parking.conf   test_sorcery.conf
app_mysql.conf    cdr_syslog.conf  console.conf      iax.conf          phone.conf        res_pgsql.conf     udptl.conf
app_skel.conf     cdr_tds.conf     dbsep.conf        iaxprov.conf     phoneprov.conf    res_pktccops.conf  unistim.conf
ari.conf          cel.conf         dnsmgr.conf       indications.conf pjproject.conf     res_snmp.conf      users.conf
ast_debug_tools.conf cel.conf.copia   dsp.conf          logger.conf       pjsip.conf        res_stun_monitor.conf voicemail.conf
asterisk.adsi     cel_custom.conf dundi.conf        manager.conf      pjsip_notify.conf rtp.conf           vpb.conf
asterisk.conf     cel_odbc.conf    enum.conf          manager.d          pjsip_wizard.conf say.conf           xmpp.conf
calendar.conf     cel_pgsql.conf   extconfig.conf    meetme.conf       queuerules.conf   sip.conf
ccss.conf         cel_sqlite3_custom.conf extensions.ael     mgcp.conf         queues.conf        sip_notify.conf
cdr_adaptive_odbc.conf cel_tds.conf     extensions.conf   minivm.conf       res_config_mysql.conf skinny.conf
cdr.conf          chan_dahdi.conf  extensions.lua     misdn.conf        res_config_sqlite3.conf sla.conf
cdr.conf.copia    chan_mobile.conf extensions_minivm.conf modules.conf       res_config_sqlite.conf smdi.conf
root@Asterisk-ISET:~#
```

Figura 4.16 Directorios de Asterisk.

Para agregar las extensiones se utilizan los directorios:

- **sip.conf**
- **extensions.conf.**

4.5.1 Directorio sip.conf

Dentro del directorio sip.conf se agregan las siguientes líneas de comando para el servicio de llamadas y el de mensajes:

4.5.1.1 vi /etc/asterisk/sip.conf (Comando 4.24)

Vi es otro editor de texto al igual que Nano, con la diferencia que nos permitirá delimitar los caracteres que deseamos ver para la configuración, ya que el archivo sip.conf cuenta con 1595 líneas en código, del cual 98% de texto son ejemplos y explicaciones de los diferentes comandos.

4.5.1.1.1 g/^\s*/d (Comando 4.25)

Borra todas las líneas comentadas, las cuales inician con el carácter de punto y coma (;)

4.5.1.1.2 g/^\\$/d (Comando 4.26)

Borra todas las líneas vacías que inician con el carácter de dinero (\$)

Para editar el texto se usó de nuevo el editor de texto Nano.

4.5.1.2 nano /etc/asterisk/sip.conf (Comando 4.27)

- **Servicio de llamadas.**

Para el servicio de llamadas es necesario escribir las siguientes líneas de código (Figura 4.18):

1. **qualify = yes** (Comando 4.28)
2. **language = es** (Comando 4.29)
3. **disallow = all** (Comando 4.30)
4. **allow = ulaw,alaw** (Comando 4.31)

```
qualify=yes           ;Monitoreo de conexión
language=es           ;Idioma del servicio
disallow=all          ;Desactiva todos los codificadores
allow=ulaw,alaw       ;Codificación en orden prederencial
```

Figura 4.17 Comando de configuración de llamadas en el archivo sip.conf.

- **Servicio de mensajería.**

Para el servicio de mensajería instantánea es necesario escribir las siguientes líneas de código (Figura 4.19):

1. **accept_outofcall_message = yes** (Comando 4.32)
2. **outofcall_message_context = Nombre del contexto-messages** (Comando 4.33)
3. **auth_message_requests = yes** (Comando 4.34)
4. **subscribecontext = Nombre del contexto-messages2**(Comando 4.35)

```
accept_outofcall_message=yes           ;Mensajes instantáneos
outofcall_message_context=messages     ;Contexto del dialplan del envío de mensajes
auth_message_requests=yes              ;Extensión de mensajería
subscribecontext=subscribe              ;Contexto de configuración
```

Figura 4.18 Comando en el archivo sip.conf para mensajería.

Para evitar que el archivo vuelva a crecer de forma significativa, se creó una plantilla (Figura 4.20), la cual ayudara a ingresar menos cantidad de comandos.

- **Plantilla.**

5. **[usuario](!)** (Comando 4.36)
6. **type = friend** (Comando 4.37)
7. **host = dynamic** (Comando 4.38)
8. **dtmfmode=rfc2833** (Comando 4.39)
9. **context = Nombre del contexto** (Comando 4.40)

```
[usuario](!)          ;Plantilla
type=friend           ;Llamar o recibir llamadas
host=dynamic          ;Registro de cualquier IP
dtmfmode=rfc2833     ;Envía tonos DTMF como paquetes RTP codificados
context=RedAsterisk  ;Contexto
```

Figura 4.19 Plantilla con el tipo de servicio de llamada, tipo de host, tipo de tono y el contexto para mensajería.

- **Extensiones.**

Se agregan las siguientes líneas (Figura 4.21):

10. [Número de extensión](usuario) (Comando 4.41)

11. username = Nombre del usuario (Comando 4.42)

12. secret = Contraseña (Comando 4.43)

13. port = 5061 (Solo si el servidor cuenta con una extensión)
(Comando 4.44)

```
;Extensión 101 Alumno A
[101](usuario)          ;Extensión 101 y plantilla
username=Alumno A      ;Nombre de usuario
secret=s1234           ;Contraseña del usuario
disallow=all           ;Desactiva todos los codificadores

;Extensión 102 Alumno B
[102](usuario)          ;Extensión 102 y platilla
username=Alumno B      ;Nombre de usuario
secret=s1234           ;Contraseña del usuario
disallow=all           ;Desactiva todos los codificadores

;Extensión 103 Alumno C
[103](usuario)          ;Extensión 102 y platilla
username=Alumno C      ;Nombre de usuario
secret=s1234           ;Contraseña del usuario
disallow=all           ;Desactiva todos los codificadores
```

Figura 4.20 Registro de Extensiones en el archivo sip.conf.

Se repiten las líneas de código anteriores las veces necesarias para la adición de nuevas extensiones, cambiando el **Número de extensión** y **Nombre de usuario**.

Para la verificación de las extensiones (Figura 4.22) es necesario entrar a la consola de comandos de **Asterisk**.

4.5.1.3 asterisk -rvvvvv (Comando 4.45)

```

root@Asterisk-ISET:~# asterisk -rvvvvv
Asterisk 13.18.3~dfsg-lubuntu4, Copyright (C) 1999 - 2014, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 13.18.3~dfsg-lubuntu4 currently running on Asterisk-ISET (pid
= 2591)
Asterisk-ISET*CLI> █

```

Figura 4.21 Consola de Asterisk.

Por medio de consola se puede también apreciar la información como el número de extensión y contraseña (Figura 4.23):

4.5.1.4 sip show users (Comando 4.46)

```

Asterisk-ISET*CLI> sip show users
Username          Secret          Accountcode     Def.Context     ACL Forcerport
101               sl234           101             RedAsterisk     No No
102               sl234           102             RedAsterisk     No No
103               sl234           103             RedAsterisk     No No
Asterisk-ISET*CLI> █

```

Figura 4.22. Información de consola de Asterisk.

Por medio del siguiente comando se puede comprobar extensión, nombre de la extensión registrada, dirección IP, puerto y estado del dispositivo (Figura 4.24):

4.5.1.5 sip show peers (Comando 4.47)

```

Asterisk-ISET*CLI> sip show peers
Name/username     Host              Dyn Forcerport Comedia  ACL Port  Status  Description
101/Alumno A     (Unspecified)    D Auto (No) No      0        UNKNOWN
102/Alumno B     (Unspecified)    D Auto (No) No      0        UNKNOWN
103/Alumno C     (Unspecified)    D Auto (No) No      0        UNKNOWN
3 sip peers [Monitored: 0 online, 3 offline Unmonitored: 0 online, 0 offline]
Asterisk-ISET*CLI> █

```

Figura 4.23 Registro de las Extensiones en consola de Asterisk.

4.5.2 Directorio extensions.conf

Por otro lado, en el archivo **extensions.conf** se debe agregar (Figura 4.25):

4.5.2.1 nano /etc/asterisk/extensions.conf (Comando 4.48)

- **Las extensiones para llamadas.**
 1. **[Nombre del contexto]** (Comando 4.49)
 2. **exten => Número de extensión,1,Dial(SIP/Número de extensión)** (Comando 4.50)

```
[RedAsterisk]
exten => 101,1,Dial(SIP/101)
exten => 102,1,Dial(SIP/102)
exten => 103,1,Dial(SIP/103)
```

Figura 4.24 Extensiones para llamadas en el archivo extensions.conf.

El punto 2 se repite las veces necesarias para las diferentes extensiones agregadas, cambiando el **Número de extensión** (Figura 4.26).

- Las extensiones para mensajería.
 3. [**Nombre del contexto-messages2**] (Comando 4.51)
 4. **exten => Número de extensión, hint, SIP/ Número de extensión** (Comando 4.52)

```
[subscribe]
exten => 101, hint, SIP/101
exten => 102, hint, SIP/102
exten => 103, hint, SIP/103
```

Figura 4.25 Extensiones para mensajes en el archivo extensions.conf.

De la misma forma se repite las veces necesarias para las diferentes extensiones, cambiando el **Número de extensión**.

- Las extensiones para mensajería (Figura 4.27)
 5. [**Nombre del contexto-messages**] (Comando 5.53)
Nombre que llevará el contexto de mensajes.
 6. **exten => _X.,1,Noop(Mensaje de \${MESSAGE(from)})** (Comando 5.54)
Muestra en consola la Extensión quien ha enviado un mensaje.
 7. **same => n,Noop(Mensaje para \${MESSAGE(to)})** (Comando 5.55)
Muestra en consola la Extensión a quien es enviado el mensaje.
 8. **same => n,Noop(Texto = \${MESSAGE(body)})** (Comando 5.56)
Muestra en consola el contenido del mensaje.
 9. **same => n,Messagesend(sip:\${EXTEN},\${MESSAGE(from)})**
Envía el mensaje de la Extensión origen al destinatario. (Comando 5.57)
 10. **same => n,Noop(Estado del mensaje \${MESSAGE_SEND_STATUS})**
Muestra en consola el estado del mensaje enviado. (Comando 5.58)
 11. **same => n,Hangup** (Comando 5.59)

```

[messages]
exten => _X.,1,Noop(Mensaje de ${MESSAGE(from)})
same => n,Noop(Mensaje para ${MESSAGE(to)})
same => n,Noop(Texto = ${MESSAGE(body)})
same => n,Messagesend(sip:${EXTEN},${MESSAGE(from)})
same => n,Noop(Estado del mensaje ${MESSAGE_SEND_STATUS})
same => n,Hangup

```

Figura 4.26 Extensiones de mensajería con serie consecutiva de acciones.

4.6 Aplicación cliente.

4.6.1 Descripción de Zoiper.

Es un **Softphone** personalizado que se ejecuta en una multitud de plataformas basado en el protocolo **SIP**, tales como:

- macOS.
- Linux
- Windows
- iOS
- Android
- Navegador de internet

Con una interfaz intuitiva es capaz de realizar las funciones de:

- Llamadas
- Video
- Fax
- Mensajería instantánea

Este **Softphone** no se basa en programación Java, Flash o .NET, por lo que está escrito y programado en C/C++, por esta razón utiliza poca memoria y CPU.

4.6.2 Registro

Para instalar la aplicación **Zoiper** en una laptop y en 2 Smartphones, las extensiones son (Figura 4.28).

- Smartphone 1: 101
- Smartphone 2: 102
 1. **Username@PBX/VoIP:** Número de Extensión @ IP Servidor Asterisk
 2. **Password:** Contraseña del usuario

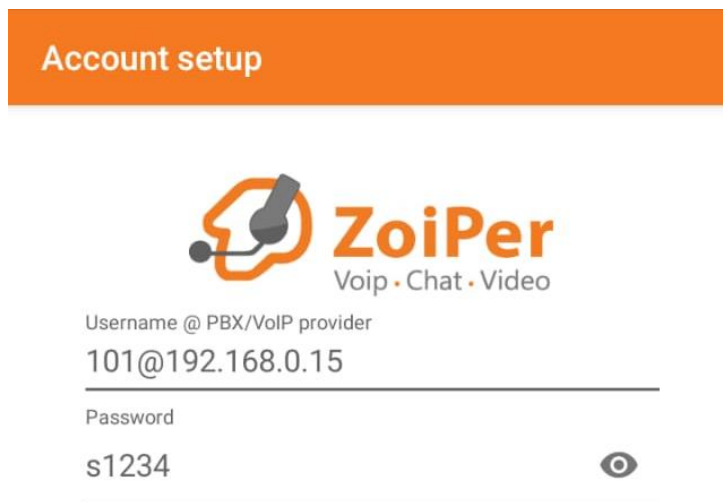


Figura 4.27 Registro y petición de la extensión en Smartphone.

3. Hostname or provider: IP Servidor Asterisk (Figura 4.29).

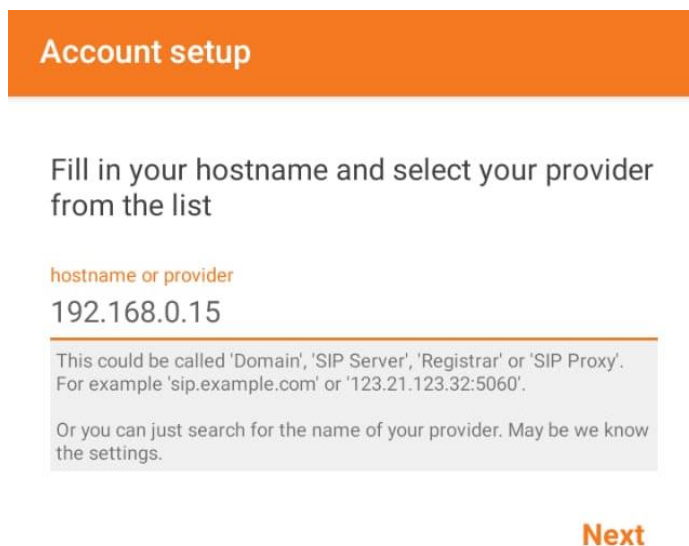


Figura 4.28 Dirección IP del servidor en Smartphone.

4. Authentication username: Número de Extensión (Figura 4.30 y Figura 4.31).



Figura 4.29 Confirmación de la extensión y de la dirección IP como servidor en Smartphone.

Account setup

Please choose between the following configurations

<input type="radio"/> SIP TLS	Not found
<input type="radio"/> SIP TCP	Not found
<input checked="" type="radio"/> SIP UDP	Found
<input type="radio"/> IAX UDP	Not found

Figura 4.30 Protocolo de envío de datos usado: UDP.

En la Consola de Asterisk se indica el registro correcto del dispositivo (Figura 4.32):

```
-- Unregistered SIP '101'
-- Registered SIP '101' at 192.168.0.6:55163
```

Figura 4.31 Registro de la Extensión101 en consola de Asterisk.

- Laptop: 103 (Figura 4.33).
 1. **Username@PBX/VoIP:** Número de Extensión @ IP Servidor Asterisk
 2. **Password:** Contraseña del usuario

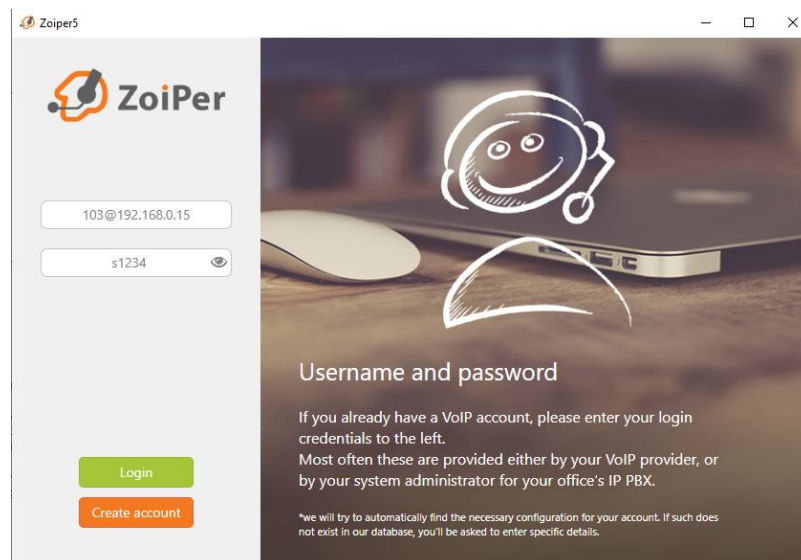


Figura 4.32 Registro y petición de la extensión en Laptop.

3. Hostname or provider: IP Servidor Asterisk (Figura 4.34).

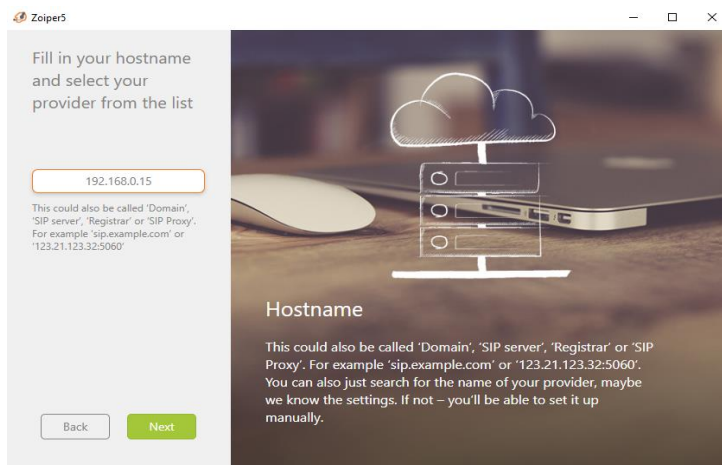


Figura 4.33 Dirección IP del servidor en Laptop.

4. Authentication username: Número de Extensión (Figura 4.35 y Figura 4.36).

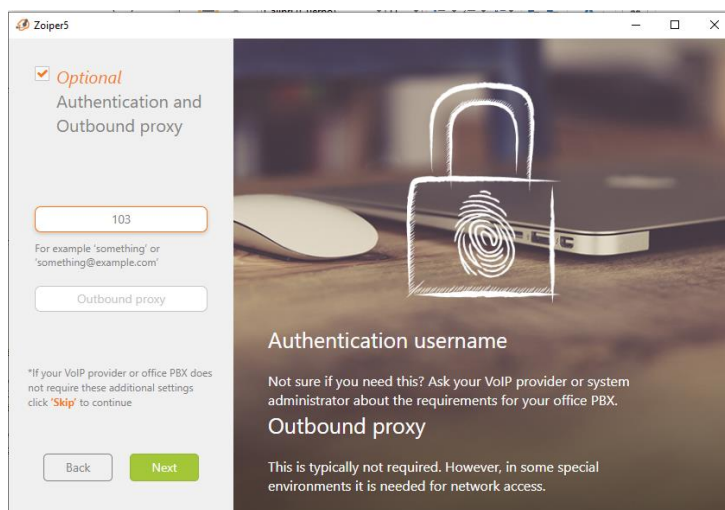


Figura 4.34 Confirmación de la extensión y de la dirección IP como servidor en Laptop.

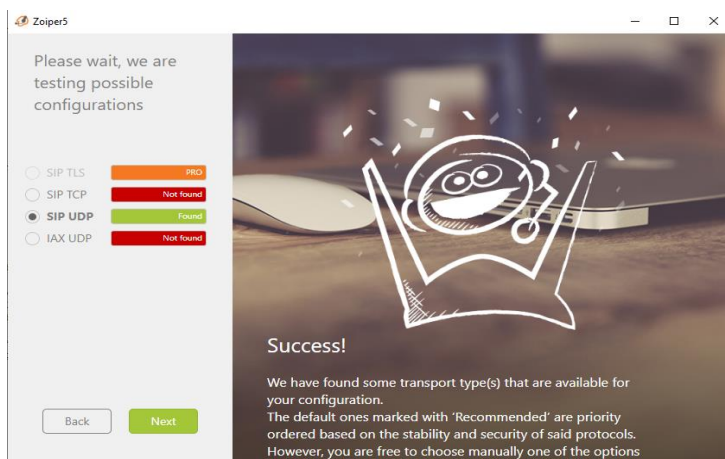


Figura 4.35 Protocolo de envío de datos usado: UDP.

En la Consola de Asterisk se indica el registro correcto del dispositivo (Figura 4.37):

```
-- Unregistered SIP '103'  
-- Registered SIP '103' at 192.168.0.13:37438
```

Figura 4.36 Registro de la Extensión 103 en consola de Asterisk.

Verificando que los registros estén correctos (Figura 4.38).

1. sip show peers

```
Asterisk-ISET*CLI> sip show peers  
Name/username      Host                Dyn Forcerport Comedia  ACL Port  Status  Description  
101/Alumno A       192.168.0.6        D Auto (No) No      55163    OK (10 ms)  
102/Alumno B       192.168.0.9        D Auto (No) No      39175    OK (7 ms)  
103/Alumno C       192.168.0.13       D Auto (No) No      37438    OK (12 ms)  
3 sip peers [Monitored: 3 online, 0 offline Unmonitored: 0 online, 0 offline]  
Asterisk-ISET*CLI>
```

Figura 4.37 Comprobación del registro correcto de Extensiones en consola de Asterisk.

4.6.3 Llamadas y mensajes

- Realizando una llamada de la extensión (Smartphone 1) 101 a (Smartphone 2) 102 (Figura 4.39).

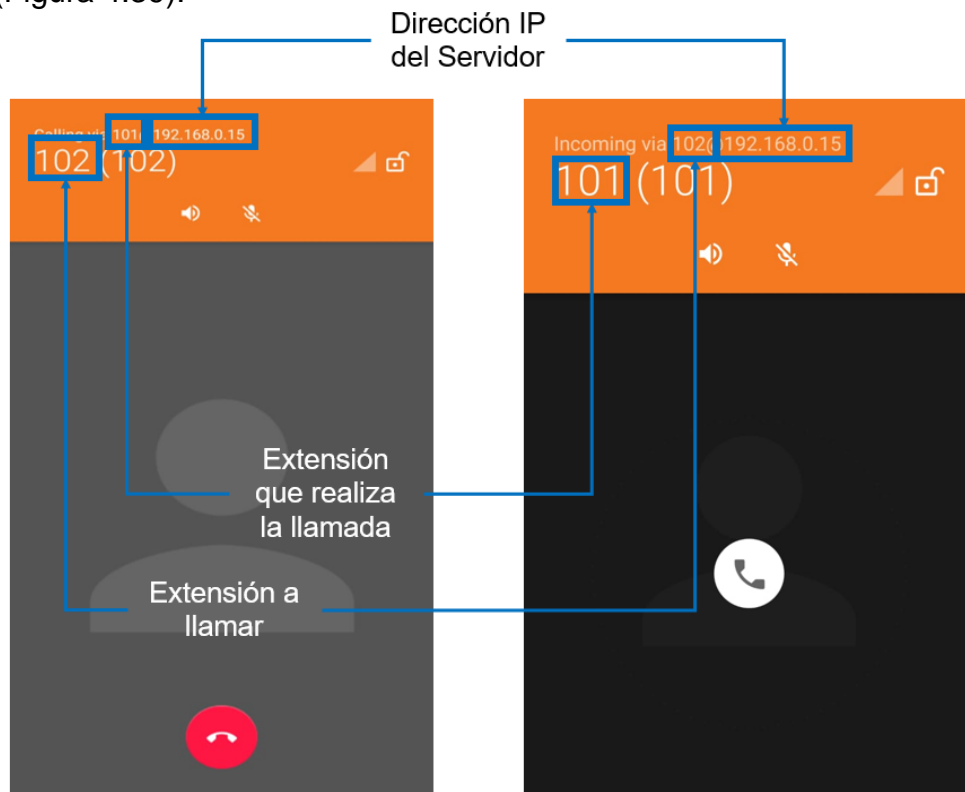


Figura 4.38 Extensión 101 realizando una llamada a la extensión 102.

Una vez aceptando la llamada en la Extensión 102 (Figura 4.40):

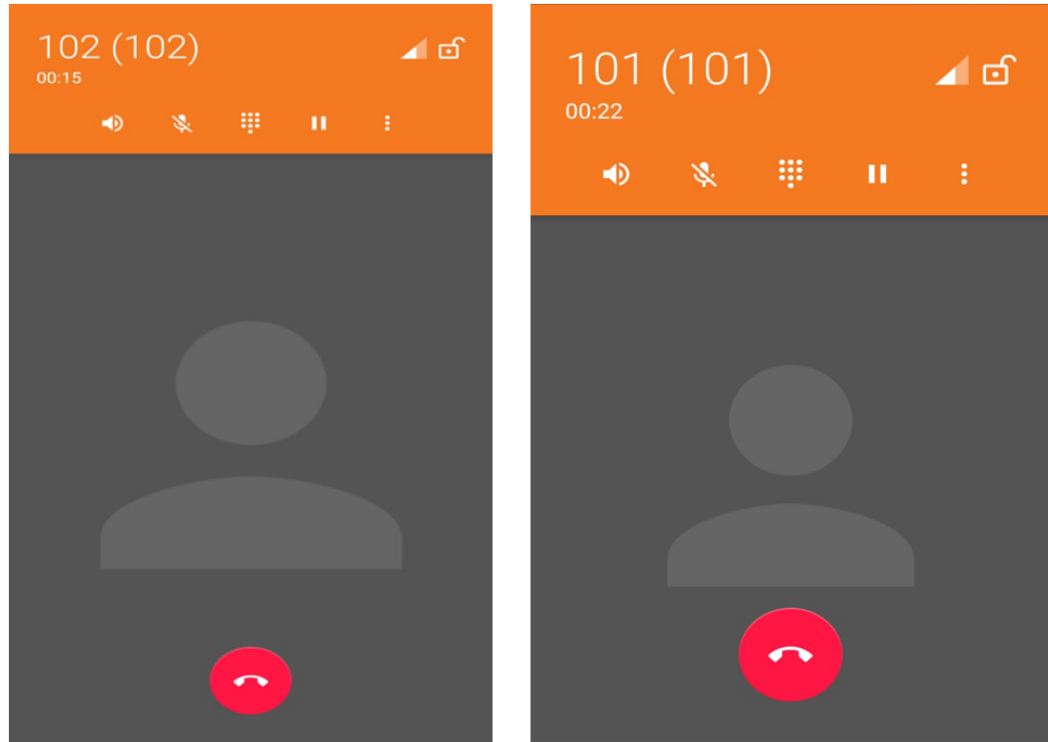


Figura 4.39 Llamada en curso entre las Extensiones 101 y 102.

- Llamada de la extensión 103 a 101 (Figura 4.41).

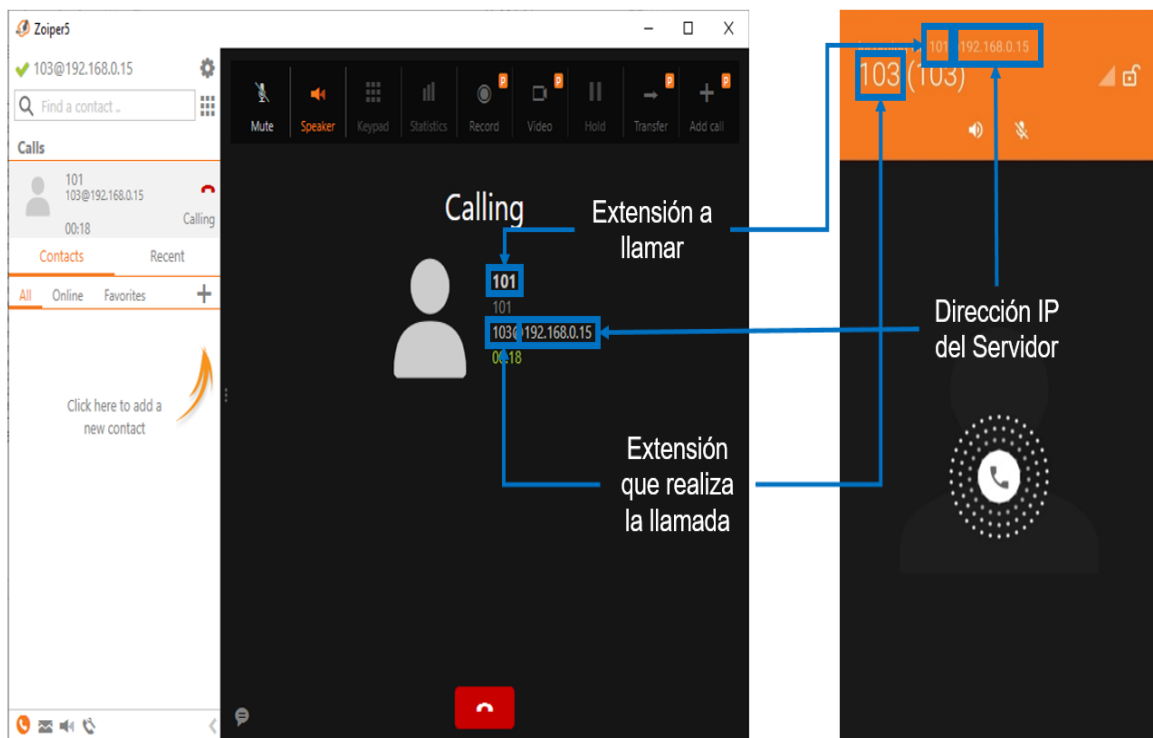


Figura 4.40 Extensión 103 realizando una llamada a la extensión 101.

Una vez aceptada la llamada en la Extensión 102 (Figura 4.42):

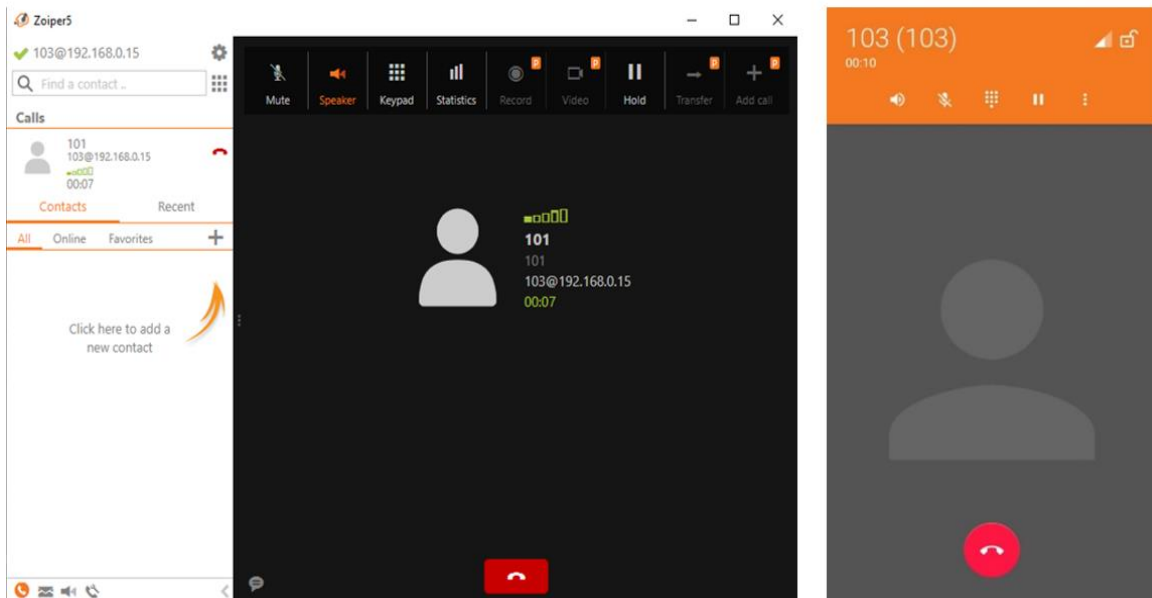


Figura 4.41 Llamada en curso entre las Extensiones 101 y 103.

Consola de Asterisk

En la consola de Asterisk se mostrarán las diferentes acciones y comunicaciones que se realizan en el servidor, en la Figura 4.41 se observa la secuencia de acciones que suceden al momento de que la terminal 102 rechaza la llamada entrante de la extensión 101 (Figura 4.43).

```

Connected to Asterisk 13.18.3~dfsg-lubuntu4 currently running on Asterisk-ISET (pid = 3431)
== Using SIP RTP CoS mark 5
> 0x6b602500 -- Strict RTP learning after remote address set to: 192.168.0.6:48424
-- Executing [102@RedAsterisk:1] Dial('SIP/101-00000002', 'SIP/102') in new stack
== Using SIP RTP CoS mark 5
-- Called SIP/102
-- SIP/102-00000003 is ringing
-- SIP/102-00000003 is ringing
-- Got SIP response 486 "Busy Here" back from 192.168.0.9:39175
-- SIP/102-00000003 is busy
== Everyone is busy/congested at this time (1:1/0/0)
-- Auto fallthrough, channel 'SIP/101-00000002' status is 'BUSY'
Asterisk-ISET*CLI>
  
```

Figura 4.42 Información de acciones en la Consola de Asterisk de una llamada desviada.

- Mensajes entre las extensiones 101 a 102 y 103 (Figura 4.44).

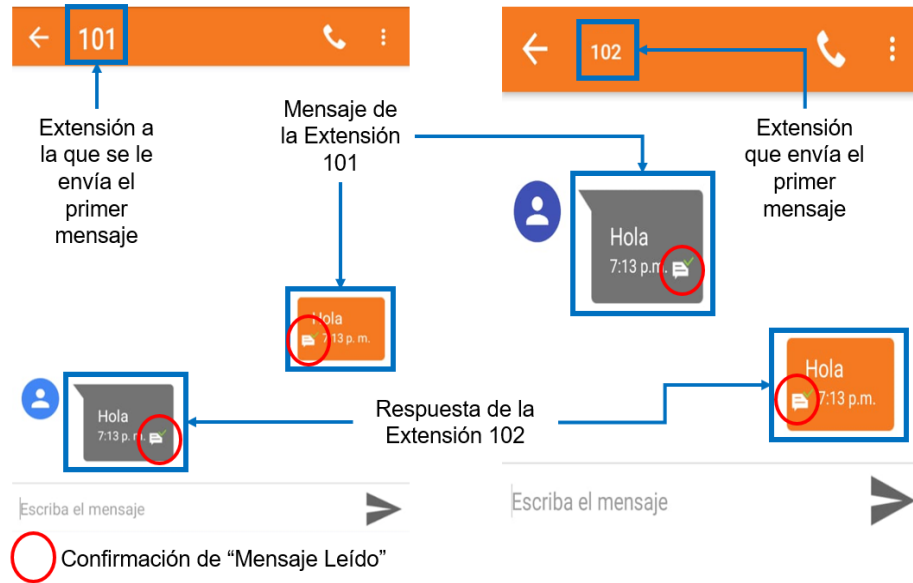


Figura 4.43 Mensajes entre las Extensiones 101 y 102.

En Consola de Asterisk se aprecian las secuencias de acciones que toma el servidor durante el intercambio de mensajes entre las Extensiones 101 y 102 (Figura 4.45).

```

Connected to Asterisk 13.18.3-dfsq-lubuntu4 currently running on Asterisk-ISET (pid = 3431)
-- Executing [102@messages:1] NoOP "Message/ast_msg_queue", "Mensaje de = < sip:101@192.168.0.15;transport=UDP>" in new stack
-- Executing [102@messages:2] NoOP "Message/ast_msg_queue", "Mensaje para = sip:102@192.168.0.15;transport=UDP" in new stack
-- Executing [102@messages:3] NoOP "Message/ast_msg_queue", "Texto = Hola" in new stack
-- Executing [102@messages:4] MessageSend "Message/ast_msg_queue", "sip:102,<sip:101@192.168.0.15;transport=UDP>" in new stack
-- Executing [102@messages:5] NoOP "Message/ast_msg_queue", "Estado del mensaje SUCCESS" in new stack
-- Executing [102@messages:6] Hangup "Message/ast_msg_queue", "" in new stack
== Spawn extension (messages, 102, 6) exited non-zero on 'Message/ast_msg_queue'
-- Executing [101@messages:1] NoOP "Message/ast_msg_queue", "Mensaje de = < sip:102@192.168.0.15;transport=UDP>" in new stack
-- Executing [101@messages:2] NoOP "Message/ast_msg_queue", "Mensaje para = sip:101@192.168.0.15;transport=UDP" in new stack
-- Executing [101@messages:3] NoOP "Message/ast_msg_queue", "Texto = Hola" in new stack
-- Executing [101@messages:4] MessageSend "Message/ast_msg_queue", "sip:101,<sip:102@192.168.0.15;transport=UDP>" in new stack
-- Executing [101@messages:5] NoOP "Message/ast_msg_queue", "Estado del mensaje SUCCESS" in new stack
-- Executing [101@messages:6] Hangup "Message/ast_msg_queue", "" in new stack
== Spawn extension (messages, 101, 6) exited non-zero on 'Message/ast_msg_queue'
Asterisk-ISET*CLI>

```

El código de consola de Asterisk muestra las acciones ejecutadas durante el intercambio de mensajes. Las acciones clave son: NoOP (para inicialización), MessageSend (para envío de mensajes) y Hangup (para finalización). El estado del mensaje enviado es SUCCESS.

Figura 4.44 Información de acciones en la Consola de Asterisk en el envío de mensajes.

4.7 Administración.

El administrador deberá abrir los siguientes archivos en la terminal del sistema.

- **Sip**
- **Extensions**

4.7.1 SIP

El archivo **sip.conf** sirve para configurar todo lo relacionado con el **Protocolo de Inicio de Sesión SIP** (Session Initiation Protocol) y añadir nuevos usuarios o conectar con proveedores **SIP**.

Se debe verificar la existencia de los siguientes comandos:

[general]

1. **context=public**

- El contexto es la unidad que organiza de manera básica el Plan de Mercado (DialPlan).
- Mantiene diferentes secciones del plan de mercado independientes entre sí.
- Separa la funcionalidad y las características, impone límites de seguridad entre las diversas partes del plan de mercado, así como proporciona diferentes clases de servicio a grupos de usuarios.
- Contexto predeterminado para llamadas entrantes

2. **allowverlap=no**

- Deshabilita el soporte de marcado superpuesto.
- Habilita / deshabilita el soporte de marcado superpuesto. Predeterminado sí (el dial de superposición proporciona un período de tiempo de espera más prolongado entre los dígitos, también llamado temporizador entre dígitos. Con el dial de superposición desactivado, la puerta de enlace espera recibir los dígitos uno tras otro entrando en esta línea con pequeño retraso entre dígitos. Con el dial de superposición activado, el dispositivo espera hasta unos 2 segundos entre dígitos).

3. **udpbindaddr=0.0.0.0**

- Dirección IP para vincular el socket de escucha UDP (0.0.0.0 se vincula a todos)

4. **tcpenable=no**

- Habilitar el servidor para las conexiones TCP entrantes (el valor predeterminado es no)

5. **tcpbindaddr=0.0.0.0**

- Dirección IP para que el servidor TCP se una (0.0.0.0 se une a todas las interfaces)

6. transport=udp

- Establece los transportes predeterminados. El orden determina el transporte predeterminado primario.

7. srvlookup=yes

- Habilita las búsquedas de **DNS SRV** en las llamadas salientes

8. [authentication]

9. [basic-options](!)

dtmfmode=rfc2833

context=office

type=friend

Plantilla

10. [natted-phone](!,basic-options)

directmedia=no

host=dynamic

Plantilla con opciones básicas

11. [public-phone] (!,basic-options)

directmedia=yes

Plantilla con opciones básicas

12. [my-codecs](!)

disallow=all

allow=ilbc

allow=g729

allow=gsm

allow=g723

allow=ulaw

Plantilla de Códec's

13. [ulaw-phone](!)

disallow=all

allow=ulaw

Plantilla para ulaw

También se deberán agregar los siguientes comandos:

1. qualify = yes

- Al estar activado el tiempo predeterminado es de 2 segundos en la verificación de que los dispositivos con una extensión estén en línea.
- Se puede especificar **qualify = XXX**, donde **XXX** es el tiempo deseado en milisegundos

2. language = es

- Cambio del idioma del inglés (en) por defecto al español (es).
- Cuenta con los idiomas: inglés, español, francés, italiano y ruso.

3. **disallow = all**

- Desactiva todos los Códec's de la configuración global.

4. **allow = ulaw,alaw**

- Ordena a todos los Códec's en orden preferencial.

5. **accept_outofcall_message = yes**

- Habilita el envío de mensajes instantáneos.

6. **outofcall_message_context = Nombre del contexto-messages**

- Contexto donde se procesaran los mensajes enviados.
- *Nombre del contexto-messages*: puede definirse con el nombre que se desee.

7. **auth_message_requests = yes**

- Autenticación de la extensión por Asterisk de los mensajes enviados.

8. **subscribecontext = Nombre del contexto-messages2**

- Contexto de configuración de extensiones hint, permitiendo a las extensiones conocer el estado de las demás.

Y finalmente las extensiones:

9. **[Número o nombre de extensión]**

- En **Número o nombre de extensión** pueden ser letras o números, dado que en el sistema de Asterisk es la etiqueta que identifica al dispositivo que este registrado con el **Número o nombre de extensión**.

10. **type = friend**

- Existen tres diferentes usos para el comando type (Tabla 4.1):

1. type = user	Permite llamadas salientes desde la extensión hacia Asterisk pero no de forma invertida. Debido a que no es posible indicar la variable host, así, Asterisk puede recibir llamadas pero no enviar llamadas al desconocer la dirección IP.
2. type = peer	Solamente permite recibir llamadas entrantes.
3. type = friend	Es una combinación de type = user y type = peer, por lo que las extensiones definidas de esta manera pueden realizar o recibir llamadas desde la terminal.

Tabla 4.1 Tipos de comando para type.

11. **host = dynamic**

- Tipo de host
- Dynamic: Indica que la extensión puede concretarse a la red desde cualquier dirección IP.
- Static: No es necesario adjuntar una contraseña.

12. dtmfmode=rfc2833

- Envía los tonos **DTMF** (Dual-Tone Multi-Frequency) por separado como **RTP** (Protocolo de transporte en tiempo real) codificados, distintos de los paquetes de audio, pero dentro de la misma red.

13. context = *Nombre del contexto*

- Nombre del contexto asignado.
- #### 14. username = *Nombre del usuario*
- Nombre del usuario al que se le asigna la extensión.

15. secret = *Contraseña*

- Asignada por el usuario y validada por el administrador.
- Puede ingresar letras y números.

4.7.2 Extensions.conf

Extensions.conf es el archivo más importante dentro **Asterisk** y tiene como finalidad definir el dialplan o plan de numeración de la central para cada contexto y para cada usuario.

El archivo **extensions.conf** se compone de contextos que se declaran dentro de corchetes []. Existen dos contextos que son [general] y [globals]

Todas las líneas de un determinado contexto tienen el mismo formato:

exten => extensión , prioridad, Comando(parametros)

- **Extensión:** Número marcado.
- **Prioridad:** Orden de ejecución de las acciones.
- **Comando:** Acción a ejecutar.

Declaración de extensiones para llamadas

12. [*Nombre del contexto*]

- Nombre de contexto indicado en el archivo sip para las extensiones asignadas para llamadas.

13. exten => *Número de extensión*,1,Dial(*SIP/Número de extensión*)

- **Exten:** Define la extensión.
- **Número de extensión:** Extensión asignada al usuario.
- **1:** Acción jerárquica.
- **Dial(SIP/Número de extensión):** El servidor realiza la llamada por el protocolo SIP

Declaración de extensiones para mensajes

1. [***Nombre del contexto-messages2***]
 - **Nombre de contexto** indicado en el archivo sip para las extensiones asignadas para mensajes.
2. **exten => Número de extensión, hint, SIP/ Número de extensión**
 - **Exten:** Define a la extensión.
 - **Número de extensión:** Extensión asignada al usuario.
 - **Hint:** Orden preferencial.
 - **SIP/Número de extensión:** Estándar de Internet para voz en tiempo real.

Registro en consola de Asterisk

1. [***Nombre del contexto-messages***]
 - Nombre de contexto para consola de **Asterisk**.
2. **exten => _X.,1,Noop(Mensaje de \${MESSAGE(from)})**
 - **_X:** Todas las extensiones.
 - **1:** Orden de prioridad
 - **Noop(Mensaje de \${MESSAGE(from)}):** Muestra en la consola de Asterisk la extensión que envía el mensaje.
3. **same => n,Noop(Mensaje para \${MESSAGE(to)})**
 - **Same:**
 - **n:** Orden de prioridad
 - **Noop(Mensaje para \${MESSAGE(to)}):** Muestra en la consola de Asterisk la extensión que recibe el mensaje.
4. **same => n,Noop(Texto = \${MESSAGE(body)})**
 - **Noop(Texto = \${MESSAGE(body)}):** Muestra en la consola de Asterisk el contenido del mensaje que ha sido enviado de una extensión a otra.
5. **same => n,Messagesend(sip:\${EXTEN},\${MESSAGE(from)})**
 - **Messagesend(sip:\${EXTEN},\${MESSAGE(from)}):**
6. **same => n,Noop(Estado del mensaje \${MESSAGE_SEND_STATUS})**
 - **Noop(Estado del mensaje \${MESSAGE_SEND_STATUS}):** Muestra en la consola de Asterisk el estado del mensaje.
7. **same => n,Hangup**
 - **Hangup:** Finaliza el servicio en curso.

4.8 Manual de usuario.

El usuario deberá descargar un **Softphone** dentro de su terminal, ya sea un teléfono móvil con un sistema operativo **Android** o **iOS** o bien una computadora.

Para los teléfonos móviles con Android la aplicación deberá ser buscada en la **Play Store**, para **iOS** deberá estar en **AppStore** la aplicación, y finalmente para una computadora deberá descargar el Softphone de la página oficial del desarrollador.

La aplicación a buscar es: **Zoiper IAX SIP VOIP Softphone**.

4.8.1 Configuración de Cliente ZoiPer.

Para iniciar sesión el usuario debe tener en cuenta los siguientes datos:

1. Su número de extensión asignado
2. La dirección **IP** del servidor **Asterisk**
3. Contraseña

Debe ingresarse primero la extensión del usuario seguido de un arroba con la dirección IP del servidor **Asterisk** (Username @ PBX/VoIP provider) y confirmar con la contraseña (Password) (Figura 4.54).

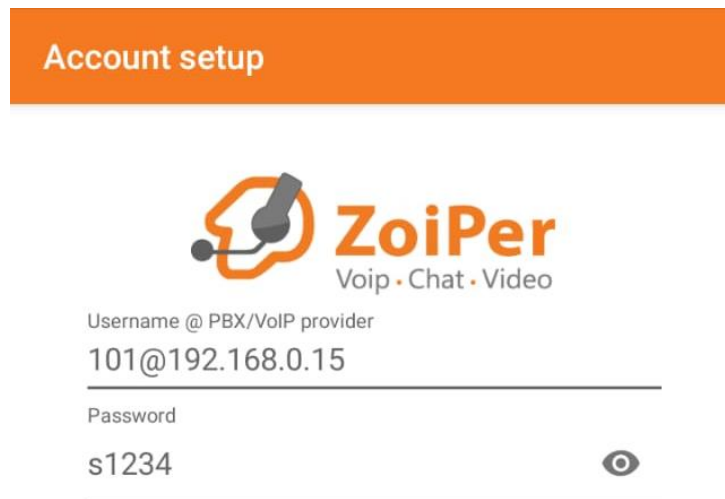


Figura 4.45 Registro y petición de la extensión.

Una vez concluida la configuración, la aplicación confirmará como en la siguiente imagen con el número de la extensión y la dirección IP del servidor **Asterisk** (Figura 4.55):



Figura 4.46 Confirmación de la extensión y de la dirección IP como servidor.

De haber ingresado algún dato incorrecto aparecerá de la siguiente manera (Figura 4.56):



Figura 4.47 Error en la extensión o error en la dirección IP como servidor.

Capítulo 5 Conclusiones

Existen diversas conclusiones derivadas de la presente tesis, a continuación se enlistan las que son consideradas como las más importantes:

- La diferencia más significativa entre la telefonía tradicional y la telefonía IP es el aprovechamiento del ancho de banda existente en sus respectivas redes. Esto se puede observar al momento en el que el sistema de Asterisk no aparta o asigna un ancho de banda para una llamada, por lo que el sistema no entra en un esfuerzo significativo, el cual, puede provocar su colapso de manera temprana.
- Existen infinidad de software dedicados para funcionar como gestor de un sistema de telefonía IP, a pesar de que los software no son la vía, si son de mucha ayuda en el momento de administrar los servicios de comunicaciones, sin embargo, muchos de estos softwares están basados en Asterisk y/o funcionan en conjunto con este.
- La tecnología de VoIP ha forjado alianzas con otros servicios aparte del hecho único de transmitir voz, como mensajería o videollamadas, siendo así, una forma de personalizar los servicios que el cliente requiere a través de una tecnología que solo brindaba un servicio (voz).
- Asterisk es por excelencia el mejor software de telefonía sobre IP, ya que se puede adaptar a las diversas exigencias del cliente. Funcionando de diversas formas como las que el administrador desee, gracias a que es del tipo Open Source.
- La combinación entre el microordenador Raspberry Pi funcionando con un sistema operativo basado en Linux y Asterisk es perfecta, tanto para una red de estudio como para la solución de una microempresa.
- La administración de recursos de la Raspberry Pi es imprescindible, por lo que si se requiere una red segura y con una cantidad considerable de llamadas al mismo tiempo, el diseño con RBP no es recomendable, debido al apartado QoS. De otra forma y siendo para uso académico

puede ser no necesario, aunque de igual forma se puede estudiar en la red implementada.

- Siendo Asterisk y Linux del tipo Open Source, el avance tecnológico de estos en gran parte son ofrecidos por sus comunidades de programación, beneficiando mayormente a las comunicaciones.
- Los costos de una implementación por medio de la RBP y Asterisk son relativamente bajos, debido a que la RBP son baratos y Asterisk es de código libre.
- La documentación de la Raspberry Pi en conjunto con Asterisk brindan al estudiante un vasto campo de conocimiento, desde el diseño de una red de computadoras hasta el aprendizaje del ambiente Linux.

Apéndice. Comandos de configuración

Comando 4.1 Sudo -i: Permite al usuario ejecutar comandos de superusuario.

Comando 4.2 iptables -L: Muestra el estado del firewall.

Comando 4.3 iptables -A INPUT -p tcp --dport 22 -j ACCEPT: Habilita el puerto 22.

Comando 4.4 /usr/bin/ssh-keygen -A: Genera una contraseña aleatoria.

Comando 4.5 service ssh start: Comienza el servicio SSH.

Comando 4.6 service ssh status: Comprueba el servicio SSH.

Comando 4.7 ifconfig: Muestra los valores de configuración de red

Comando 4.8 apt-get install asterisk: Instala la versión más estable de Asterisk.

Comando 4.9 asterisk -V: Muestra la versión instala de Asterisk.

Comando 4.10 service asterisk start: Comienza el servicio de Asterisk.

Comando 4.11 apt-cache search asterisk: Muestra las paqueterías disponibles para Asterisk.

Comando 4.12 apt-get install asterisk-core-sounds-es: Instala los sonidos en idioma español.

Comando 4.13 apt-get install asterisk-core-sounds-es-g722: Instala los sonidos G.722.

Comando 4.14 apt-get install asterisk-core-sounds-es-gsm: : Instala los sonidos GSM.

Comando 4.15 apt-get install asterisk-core-sounds-es-wav: Instala los sonidos WAV.

Comando 4.16 apt-get install asterisk-prompt-es: Instala de audios con indicaciones en español.

Comando 4.17 dpkg -l asterisk*: Muestra las paqueterías instaladas para Asterisk.

Comando 4.18 service asterisk restart: Reinicia el servicio de Asterisk.

Comando 4.19 service asterisk status: Muestra el estado del servicio de Asterisk.

Comando 4.20 nano /etc/asterisk/cel.conf: Abre el editor de texto con el archivo cel.conf.

Comando 4.21 radiuscfg => /etc/radcli/radiusclient.conf: Derivaciona la ubicación del archivo radiusclient.conf dentro del archivo cel.conf.

Comando 4.22 nano /etc/asterisk/cdr.conf: Abre el editor de texto con el archivo cdr.conf.

Comando 4.23 [radius]: Nombre del archivo dentro del cdr.conf.

Comando 4.24 vi /etc/asterisk/sip.conf: Abre el editor de texto con el archivo sip.conf.

Comando 4.25 g/^\s*/d: Delimita las líneas mostradas dentro del texto sip.conf, eliminando todas las líneas que comiencen con el carácter de punto y coma (;).

Comando 4.26 g/^\\$/d: Delimita las líneas mostradas dentro del texto sip.conf, eliminando todas las líneas que comiencen con el carácter de dinero (\$).

Comando 4.27 nano /etc/asterisk/sip.conf: Abre el editor de texto con el archivo sip.conf.

Comando 4.28 qualify = yes: Monitoreo de conexión.

Comando 4.29 language = es: Idioma del servicio Asterisk.

Comando 4.30 disallow = all: Desactiva todos los codificadores.

Comando 4.31 allow = ulaw,alaw: Codificación en orden preferencial.

Comando 4.32 accept_outofcall_message = yes: Activa los mensajes instantáneos dentro del servicio de Asterisk.

Comando 4.33 outofcall_message_context = *Nombre del contexto-messages*: Contexto del dialplan para el envío de mensajes instantáneos.

Comando 4.34 auth_message_requests = yes: Extensión de mensajería instantánea.

Comando 4.35 subscribecontext = *Nombre del contexto-messages2*: Contexto del dialplan de configuración.

Comando 4.36 [usuario](!): Plantilla llamada “*usuario*”.

Comando 4.37 type = friend: Privilegios del servicio de Asterisk al usuario.

Comando 4.38 host = dynamic: Registro de un dispositivo con cualquier dirección IP.

Comando 4.39 dtmfmode=rfc2833: Envío de tonos DTMF.

Comando 4.40 context = *Nombre del contexto*: Contexto del dialplan para llamadas.

Comando 4.41 [Número de extensión](usuario): Asignación de la extensión y llamado de la plantilla “*usuario*”.

Comando 4.42 username = *Nombre del usuario*: Nombre del usuario.

Comando 4.43 secret = *Contraseña*: Contraseña del usuario.

Comando 4.44 port = 5061: Puerto a utilizar por el servidor Asterisk.

Comando 4.45 asterisk –rvvvvv: Abre la consola de Asterisk.

Comando 4.46 sip show users: Muestra la información de las extensiones registradas en el servidor Asterisk.

Comando 4.47 sip show peers: Muestra el estado de las extensiones registradas en el servidor Asterisk.

Comando 4.48 nano /etc/asterisk/extensions.conf: Abre el editor de texto con el archivo extensions.conf.

Comando 4.49 [*Nombre del contexto*]: Nombre del contexto del dialplan para llamadas.

Comando 4.50 exten => *Número de extensión*,1,Dial(*SIP/Número de extensión*):
Procesos a ejecutar para la realización de una llamada.

Comando 4.51 [*Nombre del contexto-messages2*]: Nombre del contexto del dialplan de configuración.

Comando 4.52 exten => *Número de extensión*, hint, *SIP/ Número de extensión*:
Serie de procesos a ejecutar para el envío de un mensaje instantáneo.

Comando 4.53 [*Nombre del contexto-messages*]: Nombre del contexto del dialplan para el envío de mensajes instantáneos.

Comando 4.54 `exten => _X.,1,Noop(Mensaje de ${MESSAGE(from)})`: Muestra en consola el emisor del mensaje instantáneo.

Comando 4.55 `same => n,Noop(Mensaje para ${MESSAGE(to)})` : Muestra en consola el receptor del mensaje instantáneo.

Comando 4.56 `same => n,Noop(Texto = ${MESSAGE(body)})` : Muestra en consola el contenido del mensaje instantáneo.

Comando 4.57 `same => n,Messagesend(sip:${EXTEN},${MESSAGE(from)})` : Envío del mensaje instantáneo.

Comando 4.58 `same => n,Noop(Estado del mensaje${MESSAGE_SEND_STATUS})`: Muestra en consola el estado del mensaje instantáneo.

Comando 4.59 `same => n,Hangup`: Finaliza el servicio de mensajería.

Bibliografía

- 1.1 Anttalainen, T. (2003). *Introduction to Telecommunications Network Engineering, 2da ed.* Norwood: Artech House.
- 1.2 Cabezas Pozos, J. D. (2007). *Sistemas de Telefonía.* Madrid: Thomson.
- 1.3 del Río Ruíz, E. (2018). *Sistemas de telefonía fija y móvil.* Madrid: Paraninfo.
- 1.4 Freeman, R. L. (2004). *Telecommunication System Engineering, 4a Ed.* Nueva Jersey: Wiley-Interscience.
- 1.5 Freeman, R. L. (2005). *Fundamentals of Telecommunications, 2a Ed.* New Jersey: Wiley-Interscience.
- 1.6 Gnanasivam, P. (2006). *Telecommunication switching and networks, 2a ed.* Chennai: New Age International Publishers.
- 1.7 Moro Vallina, M. (2013). *Infraestructuras de redes de datos y sistemas de telefonía.* Madrid: Paraninfo.
- 1.8 Nopal Pascual, G. A., & Zárraga Estrada, J. C. (2008). *Tecnologías Open Source en Telefonía IP (Tesis de grado).* México: Universidad Nacional Autónoma de México.
- 1.9 Stallings, W. (2004). *Comunicaciones y Redes de Computadoras, 7a Ed.* Madrid: Pearson Educación.
- 1.10 Tanenbaum, A. S., & Wetherall, D. J. (2012). *Redes de Computadoras, 5a Ed.* México: Pearson Educación.
- 1.11 Vergara, M., & Huidobro, J. M. (2016). *Las tecnologías que cambiaron la historia.* Madrid: Ariel.

- 2.1 Briceño M., J. E. (2005). *Transmisión de datos, 3ra Edición.* Mérida: Taller de Publicaciones de la Facultad de Ingeniería, ULA.
- 2.2 Castells, M. (2001). *La Galaxia Internet.* Barcelona: Plaza & Janés Editores.
- 2.3 Dordoigne, J. (2015). *Redes Informáticas.* Barcelona: Ediciones ENI.
- 2.4 Gómez, J. A. (2010). *Servicios en red.* Madrid: Editex.
- 2.5 Hesselbach Serra, X., & Altés Bosch, J. (2002). *Análisis de redes y sistemas de comunicaciones.* Barcelona: Edicions UPC.
- 2.6 Kurose, J. F., & Ross, K. W. (2017). *Redes de computadoras. Un enfoque descendente, 7 Ed.* Madrid: Pearson Educación.
- 2.7 Pons, O. R. (2014). *Introducción a las Telecomunicaciones Fijas y Móviles.* Buenos Aires.

- 3.1 Gómez López, J., & Gil Montoya, F. (2009). *VoIP y Asterisk. Redescubriendo la telefonía*. Almería: Alfaomega RA-MA.
 - 3.2 Jordan, M. (15 de Marzo de 2015). *Asterisk*. Obtenido de Dialplan: <https://wiki.asterisk.org/wiki/display/AST/Dialplan>
 - 3.3 López Aldea, E. (2017). *Raspberry Pi Fundamentos y Aplicaciones*. Madrid: RA-MA Editorial.
 - 3.4 Machi, J. (22 de Octubre de 2019). *How is Asterisk Different from FreePBX?* Obtenido de Sangoma: <https://www.sangoma.com/blog/asterisk-freepbx-difference/>
 - 3.5 Mahler, P. (2004). *VoIP Telephony with Asterisk*. Mishawaka: Signate.
 - 3.6 Sierra Rodríguez, A. (2008). *Instalación de un sistema VoIP corporativo basado en Asterisk*. Cartagena: Universidad Politécnica de Cartagena.
 - 3.7 Upton, E., & Halfacree, G. (2016). *Raspberry Pi. User Guide 4th ed*. Chichester: Wiley.
 - 3.8 Van Meggelen, J., Bryant, R., & Madsen, L. (2019). *Asterisk. The Definitive Guide. 4th ed*. Gravenstein Highway North: O'Reilly Media.
 - 3.9 Van Meggelen, J., Madsen, L., & Smith, J. (2007). *Asterisk. The Future of Telephony. 2nd Ed*. Gravenstein Highway North: O'Reilly Media.
-
- 4.1 Sistema de comunicación 3CX (18 de Marzo de 2020). VOIP-info: a reference guide to all things VOIP. Obtenido de Asterisk: <https://www.voip-info.org/asterisk/>