



UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
VERITATE SOLA NOBIS IMPONETUR VIRILISTOGA. 1948

FACULTAD DE INGENIERÍA ELÉCTRICA

Departamento de Electrónica y Telecomunicaciones

TRABAJO DE DIPLOMA

“Solución de telefonía basada en la PBX Asterisk”

Autor: Oswald Caballero Avendaño

Tutor: M.Sc. Carlos A. Rodríguez López

Consultante: Ing. Samuel Montejo Sánchez

Santa Clara

Curso 2004 – 2005

“Año de la Alternativa Bolivariana para las Américas”



UNIVERSIDAD CENTRAL “Marta Abreu” DE LAS VILLAS

FACULTAD DE INGENIERÍA ELÉCTRICA

Departamento de Electrónica y Telecomunicaciones



TRABAJO DE DIPLOMA

“Solución de telefonía basada en la PBX Asterisk”

Autor: Oswald Caballero Avendaño

e-mail: oswaldca@yahoo.es

Tutor: M.Sc. Carlos A. Rodríguez López

Prof. Dpto. de Telecomunicaciones y electrónica

Facultad de Ing. Eléctrica. UCLV.

e-mail: crodrigz@uclv.edu.cu

Consultante: Ing. Samuel Montejo Sánchez

Prof. Dpto. de Telecomunicaciones y electrónica

Facultad de Ing. Eléctrica. UCLV.

e-mail: montejo@uclv.edu.cu

Santa Clara

Curso 2004 – 2005

“Año de la Alternativa Bolivariana para las Américas”



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería en Telecomunicaciones y Electrónica autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Autor

Firma del Jefe de Departamento
donde se defiende el trabajo

Firma del Responsable de
Información Científico-Técnica

Pensamiento

*“Solamente el esfuerzo propio
puede producir los beneficios
que muchos esperan por milagros.”*

J.A. Rosenkranz

Agradecimientos



Agradecimientos

Ante todo deseo agradecerles a mis padres por su apoyo incondicional, a mi abuela, a mis hermanos, al resto de mi familia, a mi novia por su paciencia. A todos y cada uno de ellos, llegue el más cálido y sincero de los agradecimientos.

A mi tutor a Samuel y a Héctor, por ser más que profesores, amigos.

A todos mis amigos, los que se graduaron (sobre todo a Yunior y Ariel) y por supuesto a los que no se graduaron.

A mis compañeros de aula, y a los que todavía les queda un largo camino que recorrer.

A todos mis profesores por haber contribuido en mi superación.

A todas las personas con las que tuve el placer de trabajar en el Rectorado en especial a Yobandra, Andrés, Yadel y Alcides.

Y a todos los que aunque su nombre no esté presente, queda grabado aquí.

Dedicatoria

Dedicatoria

Dedico este trabajo a todas las personas a quienes anteriormente me referí en los agradecimientos.

A la Universidad, la cual consideré mi hogar durante los años en que cursé mis estudios, y a la cual le debo los mejores años de mi vida.

Y además dedico este trabajo a mi persona, por haberme convertido en un eterno estudiante.

En fin, que llegue a todos mi más modesto y sincero homenaje.

Resumen

RESUMEN

En este trabajo se exponen las características principales de la PBX Asterisk, profundizando en aspectos relacionados con la instalación, configuración y puesta en funcionamiento de la misma, se analizan los servicios que presta así como la implementación de algunos servicios de valor agregados.

La PBX Asterisk se presenta como solución para elevar la densidad telefónica en la universidad Central “Marta Abreu” de las Villas y brindar un conjunto de servicios telefónicos sobre la red de datos. Este tipo de solución se ajusta especialmente al escenario universitario ya que aprovecha la infraestructura de la red de datos existente y mantiene un costo de implementación bajo. Con este trabajo se recomienda además el uso de la PBX en el ámbito académico para el estudio real de telefonía de la asignatura “Conmutación Telefónica” del perfil de Ingeniería en Telecomunicaciones.

Índice



ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1.....	3
DEFINICIÓN DE ASTERISK.....	3
1.1 IMPORTANCIA DE ASTERISK.....	4
1.2 CARACTERÍSTICAS Y SERVICIOS SOPORTADOS POR ASTERISK.....	5
1.3 IMPLEMENTACIÓN DE FUNCIONES COMUNES PBX CON ASTERISK.	5
1.4 ARQUITECTURA DE ASTERISK.	6
1.4.1 API DE CANAL, (CHANNEL API).....	7
1.4.2 API PARA TRADUCCIÓN DE CÓDIGO, (CODEC TRANSLATOR API)	7
1.4.3 API PARA FORMATO DE FICHEROS, (FILE FORMAT API)	7
1.4.4 API DE APLICACIONES, (APPLICATION API)	8
1.5 EL PLAN DE DISCADO DE ASTERISK.....	8
1.6 ALGUNOS EJEMPLOS DE REDES CON ASTERISK.....	9
1.7 ANÁLISIS DE COSTO DE SOLUCIONES QUE USAN SOFTWARE LIBRE.	11
CAPÍTULO 2.....	13
INSTALACIÓN Y CONFIGURACIÓN DE ASTERISK	13
2.1 ORGANIZACIÓN DE LOS FICHEROS	13
2.3 CONFIGURACIÓN DE ASTERISK.....	14
2.3.1 CANALES.....	14
2.3.2 CONFIGURACIÓN DEL PLAN DE DISCADO.	15
<i>Sección de configuración: [general]</i>	15
<i>Sección de configuración: [globals]</i>	15
2.3.3 CONFIGURACIÓN DE CONTEXTOS Y EXTENSIONES.	16
<i>Definición de Extensiones</i>	17
<i>Nombres de Patrones de Extensiones</i>	18
<i>Nombres de extensiones predefinidas para propósitos generales:</i>	18
<i>Patrones de extensiones:</i>	19
<i>Inclusión de Contexto</i>	20
2.3.4 USO DE VARIABLES EN EL DISEÑO DEL PLAN DE DISCADO.	21
<i>Aplicación de variables específicas</i>	22
<i>Variables específicas a utilizar en las Macro</i>	22
<i>Variables de entorno</i>	23
<i>Manipulación de cadenas</i>	23
<i>Subcadenas</i>	23
<i>Concatenación de cadenas</i>	24
2.3.5 IMPLEMENTACIÓN DE MACROS EN ASTERISK.	24
2.4 CONFIGURACIÓN DE SERVICIOS.....	25
2.4.1 CONFIGURACIÓN SIP.....	25
<i>Configuración SIP sección [general]</i>	25
2.4.2 CONFIGURACIÓN DEL CORREO DE VOZ.	27
<i>Parámetros de la sección [general]</i>	27
<i>Parámetros para la sección [zonemessages] en el fichero voicemail.conf</i>	32

<i>Parámetros para la sección de [CONTEXTS] del fichero voicemail.conf</i>	33
2.4.3 CONFIGURACIÓN DE INTER ASTERISK EXCHANGE (IAX).....	35
<i>Convenciones para la configuración de ficheros</i>	36
<i>Configuración de IAX en Asterisk</i>	36
<i>Configuración del cliente IAX</i>	38
<i>Configuración del Par</i>	39
<i>Codificadores Codecs</i>	40
<i>TOS (tipo de servicio)</i>	40
<i>Opciones de calidad</i>	41
<i>En los anexos se muestran ejemplos de configuración de IAX</i>	41
2.4.4 CONFIGURACIÓN DEL SERVICIO DE COLAS.....	42
<i>Menú para el usuario</i>	42
<i>Configuración del fichero queues.conf</i>	43
<i>Configuración de los agentes. agents.conf</i>	43
2.4.5 DISTRIBUCIÓN AUTOMÁTICA DE LLAMADAS (ACD).....	44
2.4.6 PARQUEO DE LLAMADAS.....	45
2.4.7 SERVICIO LASTCALLER.....	46
2.4.8 REPETICIÓN DEL ÚLTIMO NÚMERO MARCADO.....	46
2.4.9 SERVICIO DE MÚSICA EN ESPERA.....	47
2.4.10 SERVICIO DE REDIRECCIONAMIENTO DE LLAMADAS.....	49
2.4.11 SERVICIO DE CONFERENCIAS.....	51
CAPITULO 3	52
INTERFAZ CON LA PSTN Y PRUEBA DE SERVICIOS	52
3.1 EL HARDWARE PARA ASTERISK	52
3.1.1 HARDWARE ZAPTEL.....	52
3.1.2 X100P Y X101P.....	52
3.1.3 S100U – INTERFAZ DE PUERTO SIMPLE FXS USB.....	52
3.1.4 TDM 400P.....	53
3.1.5 T400P, E400P, TE410P Y TE405P – INTERFAZ T1/E1.....	53
3.1.6 TE110P, – INTERFAZ SIMPLE T1/E1.....	53
3.1.7 VALORACIÓN DEL HARDWARE A USAR EN LA SOLUCIÓN UNIVERSITARIA.....	53
3.2 PRUEBA DE SERVICIOS	54
CONCLUSIONES	55
RECOMENDACIONES	56
REFERENCIAS BIBLIOGRÁFICAS	57
ANEXO I	58
INSTALACIÓN DE ASTERISK EN LINUX REDHAT 9.....	58
ANEXO II	63
INSTALACIÓN DE ASTERISK EN LINUX KNOPPIX BASADO EN DEBIAN 3.0.....	63
ANEXO III	65
CONFIGURANDO UN CLIENTE X-LITE PARA ASTERISK.....	65
ANEXO IV	66
VARIABLES DE CANAL PREDEFINIDAS.....	66
ANEXO V	67
EJEMPLO DE CONFIGURACIÓN DE LA INTERFAZ ZAPTEL.....	67
ANEXO VI	69

EJEMPLO DE CONFIGURACIÓN DEL CORREO DE VOZ EN LA UCLV	69
ANEXO VII	71
EJEMPLO DE CONFIGURACIÓN DE IAX	71
ANEXO VIII.....	72
EJEMPLO DE CONFIGURACIÓN SIP.....	72
ANEXO IX.....	74
EJEMPLO DE CONFIGURACIÓN DEL SERVICIO DE COLAS.....	74
ANEXO X	77
CONFIGURACIÓN DEL SERVICIO MÚSICA EN ESPERA.	77
ABREVIATURAS.....	78

Introducción



INTRODUCCIÓN

En la Universidad Central de las Villas, la situación de comunicación telefónica actual es considerablemente desfavorable, según estudios realizados en trabajos para el forum “*Joven ciencia*”, la densidad telefónica es relativamente baja, no supera los 2.766 teléfonos por cada cien personas, y para el caso de los estudiantes tomando en cuenta solamente los teléfonos públicos existentes, la densidad es de solo 0.286 por cada cien personas.

En este trabajo se explora un variante tecnológica para darle solución a este problema, la cual consiste en utilizar la PBX Asterisk para desarrollar la telefonía sobre la red de datos existente en la universidad, permitiendo convertir o utilizar cada computadora como un terminal telefónico.

Este trabajo tiene como objetivo poner en funcionamiento la PBX Asterisk como soporte de servicios de telefonía.

Mediante la utilización del software propuesto en este trabajo, se puede crear una estructura que eleva considerablemente la densidad telefónica en el entorno universitario, la cual adquiere amplias posibilidades de expansión a otras universidades y centros del país que cuenten con su propia red de datos.

El conjunto de los servicios integrales de voz sobre IP nos permiten la utilización de la plataforma IP para llevar a cabo la manipulación de las llamadas internas a oficinas a nivel local en la universidad evitando así la utilización de la Red Pública de Servicios Telefónicos lo cual además significa un ahorro considerable para la universidad.

Con la prestación de estos servicios se crea una amplia flexibilidad; un ejemplo de ello entre otros es la movilidad de las extensiones, con lo cual se le permite a los usuarios realizar llamadas desde cualquier lugar en la red.

La plataforma propuesta permite además interacción con la Red de Telefonía Pública Conmutada, ampliando de esta forma las posibilidades de comunicación así como la creación de nuevos servicios de valor agregados en la universidad.

Esta solución es de elevada importancia en el ámbito académico para nuestra universidad, ya que en la actualidad no existen laboratorios para las asignaturas donde se imparten conocimientos sobre telefonía, debido al elevado costo del equipamiento necesario para tal.

Con la utilización de estos servicios se crea la posibilidad de implementar laboratorios reales y virtuales, brindándole así al estudiante la posibilidad de interactuar con un sistema de conmutación telefónica para complementar sus estudios sobre dicha materia, ampliando de esta forma su perfil de conocimientos.

El software utilizado en este trabajo es el denominado “*Asterisk*”, PBX híbrida de código abierto, la cual provee los servicios soportados por el equipamiento PBX actual y además nos brinda la posibilidad de expansión de nuevos servicios de valor agregado.

Este software es soportado por el sistema operativo Linux, del cual se utilizaron satisfactoriamente las distribuciones RedHat 9.0 y Knoppix (Debian 3.0).

Los protocolos de señalización utilizados por Asterisk son (SIP, H323, MGCP) y un protocolo propio creado para la interconexión entre servidores Asterisk llamado IAX.

Soporta además la transmisión de paquetes de voz sobre la red de datos, permite mediante un seudo TDM, el manejo y transmisión de la voz por las interfaces de telefonía bajo las normas de señalización americana y europea.

Partiendo de lo anteriormente expuesto se pretende guiar la investigación de la siguiente manera, llevando a cabo las siguientes tareas:

1. Revisión bibliográfica y estudio de experiencias en el empleo de la PBX Asterisk
2. Instalación y configuración del Software.
3. Creación del plan de direcciones y configuración de ficheros.
4. Instalación, configuración y prueba de interfaz con la PSTN
5. Prueba de servicios

Con el fin de lograr una correcta orientación a cerca del tema en cuestión se dispone de una estructuración básica de tres capítulos.

El primer capítulo abordará de manera general sobre la bibliografía relacionada con Asterisk.

En el segundo capítulo se brindará un acercamiento a los temas de configuración, instalación del software y la creación del plan de direcciones.

El tercer capítulo se refiere a la instalación del hardware para Asterisk, la configuración y experiencias realizadas con la interfaz ZAPTEL así como prueba de servicios instalados.

Capítulo 1



Capítulo 1

Definición de Asterisk

Asterisk es una PBX completa diseñada en software de código abierto según Asterisk (2005), funciona en Linux y proporciona todas las características de una PBX híbrida de paquetes de voz sobre IP y TDM, plataforma de respuesta interactiva de voz (IVR) y Soft-Switch, que brinda funciones de distribución automática de llamadas (ACD).

Está diseñado para permitir la interacción con diferentes tipos de hardware y software de telefonía IP. Asterisk puede manipular TDM y paquetes de voz a niveles inferiores mientras nos ofrece una plataforma sumamente flexible para PBX y aplicaciones de telefonía en niveles superiores.

Este software puede ser instalado sobre cualquier plataforma Linux o Unix, utilizado como servidor de telefonía IP entre máquinas conectadas a la red. El mismo puede expandir su conexión hasta la Red de Telefonía Pública a través del hardware de interfaz telefónica (ZAPTEL).

Su nombre significa según Manesh (2004) * **(asterisco)** el cual proviene de las sintaxis de comandos de UNIX y DOS y significa un comodín (cualquier cosa) lo cual lo denota como componente muy versátil en las redes de voz.

Asterisk trabaja con Voz sobre IP para lo cual puede emplear diferentes tipos de protocolos de señalización tales como SIP, MGCP y H323. Asterisk además puede convertirse en plataforma servidor de servicios de predicción de discado, IVR de configuración personal, oficina PBX remota o centralizada y servidor de conferencias.

Luego de la creación de Asterisk, su creador realizó un equipo junto con el proyecto de Telefonía Zapata para crear tarjetas de expansión que fueran baratas basadas en el hardware de Intel las cuales servirían como interfaz entre la plataforma de servidor Asterisk y la Red de Telefonía Pública.

Manesh (2004) expresa que la idea central del proyecto fue que cualquier persona en cualquier lugar pudiera comprar una computadora, instalarle el sistema operativo Linux, adicionarle a la computadora una tarjeta de expansión para conexiones FXO/E1/T1, a toda esta unión añadirle el software Asterisk y al final tener una PBX con todas sus características y servicios con un desempeño aceptable, a un costo mucho menor que cualquier solución tradicional de PBX.

1.1 Importancia de Asterisk.

Existen diferentes razones para considerar Asterisk como una importante pieza de software. Spencer (2002) hace referencia a:

- Alta reducción de costo debido al empleo de software libre combinada con el bajo precio del hardware para telefonía. Con la unión de ambas se obtiene una PBX a un menor costo que cualquier solución tradicional, además de proveer un nivel de funcionalidad equivalente a los modernos sistemas disponibles en el mercado.
- Control. Asterisk le permite al usuario tomar el control de su sistema de telefonía. Una vez que la llamada esta dentro del servidor Linux, cualquier manipulación puede ser llevada a cabo sobre la misma.
- Rápida extensión y desarrollo. Su poderosa interfaz de comando así como los ficheros texto de configuración le permiten una rápida puesta a punto, configuración y diagnóstico en tiempo real.
- Adaptación y flexibilidad. Asterisk puede ser acoplada a cualquier sistema existente, debido a la flexibilidad ofrecida por sus ficheros de configuración y código fuente.
- Plan de discado extremadamente flexible. Este permite la integración de funcionalidades IVR y PBX. Muchas de las características existentes de Asterisk, así como otras a diseñar, pueden ser implementadas usando programación por lógica de extensiones.

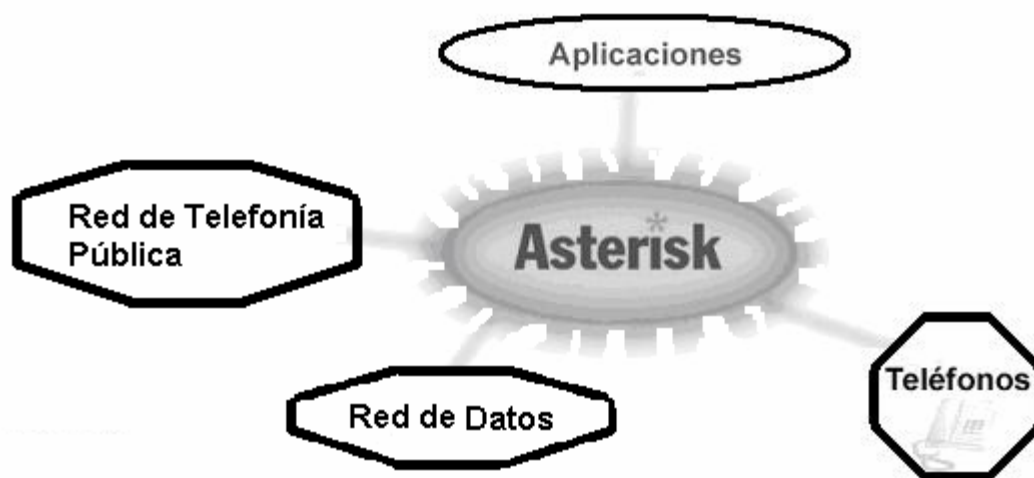


Figura 1 Ejemplo de potencialidad de Asterisk.

En la figura 1 se muestra un ejemplo de potencialidad de Asterisk. En esta se observa la conexión con la red de datos (puede ser la red interna o a la Internet) y con la Red de Telefonía Pública Conmutada, entre otras.

1.2 Características y servicios soportados por Asterisk

Asterisk se presenta como una PBX con todas sus funcionalidades actuales, soporta todas las características de llamadas convencionales en una interfaz de estación, tales como identificador de llamadas, .redireccionamiento de llamadas para llamadas ocupadas, redireccionamiento de llamadas para llamadas no contestadas, servidor de conferencias, redireccionamiento de llamadas variables, transferencias supervisadas, transferencias no supervisadas, incrementos ADSI, correo de voz con directorio, ruteo a costo mínimo, ruteo de VoIP, aplicaciones de prepago, servicio de llamadas en espera, reconocimiento de voz, discado predictivo, colas de llamadas, respuesta interactiva de voz, entre otras.

Asterisk es programable en diferentes niveles tales como:

- **Lógica de extensiones:** mediante esta y usando procesamientos lógicos de programación se pueden lograr, tomando como basamento ejemplos ya creados, simples aplicaciones, servicios entre otros.
- **Interfaz de Pasarelas Asterisk (AGI):** esta es utilizada en la implementación de tareas más complejas. A través de esta se puede crear Scripts con lenguajes tales como PHP, PERL etc.
- **Nivel de programación C:** este es el nivel más bajo de programación, es utilizado para implementar aplicaciones, crear controladores (drivers), y demás. En general este nivel es necesario para la creación de aplicaciones que requieren de mayor envergadura en su programación (CVS).

1.3 Implementación de funciones comunes PBX con Asterisk.

Algunas funciones en Asterisk son implementadas como aplicaciones o son soportadas por combinación de aplicaciones que se usan en el plan de discado generalmente soportadas por todos los canales como se explica detalladamente en Asterisk (2005).

A continuación se muestra una lista de algunas funciones PBX soportadas en Asterisk.

- Music on Hold (Música en espera): ver comando MusicOnHold()
- Call Parking (Parqueo de llamadas): soportado en la instalación estándar.
- Call Recording (grabación de llamadas): Usando la aplicación 'Monitor'
- Conferencing (servicio de conferencias): Usando la aplicación 'MeetMe'
- IVR: (Respuesta de Voz Interactiva) Empleando AGI o EAGI

- Unconditional Call Forwarding (redireccionamiento incondicional)
- Attended Transfer (or "consultative transfer")
- No Answer Call Forwarding: Implementada en el plan de discado.
- Busy Call Forwarding: Implementada en el plan de discado
- 3-way Calling: Normalmente implementada por el teléfono.
- Incoming Call Screening: Implementado en el plan de discado.
- Automatic Redial(Marcado automático):implementada en el plan de discado
- Manual Redial (rediscado manual).
- Do-not-disturb (DND) Normalmente implementada por su teléfono; también es implementada en Asterisk (*78 y *79)
- Unattended Transfer (or "blind transfer")
- Attended Transfer (or "consultative transfer")
- No Answer Call Forwarding: Implementada en el plan de discado.
- Busy Call Forwarding: Implementada en el plan de discado.
- Single-Line Extension.
- Call Deflection (CD) (redireccionado sin respuesta): Implementada por la aplicación chan_capi
- CID & DNID: Implementada por la aplicación chan_capi

1.4 *Arquitectura de Asterisk.*

Asterisk sigue un diseño modular. Las diferentes funciones del servidor son implementadas como módulos las cuales son cargadas e inicializadas por el cargador en tiempo de ejecución (run time) Spencer (2003). Cuando Asterisk es inicializado, el cargador dinámico de módulos carga e inicializa cada uno de los controladores (drivers) los cuales proporcionan controladores de canal, formatos de ficheros, Codecs, aplicaciones etc. Luego los enlaza con la API interna apropiada para su tratamiento. Se puede encontrar cuatro APIs diferentes para manipular los distintos componentes.

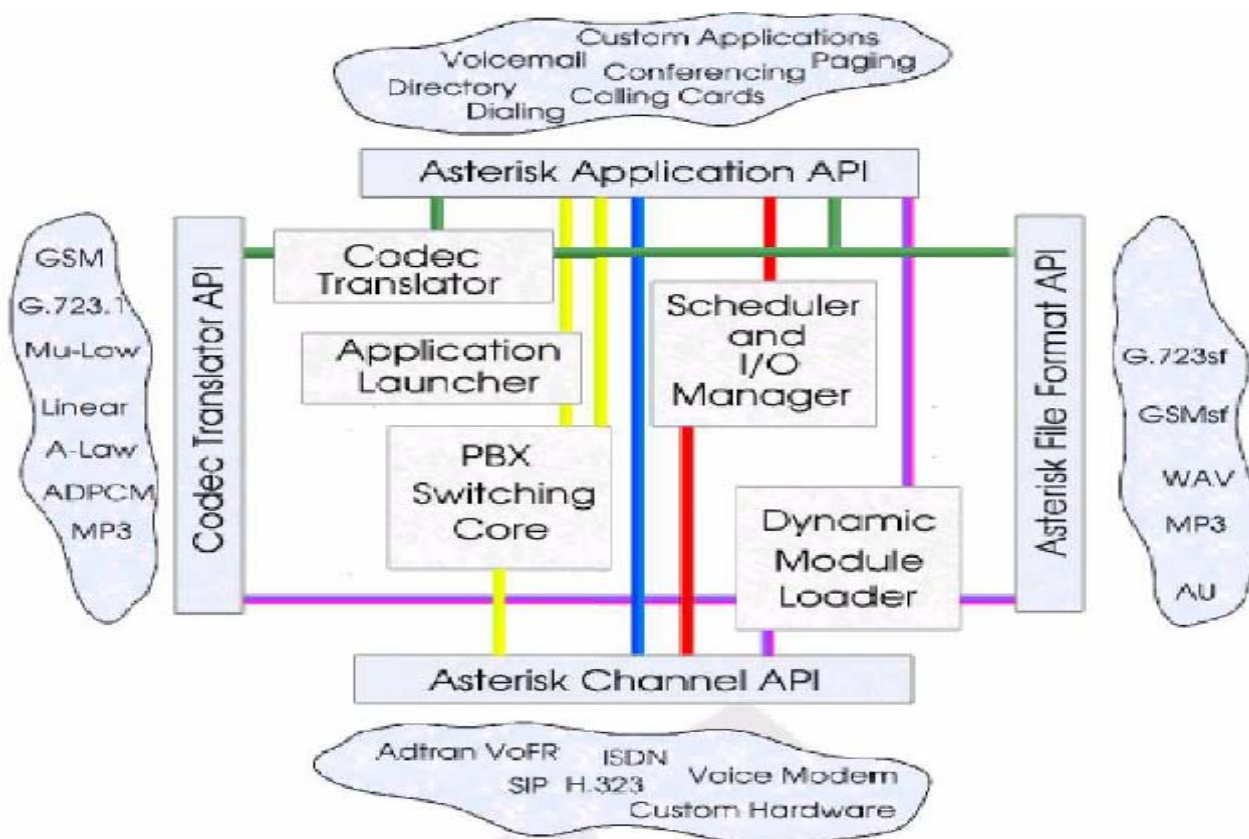


Figura 2 Arquitectura de Asterisk.

1.4.1 API de canal, (Channel API)

La API de canal le permite al núcleo de conmutación de Asterisk interconectarse con las fuentes TDM o las de paquetes de voz. Las fuentes TDM son generalmente dispositivos pseudo TDM para los cuales el código desempeña funciones DSP.

Esta API es usada también para otras fuentes TDM tales como tarjetas de lógica de discado las cuales están soportadas como ADD-ONS de Asterisk.

1.4.2 API para traducción de código, (Codec translator API)

Esta API provee de una forma flexible para que el núcleo interactúe con voz codificada no importando de cuales canales provenga esta. Los formatos tales como GSM, G.723, ADPCM y MP3 son soportados.

1.4.3 API para formato de ficheros, (File format API)

Le permite a Asterisk la capacidad de leer y reproducir sonidos en diferentes formatos, incluyendo WAV, AU, GSM y MP3. Además le da a las aplicaciones basadas en Asterisk mayor flexibilidad con lo referente a tonos de discado, DTMF, etc.

1.4.4 API de aplicaciones, (Application API)

Esta API le permite a Asterisk ser usada por terceras aplicaciones, por ejemplo conferencias, el correo de voz, tarjetas de llamada, entre otras. Todas estas aplicaciones pueden tomar ventaja con esta API. Esta hace además posible escribir nuevas aplicaciones de telefonía que interactúen directamente con el núcleo PBX.

Asterisk posee además una interfaz separada: Asterisk Gateway Interface (AGI) la cual representa un mecanismo para ejecutar programas externos desde dentro de Asterisk basado en el las reglas del plan de discado

1.5 El Plan de Discado de Asterisk.

El plan de discado le proporciona a Asterisk gran parte de su flexibilidad, ya que este es el encargado de determinar como las llamadas son enrutadas a través del sistema. Según Spencer (2002) el plan está compuesto por grupos (generalmente conectados) de contextos de extensiones los cuales no son más que simples colecciones de extensiones. Una extensión puede aparecer en más de un contexto y a la vez un contexto puede incluir a otro con limitaciones opcionales de tiempo y fecha. Los contextos además pueden referenciar conmutadores externos (ejemplo de esto es IAX) permitiéndole de esta forma al plan de discado ser aumentado.

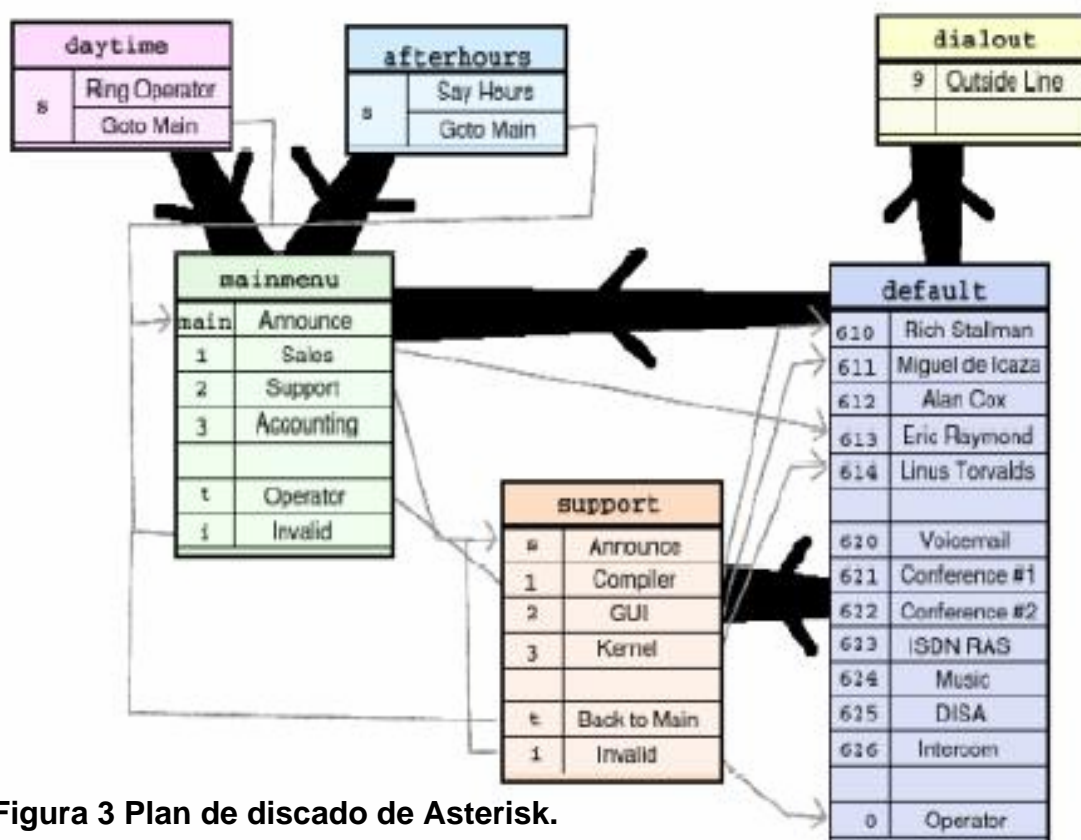


Figura 3 Plan de discado de Asterisk.

El diagrama de la figura 3 representa una PBX ficticia para la comunidad Linux GNU. Cada contexto está en su propia tabla, listando las extensiones y lo que significa. La línea fina gris representa enlaces entre extensiones de un contexto y otro. Las líneas gruesas representan la inclusión de un contexto en otro. Las flechas apuntan desde el contexto que ha sido incluido.

El uso creativo de los contextos puede ser usado par implementar diferentes características tales como:

- **Seguridad:** el plan de discado puede ser segmentado para permitirle a cierto número de extensiones la accesibilidad a distintas interfases o usuarios VoIP.
- **Menús de voz de múltiples niveles:** debido a que las extensiones pueden ser de cualquier tamaño y también se pueden rehusar en cualquier contexto, se puede crear menú de voz.
- **Servicios de Autenticación:** los contextos pueden ser usados para verificar claves o códigos de acceso para la utilización servicios específicos.
- **Retorno de llamada:** para este tipo de servicio pueden ser usadas los contextos de extensiones, scripts externos, y la aplicación app_qcall.
- **Centralización del plan de discado:** usando el protocolo IAX se pueden conectar varios planes de discado centralizando en un solo servidor un plan de discado complejo.
- **Modos Día/Noche:** se puede variar el comportamiento de la PBX según el horario.
- **Ruteo:** ruteo de llamadas basado en extensiones.

Entre otras

1.6 Algunos ejemplos de redes con Asterisk.

En este punto se presentan algunos ejemplos de diagramas de redes que se pueden implementar con Asterisk así como algunos ya implementados como prueba en la universidad.



Figura 4 PBX 1x1.

En esta figura se muestra la conexión de Asterisk con un puerto a la Red de Telefonía Pública un puerto a teléfonos IP.

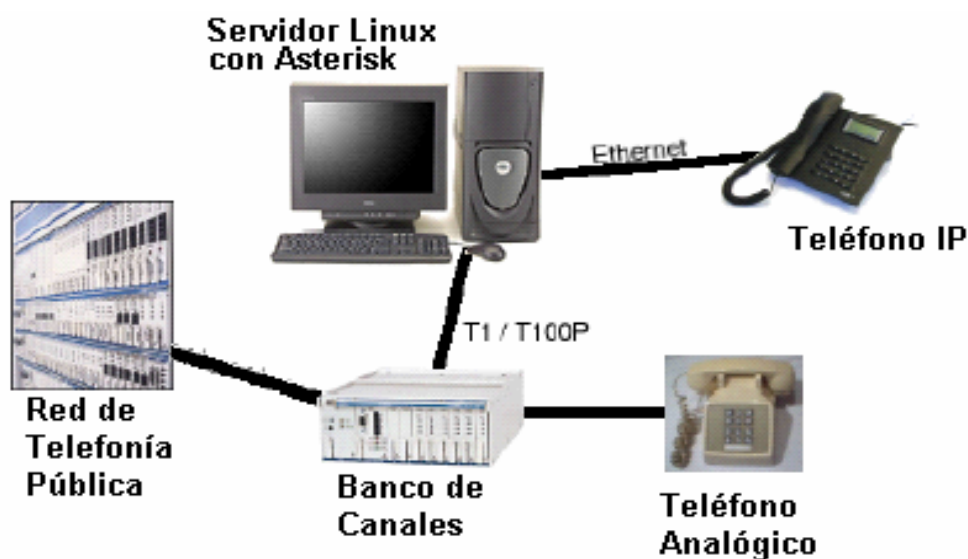


Figura 5 Conexión de Asterisk a un banco de canales.

En esta figura se muestra otra variación de Asterisk a la cual se le conecta un banco de canales mediante un T1 para darle servicio a 16 estaciones y además internamente puede tener 8 líneas de entrada.

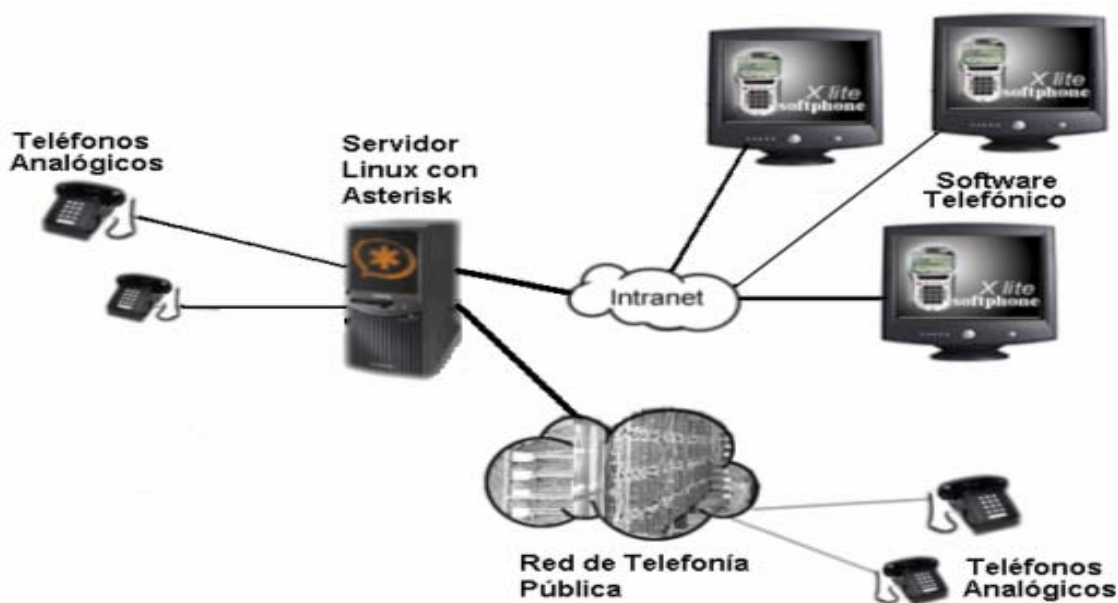


Figura 6 Ejemplo de aplicación de Asterisk en la UCLV.

En esta figura se muestra un modelo el cual actualmente se encuentra aplicado en modo de prueba en la red interna de la UCLV.

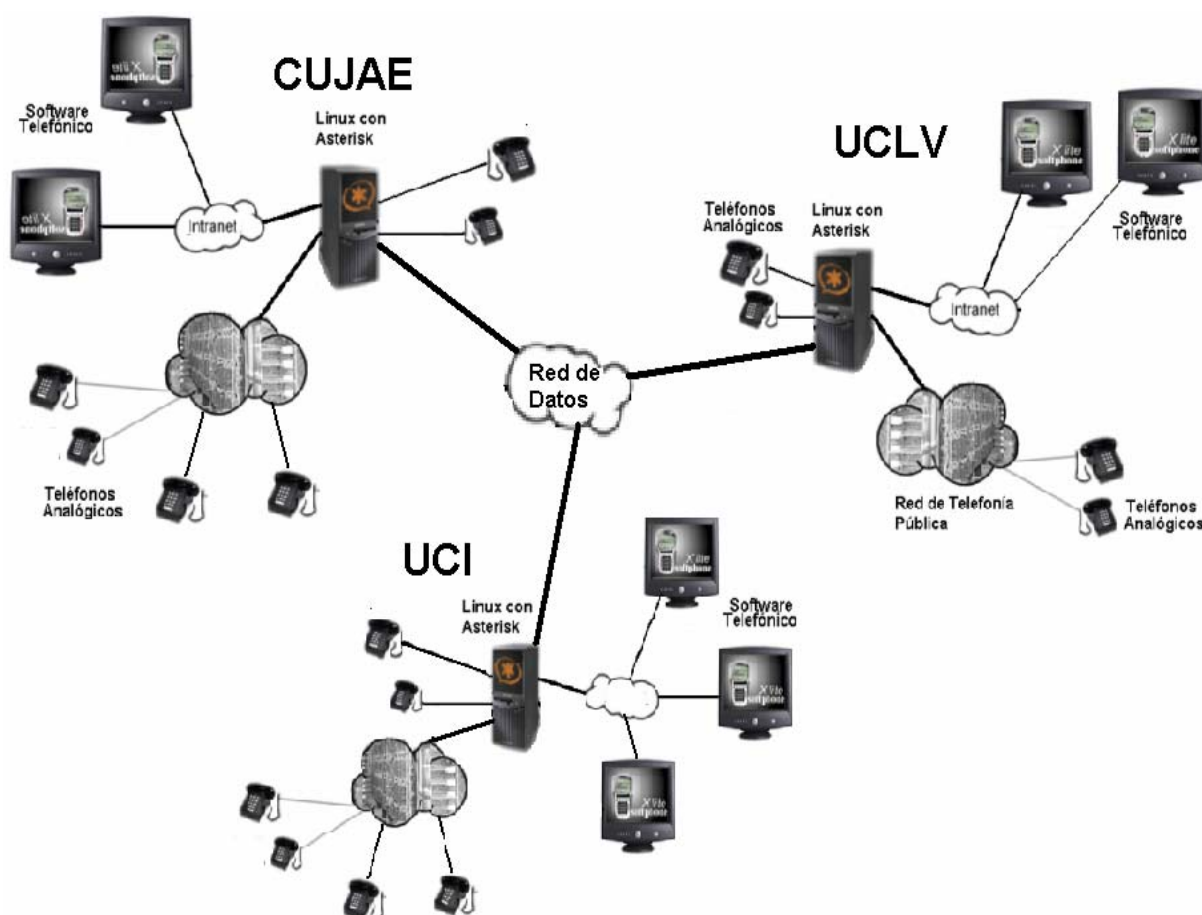


Figura 7 Conexión de diversos centros mediante Asterisk.

Este esquema de conexión muestra como a través de servidores Asterisk se podría implementar comunicación entre diversos CES u otros organismos que cuenten con su propia red de datos.

1.7 Análisis de costo de soluciones que usan software Libre.

Mediante la utilización del Software Libre se puede alcanzar una relación costo beneficio favorable, esto se aprecia a partir de la comparación con implementaciones basadas en proyectos que utilizan Software Propietario.

Con el uso de las soluciones de software libre, el costo disminuye, facilitándose además el mantenimiento, la actualización y el mejoramiento debido al libre acceso al código fuente de las instalaciones así como de las actualizaciones del software.

La siguiente tabla nos sirve para ilustrar lo antes mencionado. VoiceTronix (2002)

Componente	Sistema Propietario	Código Abierto
PC	\$700	\$500
Tarjeta de Interfaz con la PSTN	\$595	\$595
Sistema Operativo	\$300 (Microsoft Win2k)	Free (Linux)
Aplicación	\$3000	Free
Total	\$4595	\$1095

Notas:

- Los precios mostrados son aproximados en USD.
- Para la mayoría de las aplicaciones los requerimientos de computadoras no son demasiado altos.

En esta tabla se podrá apreciar un ahorro sustancial para los sistemas de código abierto, los beneficios en costo pueden representar alrededor de un 70%.

Capítulo 2



Capítulo 2

Instalación y configuración de Asterisk

Prerrequisitos, Conocimientos y Habilidades.

Para configurar el sistema satisfactoriamente se requiere algo más que una pasajera familiaridad con diversos conceptos técnicos tales como instalación y administración de Linux así como la asimilación de algunos conceptos de Telecomunicaciones. En este capítulo se tratarán ligeramente los temas sobre estas complejas tecnologías haciendo énfasis solamente en los conceptos relevantes a la instalación, configuración y administración de Asterisk.

En los anexos se brindan guías detalladas de cómo instalar Asterisk en las distribuciones Linux RedHat 9 y en Knoppix 3.3 basadas en las experiencias obtenidas con la implementación de Asterisk en ambos sistemas operativos.

2.1 Organización de los ficheros

Una vez compilado Asterisk, el mismo tendrá una organización de sus ficheros de la siguiente forma:

- **/etc/asterisk** contiene todos los ficheros de configuración de Asterisk.
- **/usr/sbin** contiene todos los ficheros ejecutables y los scripts incluyendo Asterisk, astman, astgenkey, y safe_key.
- **/usr/lib/asterisk** contiene todos los objetos binarios relacionados con Asterisk.
- **/usr/lib/asterisk/modules** contiene los módulos para aplicaciones, drivers de canal, codecs, formato de fichero etc.
- **/usr/include/asterisk** contiene los ficheros de cabecera requeridos para compilar la aplicación.
- **/var/lib/asterisk** contiene los datos variables usados por Asterisk en una operación normal.
- **/var/lib/asterisk/agi-bin** localización de los scripts agi para ser utilizados.
- **var/lib/asterisk/astdb** es la base de datos de Asterisk, el equivalente del registro de Windows.
- **var/lib/asterisk/images** área de almacenamiento de imágenes.
- **var/lib/asterisk/keys** área de almacenamiento para las claves públicas y privadas usadas para la autenticación RSA.
- **var/lib/asterisk/mohmp3** área de almacenamiento para los ficheros mp3
- **var/lib/asterisk/sounds** área de almacenamiento de los sonidos.

2.3 Configuración de Asterisk.

Asterisk presenta una elevada flexibilidad a través de sus ficheros de configuración, localizados en el directorio **/etc/Asterisk**. Su sintaxis de configuración está diseñada para ser editada por software (como por la interfaz de configuración GUI) y puede ser editada manualmente en el mismo fichero. Estos ficheros tienen formato ASCII y están divididos en secciones encerradas entre corchetes. Las palabras claves van separadas por el símbolo igual y el símbolo mayor que “=**>**”. El punto y coma sirve para comentar las líneas así como el carácter **#**. Las líneas en blanco son ignoradas.

Luego de verificar que Asterisk se ha iniciado correctamente, estará listo para configurar la PBX. Primeramente se necesita declararle a Asterisk que tipos de conexiones telefónicas se utilizarán, las cuales serán denominadas canales. El próximo paso será la configuración del plan de discado el cual se encargará de enrutar las llamadas a través de varias aplicaciones hasta su destino final.

2.3.1 Canales.

Un canal es un tipo de conexión a través de la cual se le introducen las llamadas a Asterisk. Un canal puede ser una conexión a un teléfono ordinario, o a una línea telefónica común y corriente, también puede ser una llamada lógica (como las llamadas IP).

Tipos de canales

- **Agent:** Tipo de canal agente, ACD.
- **H.323:** Un canal de Protocolo VoIP.
- **IAX e IAX2:** Protocolo de intercambio interno de Asterisk. (protocolo propio de Asterisk)
- **Local:** Lazo dentro de contextos.
- **MGCP:** Otro protocolo VoIP.
- **Modem:** Para conexión con la ISDN (no es para el uso de Modems.)
- **Phone:** Canal de telefonía Linux.
- **SIP:** El protocolo VoIP comúnmente más utilizado.
- **VPB:** Para conectarse a líneas telefónicas y teléfonos comunes usando tarjetas VoiceTronix.
- **Zap:** Para conectarse a líneas telefónicas y teléfonos comunes usando tarjetas Digium y TDM.
- **Bluetooth:** Le permite el uso de dispositivos Bluetooth
- **CAPI:** Canales CAPI de la ISDN
- **mISDN:** Canales mISDN.

Configuración.

- **Tarjetas de interfaz Digium:** se necesita tener instalado y configurados los drivers de Zaptel para permitirle a Asterisk el uso de las tarjetas Digium mediante los canales Zap. Estos se configuran en **zapata.conf**
- **Tarjetas de interfaz VoiceTronix:** se necesitan tener instalados estos dispositivos para permitirle a Asterisk el acceso a las tarjetas VoiceTronix usando los canales VPB. Estos pueden configurarse en **vpb.conf**.
- **Los Teléfonos SIP:** Para los canales SIP, se configuran en el fichero **sip.conf**.
- **Los Teléfonos o Servidores IAX:** se encuentran en el fichero **iax.conf**.

2.3.2 Configuración del Plan de discado.

El plan de discado es configurado en el fichero **extensions.conf**. Este es el fichero de configuración más importante de Asterisk, el mismo controla todas las llamadas entrantes y salientes que pueden ser enrutadas y manipuladas por el sistema. Este fichero presenta una distribución en la cual las secciones están separadas por corchetes.

Sección de configuración: [general]

Al principio del fichero aparece esta sección donde se configuran algunas características generales tales como:

Static: Esta opción afecta solo la operación del comando **save dialplan** (salvar el plan de discado). El valor predeterminado es *no*.

El comando **save dialplan** sobrescribirá el fichero existente **extensions.conf** por uno nuevo generado desde el actual. La copia del viejo no se efectúa y todos los comentarios en el fichero se pierden.

Writeprotect: Si esta opción esta negada y la opción **static=yes**, entonces usted podrá salvar el plan de discado desde la interfaz de comandos de Asterisk con el comando **save dialplan**. Esta opción por defecto esta negada.

Ejemplo:

```
[general]
static=yes
writeprotect=yes
```

Sección de configuración: [globals]

Luego de la sección [general] aparece esta sección en la cual se definen las constantes o las variables globales con sus valores iniciales.

Los valores actuales de las variables globales pueden ser referenciados dentro del plan de discado usando la siguiente sintaxis: **`\${Nombre_de_Variable}`**. Note que los nombres de las variables globales no se distinguen entre mayúsculas ni minúsculas ejemplo: **`\${MYGLOBAL}`** y **`\${myGLObal}`** son equivalentes.

A continuación se muestra un ejemplo donde se definen variables globales usadas en el plan de discado de Asterisk.

[globals]

; Cuales extensiones timbrarán cuando llegue una llamada entrante.

INCOMING => Zap/3&Zap/4

; Cantidad de tiempo que timbrarán estas extensiones antes de que se active el servicio de correo de voz.

RINGTIME => 3

; Que fichero de sonido se utilizará en el anuncio de correo de voz.

VMANNOUNCE => mysounds/my-vm-announce

; Definición de los canales a los cuales nuestras extensiones están conectadas.

COCINA => Zap/3

CUARTO => Zap/4

SALA => Zap/5

; Que línea(s) usar cuando se desee realizar una llamada afuera del sistema.

SALIDA => Zap/1&Zap/2

Asterisk no conoce intrínsecamente que hacer con estas variables, es tarea suya definir la variable y hacer uso de ella en la forma que desea en el plan de discado.

En este ejemplo, las variables globales han sido escritas con nombres en mayúscula como convención para distinguir fácilmente las variables globales de las variables de canal, las cuales son usualmente escritas mezclando mayúsculas y minúsculas.

2.3.3 Configuración de contextos y extensiones.

Luego de las secciones [general] y [global], el resto del fichero **extensions.conf** está reservado para la definición del plan de discado el cual consiste en una colección de contextos que a la vez cada contexto es una colección de extensiones.

Ejemplo de contexto

Context "default"

Extension	Description
101	Samuel Montejo
102	Carlos Rodríguez
103	Chequeo de correo
104	Salón de Conferencia
0	Operador

En este ejemplo se ha nombrado el contexto por defecto (default) donde las primeras dos extensiones están asociadas a empleados, la tercera extensión está asociada para permitir que cualquiera revise su correo de voz, la cuarta extensión será asociada al salón de conferencias y finalmente la extensión 0 se asocia al operador.

Definición de Extensiones.

Con diferencia de las PBX tradicionales, donde las extensiones están asociadas con teléfonos, interfaces, menús, y demás, una extensión en Asterisk está definida como una lista de comandos a ejecutar. Los comandos generalmente serán ejecutados según el orden de prioridad que tengan, aunque los comandos **Dial** y **Gotof** están habilitados para redireccionar hacia otro comando que tenga mayor prioridad.

Cuando una extensión es marcada, el comando marcado con prioridad 1 es ejecutado, seguido por el comando con prioridad 2 y así sucesivamente hasta que la llamada sea colgada. En la sintaxis del fichero extensions.conf, cada paso de comando en una extensión es escrito de la siguiente forma:

exten => pattern,priority,Command(parameters)

- **Exten:** es el nombre de la extensión, tanto como un nombre de extensión literal o un patrón de extensión. Se repetirá exactamente el mismo nombre para cada comando de la extensión.
- **Priority:** es un número entero utilizado para enumerar los comandos relacionados con la extensión dada. Asterisk comenzará con la ejecución de la prioridad 1. Si no existe una línea con prioridad 1 entonces la extensión no se corresponde con ningún número discado. Luego de ejecutar la prioridad 1 Asterisk ejecutará la prioridad 2 hasta que procese todas las extensiones con prioridades siguientes. Algunos comandos ejecutan saltos entre prioridades dependiendo de la lógica de las prioridades.
- **Command:** es el nombre del comando también llamado aplicación a ejecutar.
- **Parameters:** los parámetros dependen del comando. Algunos comandos no toman parámetros en los cuales se puede omitir los paréntesis.

Ejemplo:

```
exten => 123,1,Answer()
exten => 123,2,Playback(tt-weasels)
exten => 123,3,Voicemail(44)
exten => 123,4,Hangup()
```

Está es la definición de una simple extensión llamada “123”. Cuando una llamada es hecha a la esta extensión, Asterisk esperará hasta que se conteste la llamada, ejecutará un fichero de sonido llamado “tt-weasels”, si no se contesta le dará al usuario la oportunidad de dejar un mensaje de voz y colgará.

Otro ejemplo:

```
exten => 1234/_256NXXXXXX,1,Answer()
```

Con esta opción solo se corresponderá la extensión 1234 si el número de ID es cualquier cosa que comience con 256. Esta opción es muy útil cuando se desea crear servicios prepagados.

Nombres de Patrones de Extensiones.

Las extensiones del plan de discado pueden ser números simples como “**415**” o “**0**”. También pueden ser caracteres alfanuméricos como “**pedro**”, “**Juan**”, “**A95**”. Aunque los teléfonos típicos no pueden marcar una extensión llamada “Juan” (otros si pueden), frecuentemente la lógica del plan de discado conlleva saltos de una extensión a otra distinta, y en estos saltos se pueden definir nombres de extensiones con los nombres que usted desee.

La mayoría de los teléfonos modernos no solo tienen los dígitos del 0 al 9, además tienen * y #. Otros teléfonos (DTMF) tienen dígitos extras tales como A, B, C y D. si se poseen estos tipos de teléfonos en su organización, no existe nada que le impida el uso de estos botones extra de propósito general.

Para usar extensiones que se activen mediante el carácter #, se debe hacer el uso del patrón de extensión. Asterisk no reconoce # como un dígito ordinario aunque aparezca en todos los teléfonos DTMF.

Nombres de extensiones predefinidas para propósitos generales:

i: Invalid	(inválido)
s: Start	(comienzo)
h: Hangup	(colgar)
t: Timeout	(fin de tiempo)
T: AbsoluteTimeout	(fin de tiempo absoluto)
o: Operator	(operador)

Patrones de extensiones:

Los nombres de extensiones no son limitados a simples números, una extensión puede corresponderse con un patrón.

En el fichero **extensions.conf**, un nombre de extensión es un patrón si comienza con el carácter “_”. En un patrón de extensiones el carácter que le sigue tiene un significado especial.

Caracteres especiales que corresponde con patrones.

- **X** se corresponde con cualquier dígito del 0 al 9.
- **Z** se corresponde con cualquier dígito del 1 al 9.
- **N** se corresponde con cualquier dígito del 2 al 9.
- **.** comodín que se corresponde con cualquier carácter o dígito.
- **[1237-9]** se corresponde con cualquier dígito o letra entre corchetes (en este ejemplo 1,2,3,7,8,9).

Considere el siguiente contexto como ejemplo:

Context "routing":

Extension	Description
_61XX	Oficina Economía
_63XX	Oficina Economía
_62XX	Oficina Rector
_7[1-3]XX	Oficina Secretaría
_7[04-9]XX	Oficina Profesores

En este contexto, se decidió que todas sus extensiones fueran de cuatro dígitos de longitud. Si el usuario disca una extensión que comience con 61 o 63 será enviado a la oficina de Economía, cualquiera que comience con 71, 72, 73 se enrutará a la oficina de secretaría y cualquier extensión que se marque con 70, 74, 75, 76, 77, 78, ó 79 se envía a la oficina de los Profesores.

- _NXXXXXX se corresponde con un número telefónico normal de 7 dígitos.
- _1NXXNXXXXXX se corresponde con un código de área y número telefónico normal precedido por uno.
- _9011. Se corresponde con cualquier cadena de al menos 5 caracteres que comienzan con 9011 pero no se corresponde con la misma cadena de cuatro caracteres 9011.
- _# se corresponde con presionar el carácter “#”.

Inclusión de Contexto.

Un contexto de extensión puede incluir el contenido de otro contexto. Considere el siguiente ejemplo:

Context "default":

Extension	Description
101	Samuel Montejo
102	Carlos Rodríguez
0	Operator

Context "local":

Extension	Description
_9NXXXXXX	Local calls
include =>	"default"

Context "longdistance":

Extension	Description
_91NXXNXXXXXX	Long distance calls
include =>	"local"

En este ejemplo se tienen definidas tres extensiones:

- El contexto **default** permite el discado de tres extensiones.
- El contexto local tiene un patrón de extensión que permite solamente el discado de números de siete dígitos (llamadas locales), y además incluye el contexto default, de este modo le permite al usuario discar las extensiones Samuel, Carlos o el operador.
- El contexto long distance tiene un patrón de extensión que permite el discado de larga distancia, pero además incluye el contexto local, permitiendo de este modo al usuario realizar llamadas locales y acceder a las extensiones de Samuel, Carlos y el operador.
- Mediante el uso de los contextos de extensiones se puede manipular cuidadosamente el acceso a los servicios de tarifas.

2.3.4 *Uso de variables en el diseño del Plan de discado.*

Asterisk puede hacer uso de variables globales o específicas de canales como argumento de los comandos. Las variables son referenciadas en el plan de discado (fichero `extensions.conf`) y estas usan la siguiente sintaxis:

`${foo}`

Donde `foo` es el nombre de la variable. El nombre de la variable puede ser cualquier cadena alfanumérica que debe comenzar con una letra. Las variables pueden ser definidas por el usuario o predefinidas del sistema.

Las variables definidas por el usuario no se distinguen entre mayúsculas y minúsculas. `${FOO}` es lo mismo que `${foo}`.

Las variables predefinidas de Asterisk si hay que diferenciar las mayúsculas de las minúsculas. La variable predefinida `${EXTEN}` funcionará mientras que `${exten}` no lo hará.

Existen tres tipos de variables: variables globales, variables de canal, y variables de entorno.

Las variables globales pueden ser definidas en la sección `[general]` del fichero **`extensions.conf`** o utilizando el comando **`SetGlobalVar`**. Una vez definidas, estas pueden ser referenciadas por cualquier canal en cualquier momento.

Las variables de canal son definidas usando el comando **`SetVar`**. Cada canal toma su propio espacio de variables, por lo tanto no hay chance de colisión entre distintas llamadas y la variable es automáticamente destruida cuando se cuelga el canal.

Las variables de entorno proveen un mecanismo para acceder a las variables de entorno de Linux desde dentro de Asterisk.

Si se define una variable de canal con el mismo nombre de una variable global, las referencias a este nombre de variable le devolverán los valores de las variables de canal. Por ejemplo si se define el siguiente contexto con una simple extensión:

```
[FooTest]
exten => 100,1,SetGlobalVar(FOO=5)
exten => 100,2,NoOp(${FOO})
exten => 100,3,NoOp(${foo})
exten => 100,4,SetVar(foo=8)
exten => 100,5,NoOp(${FOO})
exten => 100,6,NoOp(${foo})
```

Si usted marca la extensión 100 y está ejecutando Asterisk con una consola en modo verboso, se podrá observar la siguiente salida.

- Executing SetGlobalVar("Zap/1-1", "FOO=5") in new stack
- Setting global variable 'FOO' to '5'
- Executing NoOp("Zap/1-1", "5") in new stack
- Executing NoOp("Zap/1-1", "5") in new stack
- Executing SetVar("Zap/1-1", "foo=8") in new stack
- Executing NoOp("Zap/1-1", "8") in new stack
- Executing NoOp("Zap/1-1", "8") in new stack

Después de utilizar el comando SetGlobalVar, \${FOO} y \${foo} devuelven los valores de la variable global, en este caso 5. Luego de utilizar el comando SetVar la variable foo toma el valor de 8 pero al hacer referencia de FOO que es una variable global, esta devuelve el valor que tomo de la variable de canal foo.

En los anexos se muestra una lista de las variables de canal predefinidas.

Aplicación de variables específicas.

El comando ChanlsAvail devuelve \${AVAILCHAN}: el primer canal disponible.

El comando Dial toma como entrada la variable \${VXML_URL}

El comando Dial toma entrada de \${ALERT_INFO}: y configura la cadencia de timbrado para los teléfonos Cisco.

El comando Dial devuelve \${CAUSECODE}: si el comando dial falla, esta variable guarda el mensaje de error de la causa del fallo.

El comando Dial devuelve \${DIALSTATUS}

El comando EnumLookup devuelve \${ENUM}: resultado de la búsqueda.

El comando TXTLookup devuelve \${TXTCIDNAME}: resultado de la búsqueda DNS.

El comando MeetMe toma entrada de {MEETME_AGI_BACKGROUND}: es un script AGI a ejecutar.

El comando MeetMe devuelve \${MEETMESECS}: cantidad de Segundos que un usuario permaneció en conferencia.

El comando Hangup devuelve la variable \${PRI_CAUSE}

Variables específicas a utilizar en las Macro.

\${ARG1}: Primer argumento pasado a la macro.

\${ARG2}: Segundo argumento pasado a la macro.

\${MACRO_CONTEXT}: El contexto de la extensión que utiliza la macro.

\${MACRO_EXTEN}: Extensión que utiliza la macro.

\${MACRO_PRIORITY}: Prioridad en la extensión donde se utiliza la macro.

\${MACRO_OFFSET}: Prioridad establecida por la macro donde la ejecución continuará una vez finalizada la macro.

Variables de entorno.

Se puede acceder a las variables de entorno de UNIX mediante la siguiente sintaxis.

`${ENV(foo)}`

`${ENV(ASTERISK_PROMPT)}`: Indicador CLI actual de Asterisk.

`${ENV(RECORDED_FILE)}`: nombre del último fichero salvado por el comando Record.

Manipulación de cadenas

Longitud de cadenas

`${LEN(foo)}`

Devuelve la longitud de de la cadena foo.

Ejemplo:

```
exten => 100,1,SetVar(Fruit=pear)
exten => 100,2,NoOp(${LEN(Fruit)})
exten => 100,3,NoOp(${LEN(${Fruit}}))
```

El primer NoOp mostrara el valor 5 (la longitud de la cadena "fruit"). El segundo NoOp mostrara el valor 4 (la longitud de la cadena "pear", contenido de la variable "fruit").

Subcadenas.

`${foo:offset:length}`

Devuelve una subcadena de la cadena **foo**, comenzando en el desplazamiento **offset** y devuelve los próximos **length** caracteres.

Si el desplazamiento es negativo, la cadena se mostrara de izquierda a derecha.

Si la longitud es omitida o es negativa, entonces se mostrara el resto de la cadena a partir del desplazamiento **offset**.

Ejemplos:

```
${123456789:1}    - devuelve la cadena 23456789
${123456789:-4}   - devuelve la cadena 6789
${123456789:0:3}  - devuelve la cadena 123
${123456789:2:3}  - devuelve la cadena 345
${123456789:-4:3} - devuelve la cadena 678
```

Ejemplos:

exten=>_NXX.,1,SetVar(areacode=\${EXTEN:0:3}) toma los tres primeros dígitos de \${EXTEN}

exten=>_516XXXXXXX,1,Dial(\${EXTEN:3}) toma todos menos los tres primeros dígitos de \${EXTEN}

exten => 100,1,SetVar(whichVowel=4)

exten=> 100,2,SetVar(foo=AEIOU:\${whichVowel}:1) ; le asigna a la variable \${foo} el valor de la letra "U"

Concatenación de cadenas.

\${foo}\${bar}

555\${theNumber}

\${longDistancePrefix}555\${theNumber}

2.3.5 Implementación de macros en Asterisk.

Para utilizar las macros se debe utilizar el comando macro de la siguiente forma:

Macro(nombre_macro,arg1,arg2...)

Este comando ejecutara la macro usando el contexto '**macro-<nombre_macro>**' saltando a la extensión 's' de este contexto, ejecutando cada paso y retornando al contexto llamante cuando finalicen los pasos en esta extensión.

Nota: no podrá usar ninguna otra extensión que no sea la extensión 's' para construir la macro.

La extensión llamante, el contexto y la prioridad son guardadas en las variables **\${MACRO_EXTEN}**, **\${MACRO_CONTEXT}** y **\${MACRO_PRIORITY}** respectivamente.

Los argumentos que se le pasan a la macro para su ejecución se guardan en las variables **\${ARG1}**, **\${ARG2}**, etc. en el contexto de la macro.

Si utiliza el comando Goto() y sale fuera del contexto macro, la macro terminará y el control retornará en la localización referida por el comando Goto().

De otro modo si se guardaron valores en la variable **\${MACRO_OFFSET}**, será el final y entonces la macro tratará de continuar en la prioridad **MACRO_OFFSET + N + 1** si existe, sino terminará en la prioridad **N + 1**.

Códigos de retorno.

La macro devuelve -1 si cualquier paso de la macro devuelve -1 y devuelve cero si devuelven 0 los pasos.

Ejemplo:

```
[macro-stdexten];
; definición estándar de macro.
; ${ARG1} – Dispositivo a timbrar
exten=>s,1,Dial(${ARG1},20) ; timbrará la interfaz, por 20 segundos.
exten=>s,2,Voicemail(u${MACRO_EXTEN}); si no está disponible marcará el
;correo de voz
exten=>s,3,Goto(default,s,1); si presiona la tecla "#", retorna al comienzo.
exten=>s,102,Voicemail(b${MACRO_EXTEN}) ; si está ocupado lo envía al
correo ;de voz o al anuncio de ocupado.
exten=>s,103,Goto(default,s,1) ; si presiona la tecla "#", retorna al comienzo.
Para utilizar esta macro lo hará de la siguiente forma:
exten => 1234,2,Macro(stdexten,sip/7960mark)
```

2.4 Configuración de servicios.

2.4.1 Configuración SIP.

Los canales SIP se Definen en el fichero **sip.conf**. Cada cliente y servidor SIP son identificados de la siguiente forma:

```
[xxx]
type=yyy
parameter1=value
parameter2=value
```

Donde xxx es el nombre de usuario asociado con el cliente SIP, o es un nombre arbitrario usado por el fichero de configuración para referirse a un dispositivo SIP. Propiamente si un teléfono SIP tiene un número de extensión llamado 123, entonces se corresponderá con una entrada en el fichero **sip.conf** llamada 123 la cual deberá estar habilitada en el plan de discado.

Configuración SIP sección [general].

La sección **[general]** del fichero **sip.conf** incluye las siguientes variables:

user: es el ID de usuario para este servidor SIP.
authuser: es una autorización opcional para el usuario del servidor SIP
secret: contraseña del usuario.
host: es el dominio o nombre del host para el servidor SIP.
port: es el Puerto SIP , usualmente es el 5060
allow=<codec>: Permite los codecs en orden de preferencia.
disallow=all: prohíbe todos los codecs (configuración global)
autocreatepeer= yes | no: si está habilitado, cualquiera será capaz de registrarse como un par sin chequeo de credenciales. Es útil para operaciones con SER
bindaddr = 0.0.0.0: dirección IP a conectarse para escuchar.

canreinvite = update | yes | no se puede reinvitar si falla la conexión.

context = <nombrecontexto>: contexto por defecto para las llamadas entrantes en el fichero **extensions.conf**

defaultexpirey=120: longitud por defecto de los registros de entrada/salida.

defaultip: dirección IP del usuario, generalmente es especificada como DYNAMIC.

externip = 200.201.202.203: dirección que se pondrá para los mensajes SIP cuando se encuentra detrás de NAT.

localnet = 192.168.1.0/255.255.255.0: Red local y su mascara.

fromdomain=<domain>: habilitado por defecto para cuando se actúa como cliente.

maxexpirey=3600: longitud de máxima de registros permitidos.

nat= yes | no: cambia el comportamiento de Asterisk cuando se usa detrás de un Firewall NAT.

notifymimeytype=text/plain: permite controlar los tipos de ficheros MIME usados en el correo de voz.

srvlookup = yes|no: habilita la búsqueda por DNS en las llamadas.

tos=<value>: Habilita los parámetros de QoS.

videosupport=yes|no: Activa el soporte de video para SIP.

trustpid = yes|no: si se desea confirmar un ID remoto en conferencias.

realm = my realm: cambiar el dominio de autenticación por defecto de Asterisk.

Amaflags: Categorización para los records CDR.

dtmfmode: como los clientes manipulan la señalización DTMF.

language: el código de lenguaje definido en **indications.conf**

mailbox: extensión del correo de voz.

md5secret: código MD5 del usuario. Puede ser usado en vez de secret.

permit, deny,mask: restricciones de redes y direcciones IP.

restrictcid: (yes/no) cuando se tiene una ID de llamado restringida.

rtptimeout: termina la llamada si hay x segundos de no actividad RPT cuando no se ha colgado

rtpholdtimeout: termina la llamada si hay x segundos de no actividad RTP cuando se está en espera (debe ser mayor que rtptimeout)

musiconhold: habilita la clase de música en espera en llamadas desde teléfonos SIP.

Notas:

Asterisk no soporta SIP sobre TCP. Solamente soporta SIP sobre UDP.

Para los teléfonos Grandstream: debe habilitar **dtmfmode=info**

Asterisk usa el flujo entrante RTP como una fuente de temporización para enviar su flujo de salida. Si el flujo de entrada es interrumpido debido a la supresión de silencio entonces la música en espera será cortada. Concluyendo, no puede usar supresión de silencio. Asegúrese que todos los teléfonos SIP tengan deshabilitada la supresión de silencio.

En los anexos se muestran ejemplos de configuración SIP.

2.4.2 Configuración del correo de voz.

Uno de los servicios más importantes que brinda Asterisk es el correo de voz, el mismo nos brinda la posibilidad de dejarle un mensaje al usuario deseado si el mismo no se encuentra actualmente activo en el sistema. Este servicio se configura en el fichero **voicemail.conf** además de hacer algunas especificaciones en el fichero **extensions.conf**.

El fichero **voicemail.conf**. Presenta el siguiente formato:

```
[general]
setting=valor
...
[zonemessages]
newzonename=país/opciones_de_ciudad.
...
[context_section]
extension_number=>voicemail_password,user_name,user_email_address,user_p
ager_email_address,user_option(s)
...
[another_context_section]
extension_number=>voicemail_password,user_name,user_email_address,user_p
ager_email_address,user_option(s)
...
```

El fichero **voicemail.conf** contiene una sección general, una de zona de mensajes y una o más secciones de contextos definidas para los usuarios. La sección [general] manipula los valores de configuración de información global. Contiene valores los cuales se pueden aplicar como políticas comunes para todos los usuarios.

Parámetros de la sección [general].

attach: este valor hace que asterisk copie el mensaje de voz a un fichero de audio y se lo envíe al usuario como un adjunto en un mensaje de notificación de correo. Attach adquiere solamente dos valores, **yes** o **no**, y por defecto esta desactivado

delete: si esta activado (yes) el mensaje será borrado del buzón de voz después de haberse enviado. Cuando este valor es usado solo, provee de una funcionalidad que le permite al usuario recibir el correo de voz vía correo solamente.

Nota: este comando no siempre funciona como un valor global, es recomendado utilizarlo en cada buzón de voz de los cuales los mensajes serán borrados después de ser enviados.

mailcmd: le permite al administrador cambiar el comando de correos por defecto con un comando definido. Mailcmd toma un valor de cadenas de caracteres como parámetro el cual será la línea de comando deseada a ejecutar cuando el usuario necesite ser notificado con un mensaje de correo de voz. La línea de comandos por defecto es **mailcmd 'usr/sbin/sendmail -t'**

maxsilence: define cuanto tiempo Asterisk esperará por un período continuo de silencio antes de terminar con una llamada entrante de correo de voz. El valor por defecto es 0 el cual significa que el detector de silencio está deshabilitado y el tiempo de espera es infinito. maxsilence toma cero y valores enteros positivos los cuales significan el número de segundos de silencio a esperar antes de desconectarse.

envelope: controla como Asterisk mostrará o no, un mensaje envolvente (date/time) antes de mostrar el mensaje de voz. Esta opción no afecta la operación de la opción envelope del menú avanzado de correo de voz. Toma solamente dos valores, **yes** o **no** y por defecto está activado (yes).

Externnotify: se utiliza si se desea correr un programa externo cada vez que el llamante deja un mensaje de voz para usuario del correo de voz. Externnotify toma una cadena de caracteres como parámetro de línea de comando la cual es el comando deseado para que se ejecute cuando se le deje un mensaje a un usuario.

silencethreshold: le permite al administrador ajustar el umbral de silencio. El valor del umbral de silencio por defecto es 128. Números mayores elevan el umbral y por lo tanto se necesitan más ruidos de fondo para que se restablezca el detector de silencio. Se necesita experimentación cuando se utilice este parámetro para un mejor resultado.

serveremail: este parámetro puede ser usado para identificar la fuente de una notificación de correo de voz. El valor es una cadena la cual puede ser codificada una de dos maneras. Si la cadena es de la forma: [alguien@host.com](#), entonces la cadena puede ser usada como la dirección fuente de todas las notificaciones del correo de voz. Si la cadena es de la forma **alguien**, entonces el nombre de la máquina donde se está ejecutando Asterisk será pospuesto a la cadena después de la inserción del carácter “@”. Ejemplo: [alguien@asteriskserver.com](#).

maxmessage: define la máxima duración de tiempo en segundos de un mensaje entrante. Use este parámetro cuando existan muchos usuarios y el espacio de disco duro sea limitado. El valor por defecto de este parámetro es 0 lo cual significa que no existe duración de tiempo límite para los mensajes.

minmessage: este parámetro puede ser utilizado para eliminar mensajes los cuales sean menores que una duración de tiempo dada en segundos. Por defecto esta en 0, para no limitación.

format: se utiliza para seleccionar los ficheros de formatos de audio que serán usados cuando se guarden los mensajes de correo de voz. El valor es una cadena que define el formato de fichero de audio. La cadena de formato por defecto es **wav49|gsm|wav** lo cual significa que Asterisk puede salvar los mensajes en los tres formatos soportados. Cuando se envía un adjunto, se enviará el formato que primero está definido.

wav49: en este formato el tamaño del fichero será pequeño, la calidad buena y es una buena opción para enviar correos de voz en mensajes de correo electrónico. El fichero tendrá una extensión **.wav**, la cual no tiene problemas para cualquier usuario de Windows.

gsm: el correo guardado en este formato tendrá alrededor del mismo tamaño que y la misma calidad que un fichero con formato wav49. Algunos usuarios tendrán problemas para que su sistema soporte este formato.

wav: este es un formato de sonido no comprimido, por lo cual el fichero es muy grande. No es recomendado.

maxgreet: este parámetro le permite al administrador limitar el tamaño de los mensajes de bienvenida que pueden ser grabados por los usuarios. Se puede usar en sistemas con muchos usuarios y poco espacio en disco. El valor es un número entero que define la máxima cantidad en segundos del mensaje de bienvenida. El valor por defecto es 0 lo cual significa que el mensaje puede ser de cualquier tamaño. Con este parámetro se controla la longitud de los mensajes **unavailable greeting, busy greeting** y mensajes de usuario.

skipms: este parámetro define un intervalo en milisegundos para usarse cuando se desea avanzar adelante o detrás cuando el mensaje de voz se está reproduciendo. El valor entrado debe ser un valor entero positivo y por defecto tiene un valor de 3000 (3 segundos).

maxlogins: este parámetro define el número de reintentos que el usuario tiene para introducir la palabra clave antes que Asterisk lo desconecte. El valor debe ser un entero positivo y por defecto tiene como valor 3.

cidinternalcontexts: define el contexto interno usado para determinar el tipo de anuncio de voz a reproducir cuando se relee el identificador de llamado del envoltorio del mensaje o cuando la característica avanzada **saycid** está habilitada. En caso de una llamada interna, en el indicador se leerá "llamada desde la extensión..." en lugar de "llamada desde 12345678". El valor es una cadena para listar los contextos internos. Cada contexto interno debe ser separado por coma. Por defecto es una cadena vacía sin definir contextos internos.

review: es buena práctica dejar que el usuario pueda revisar su mensaje antes de enviarlo al buzón de correo. Este parámetro solo toma valor de **yes** o **no**. Si esta activado (**yes**) al llamante se le preguntará si desea revisar el mensaje o lo guardará si luego presiona “#”. Si está desactivado, el mensaje será salvado y el sistema de correo de voz desconectará al usuario. Por defecto este parámetro está deshabilitado. Este parámetro esta disponibles solo en versiones de Asterisk que soporten la característica avanzada de correo.

operator: permite que el usuario contacte un operador durante el tiempo en que un mensaje de correo de voz está siendo grabado o una vez que el mensaje de voz ha sido enviado si la opción **review** está activada (yes). Este parámetro toma los valores **yes** o **no**. El operador debe especificar una extensión “o” en el fichero extensión.conf.

saycid: anuncia el número telefónico del usuario llamante, anteriormente a mostrar el mensaje entrante y después de anunciar la fecha y hora en que el mensaje fue recibido. Toma valores de **yes** o **no**. Por defecto esta desactivado

dialout: especifica el contexto a ser usado para enviar las llamadas de salida en la característica del menú avanzado de correo de voz. Este parámetro toma un valor de cadena que significa el contexto a usar. Por defecto es una cadena vacía.

callback: especifica el contexto para utilizar la característica de retorno de llamada. Se le pasa una cadena como valor y por defecto es una cadena vacía.

dbuser: especifica el nombre de usuario de la base de datos MySQL para la aplicación de correo de voz en uso. El valor que toma es una cadena y por defecto su valor es “test”. Asterisk debe ser recompilado para que la característica MySQL pueda utilizarse.

dbpass: especifica la clave para la base de datos MySQL de la aplicación correo de voz en uso. El valor que toma es una cadena y por defecto su valor es “test”

dbhost: especifica el nombre del servidor de la base de datos MySQL para la aplicación correo de voz en uso. El valor que toma es una cadena de caracteres y por defecto está vacía.

Dbname: especifica el nombre de la base de datos MySQL. El valor es una cadena y por defecto tiene: “vmdb”

dboption: especifica la opción de base de datos Postgres. El valor que recibe es una cadena. Tiene por defecto una cadena vacía.

pbxskip: este parámetro cambia la línea Asunto en un mensaje de notificación de correo de voz. Este parámetro toma valores de **yes** o **no**. Por defecto su valor es no. Cuando se activa, la línea asunto: del correo introduce “nuevo mensaje m en

el buzón B". cuando no esta activa se lee en la línea de asunto del correo:
Subject: [PBX]: nuevo mensaje M en buzón B"

fromstring: le permite al administrador sobre escribir una porción de la línea From en los mensajes de notificación del correo de voz. Por defecto Asterisk envía una cadena "From: Asterisk PBX <who>". La línea Asterisk PBX se puede cambiar especificando una nueva con este parámetro

emailsubject: este parámetro sobre escribe completamente la línea Subject: en el mensaje de notificación del correo de voz. El valor pasado es una cadena que contiene el texto que será copiado en la línea subject.

emailbody: este parámetro sobrescribe el texto mensaje normal enviado en el cuerpo del mensaje de notificación de correo de voz. Soporta además variables de sustitución las cuales pueden ser usadas para hacer mensajes más significativos.

exitcontext: es un contexto opcional donde caerá el usuario que ha presionado "*" o "o" para salir del correo de voz.

nextaftercmd: si está activado (yes) después de borrar un mensaje de voz el sistema automáticamente mostrará el próximo mensaje.

Algunas variables a usar en los parámetros **emailsubject** y **emailbody**

VM_NAME
VM_DUR
VM_MSGNUM
VM_MAILBOX
VM_CIDNUM
VM_CIDNAME
VM_CALLERID
VM_DATE

Un ejemplo de algunos parámetros en la sección general.
[general]

; enviar fichero de voz adjunto en la notificación de correo.
attach=yes

; usar el formato wav para todos los mensajes
format=wav

; limitar la longitud máxima del mensaje a 180 segundos

maxmessage=180

; limitar la longitud mínima del mensaje a 3 segundos.

minmessage=3

; anunciar el número del usuario antes de escuchar el mensaje.

saycid=yes

; limitar los reintentos de autenticarse antes de Asterisk desconecte al usuario.

maxlogins=3

; define el contexto interno para las funciones de reporte de identificador de llamante

cidinternalcontexts=house_local,house_toll,house_admin

Parámetros para la sección [zonemessages] en el fichero voicemail.conf

La sección [zonemessages] zona de mensajes es utilizada para definir zonas de tiempo, cambiar la manera en que se realizan los anuncios en determinadas zonas de tiempo. Los usuarios pueden tener diferentes configuraciones de zonas de tiempo y además pueden usar diferentes formas de anunciar las fechas y el tiempo de los mensajes de voz.

Cuando se define una zona de tiempo, se podrá utilizar en buzones individuales especificando la opción **tz=** en los datos del buzón individual.

El formato de [zonemessages] es:

newzonename=Country/City|Options

Newzonename es el nombre para escoger la zona de tiempo deseada. El campo **Country** contiene el nombre del país donde se localiza la zona de tiempo. El campo **City** contiene el nombre de la ciudad del país y el campo **Options** contiene un juego de opciones para escoger y adaptar el formato del anuncio de tiempo.

El formato para los campos de zona del tiempo **Country/City** es el mismo que es usado cuando se instala Linux en la PC. La zona de tiempo que es usada por su PC esta localizada en el siguiente camino **/usr/share/zoneinfo**.

Uno o más ficheros de sonido o opciones de anuncios pueden ser insertadas después del delimitador '|'. Los ficheros de sonido deben ser encerrados por simple comilla ejemplo: **'vm-received'**.

Las opciones de anuncios son cualquier carácter o sustitución de la variable (**\${xxx}**). Cadenas de múltiples caracteres para las opciones de anuncios son permitidas como HM.

Las opciones de los anuncios que se pueden especificar son:

Opción	Descripción
'filename'	Nombre del fichero de sonido (encerrado entre comillas simples.)
\${VAR}	Variable de sustitución.
A o a	Día de la semana (sábado, domingo, lunes...)
B o b o h	Nombre del mes (enero, febrero...)
d o e	Día numérico del mes (primero, segundo, ..., treinta y uno)
Y	Año
l o i	Hora, reloj de 12 horas
H	Hora, reloj de 24 horas
k	Hora, reloj de 24 horas
M	Minutos
P o p	AM o PM
Q	"hoy", "ayer" or ABdY
q	"hoy", "ayer", día de la semana, or ABdY
R	Tiempo de 24 horas incluyendo los minutos

Ejemplo de la sección [zonemessages].

[zonemessages]

;activar el formato de tiempo de 12 horas para la zona del pacífico.

san-diego=America/Tijuana|'vm-received' Q 'digits/at' IMP

; activar el formato de tiempo de 24 horas para la zona del pacífico.

san-diego24=America/Tijuana|'vm-received' Q 'digits/at' R

Parámetros para la sección de [CONTEXTS] del fichero voicemail.conf

La parte final del fichero de configuración voicemail.conf contiene una o varias secciones de contextos donde se definen y configuran los buzones del correo de voz. En una sección de contexto se pueden definir múltiples buzones de voz.

El formato de la sección de contexto es:

[seccion_de_contexto]

**numero_extension=>voicemail_password,user_name,user_email_address,
user_pager_email_address,user_option(s)**

...

La seccion_de_contexto es el número de extensión el cual será asignado a este buzón de voz. Existen cinco parámetros los cuales definen la configuración para esta entrada.

El campo **voicemail_password** contiene la contraseña numérica para este buzón.

El campo **user_name** es una cadena alfanumérica usada para escribir el nombre y el apellido del usuario

El campo **user_email_address** es usado para la dirección de correo del usuario para notificar cuando le llegue un correo de voz.

El campo **pager_email_address** dirección del buscador.

El campo **user_option(s)** puede ser usado para ignorar las opciones por defecto definidas en la sección general o para especificar una zona de tiempo específica para el usuario. Existen nueve pares de valores que se pueden especificar en este campo. Cada par se delimita con las barras (|) y estos son: attach, serveremail, tz, saycid, review, operator, callback, dialout y exitcontext. Con la excepción de tz todos los demás valores funcionan exactamente iguales a los definidos en la sección [general]. El parámetro tz es usado para anular los valores por defecto de la sección [zonemessages]

Ejemplo de la sección de contexto.

; todos los buzones de correo para este sistema estarán en el contexto [default]

[default]

; se define un usuario en la extensión número 123, la contraseña es 2048 y ;el nombre del usuario es Joe. Con una dirección de correo para las ;notificaciones con la zona de tiempo configurada a san-diego y permitiéndole ;adjuntar le mensaje de voz al correo en la notificación.

123=>2048, Joe User, juser@somewhere.net,,tz=san-diego|attach=yes

; definición de un usuario en la extensión 456 con contraseña 4096, con nombre Jane, sin dirección de correo para notificaciones y sin opciones.

456=>4096,Jane User,,,

Note que no se dejan espacios en blanco entre los campos y las opciones.

Voicemail broadcasts

Voicemail broadcasts puede ser usado creado con el comando VoiceMail en el plan de discado. Este se puede combinar con la opción **delete** para borrar el fichero original.

En el fichero **extensions.conf**:
exten => 100,1,VoiceMail(u101&102&103)

en el fichero **voicemail.conf**:

101 => 4242,Group Mailbox,,,delete=1
102 => 4242,Buckaroo Bonzai,buckaroo@bonzai.com
103 => 4242,John Whorphin,john@monkeyboy.com

Esta entrada crea un grupo llamado “Group Mailbox” en el buzón 101, el cual cuando es recibido es copiado a los buzones 102 y 103. Seguido la copia de estos buzones el mensaje original es borrado

Cuando se especifican múltiples buzones, el mensaje ocupado o no disponible será tomado desde el primer buzón especificado. Esto puede servir para guardar un mensaje especial de ocupado o no disponible para el buzón 101 en este ejemplo.

En los anexos se muestran ejemplos de configuración del correo de Voz.

2.4.3 Configuración de Inter Asterisk eXchange (IAX).

Se pretende que Asterisk sea una poderosa herramienta de comunicaciones. Por tanto una característica primaria que se necesita de un protocolo IP para Asterisk es que tenga la capacidad de satisfacer sus propias necesidades y que tenga la suficiente flexibilidad para poderlo usar como un tipo de laboratorio para implementar nuevos conceptos.

Por estas razones es que se utiliza IAX el cual es un protocolo propietario de Asterisk, algunas de sus características son:

- Interoperabilidad con los cortafuegos NAT/PAT. IAX opera a través de cualquier tipo de cortafuegos NAT y PAT incluyendo la capacidad de recibir llamadas y transferirlas a otras estaciones.
- Protocolo de alto rendimiento y bajo costo. Cuando se esta ejecutando en una conexión de bajo ancho de banda, o cuando está ejecutando un elevado número de llamadas, optimiza el ancho de banda. Solamente utiliza una encabezado de 4 bytes.
- Alto soporte de internacionalización. IAX transmite información de idioma, por lo cual en la PBX remota el contenido se puede entregar en el idioma nativo del abonado en ese lugar.
- Encuestado remoto del plan de discado. IAX le permite a la PBX o al teléfono IP la capacidad de conectarse con un número del servidor remoto. Esto le permite a la PBX centralizar el plan de discado.
- Autenticación flexible. IAX soporta autenticación en texto plano, md5, y RSA lo cual provee de un modelo flexible de seguridad para los servicios de registro, autenticación y llamadas de salida.

- Protocolo multimedia. IAX soporta la transmisión de voz, video, imágenes, texto, HTML, DTMF, y URL. Los menús pueden ser presentados de forma audible y visible.
- Recolección estadística de llamadas. IAX recolecta estadísticas del funcionamiento de la red (incluyendo jitter y latencia), así como provee de una medición punto a punto de latencia.
- Comunicación de parámetros de las llamadas. Son comunicados a través de la llamada los parámetros tales como Caller*ID, extensión requerida, contexto requerido, etc.
- Diseño de socket simple. El diseño de socket simple de IAX permite hasta 32768 llamadas a ser multiplexadas.

Convenciones para la configuración de ficheros.

Las líneas entre barras [] representan etiquetas de secciones.

ejemplo [mi_sección]

Las opciones se asignan con el símbolo “=”.

Ejemplo mi_opcción = valor

Algunas veces una opción tendrá un número de valores discretos que puede tomar. En este caso las opciones serán listadas en la documentación encerradas entre barras “[]” y se separan por el símbolo “|”

Ejemplo: mi_opcion = [Valor1|Valor2|Valor3] lo cual significa que a mi_opcion se le pueden asignar los valores “Valor1”, “Valor2” y “ Valor3”.

Los objetos o los pseudo-objetos son instanciados usando “=>”.

Ejemplo: mi_objeto=> parámetro. Se creo un objeto llamado “mi_objeto” con algún parámetro cuya definición será específica de este objeto.

El símbolo de comentario utilizado es el punto y coma “;” por la razón de que el carácter “#” es utilizado como símbolo de extensiones.

Configuración de IAX en Asterisk.

Como todo fichero de configuración de Asterisk, el fichero de configuración de IAX se encuentra en el directorio **/etc/asterisk** específicamente en el fichero **/etc/asterisk/iax.conf**

La configuración de IAX es una colección de secciones, cada una exceptuando a la sección “general” la cual representa una entidad dentro del ámbito de Asterisk.

La primera sección es la sección general, en esta área, se especifican diversos parámetros que afectan el comportamiento del sistema completo. Ejemplo de

estos son las direcciones y los puertos de comunicación, el comportamiento del jitter, los codecs por defecto, los bits TOS, los registros, y demás.

La primera línea de esta sección siempre es:

[general]

A esta sección le sigue la lista de parámetros a configurar:

port = <portnum>: Puerto donde IAX se conectará. El puerto por defecto es el 5036. Se recomienda que no se altere este valor.

bindaddr = <ipaddr>: Permite a IAX conectarse a una dirección IP local específica en vez de conectarse a todas las direcciones. Se utiliza para ampliar la seguridad en la red.

bandwidth=[low|medium|high]: El parámetro ancho de banda inicializa la selección de los codificadores para valores apropiados para un ancho de banda dado. La opción “**high**” habilita todos los codificadores disponibles y es recomendada solo en conexiones de 10 Mbps, 100Mbps o mayores.

La opción “**medium**” elimina los codificadores codecs “signed linear”, “Mu-law”, “A-law”, dejando solamente los codificadores de 32 kbps o menores con el MP3 en casos especiales. Esta opción puede utilizarse en conexiones de banda ancha. La opción “**low**” elimina los formatos ADPCM y MP3 dejando solamente los formatos G.723.1, GSM, y LPC10.

allow = [gsm|lpc10|g723.1|adpcm|ulaw|alaw|mp3|slinear|all]

disallow = [gsm|lpc10|g723.1|adpcm|ulaw|alaw|mp3|slinear|all]: Las opciones “allow” y “disallow” le permiten realizar una selección de uno por uno de los codificadores básicos para extender el ancho de banda inicial. La configuración recomendada es seleccionar “low” en ancho de banda y deseleccionar “disallow” el codificador LPC10 porque no trabaja muy bien.

jitterbuffer = [yes|no]

dropcount = <dropamount>

maxjitterbuffer = <max>

maxexcessbuffer = <max>:

Estos parámetros controlan la operación del jitter. El “**jitterbuffer**” siempre debe habilitarse a menos que se espere utilizar las conexiones sobre una red LAN. El valor de “**dropcount**” es el número máximo de paquetes de voz a permitir una caída menor de 100. Valores útiles de este parámetro son entre 3 y 10. El **maxjitterbuffer** es la cantidad máxima del buffer del jitter permitida para ser usada. El parámetro “**maxexcessbuffer**” es la cantidad máxima de exceso del jitter que es permitida antes que el buffer del jitter sea lentamente encogido para eliminar la latencia.

accountcode = <code>

amaflags = [default|omit|billing|documentation]: Estos parámetros afectan a la generación del registro detallado de llamadas. El **accountcode** activa el código

de cuenta para los registros recibidos por IAX, puede ser sobre controlado en un sistema basado en par-usuario para llamadas entrantes.

El “**amaflags**” controla como el registro es rotulado. “**omit**” causa que no se escriba ningún registro, “**billing**” y “**documentation**” sirven para rotular los registros como facturación o documentación respectivamente y el “**default**” selecciona los que hay por defecto en el sistema.

tos = [lowdelay|throughput|reliability|mincost|none]: IAX puede ocasionalmente activar el bit de tipo de servicio (TOS) para especificar valores para ayudar a mejorar el rendimiento en el ruteo. El valor recomendado es el de “**lowdelay**”, al cual muchos routers le darán prioridades a estos tipos de paquetes mejorando la calidad de la voz.

register => <name>[:<secret>]@<host>[:port]: cualquier número de entradas de registro puede ser instaneada en la sección general. Los registros le permiten a Asterisk notificar a un servidor remoto Asterisk (con una dirección fija), cual es nuestra dirección actual. Para que funcione el sistema de registros, el servidor Asterisk remoto necesitará tener una entrada o par dinámico con el mismo nombre y clave. El nombre es un campo requerido y es el nombre del par remoto que se quiere identificar. Se le puede adicionar una clave aunque no es necesario. Generalmente es una clave compartida entre el servidor local y el cliente remoto. Si la clave está encerrada entre corchetes entonces será interpretada como una llave a usar. En este caso el servidor local deberá tener la llave privada en el fichero (**/var/lib/asterisk/keys/<name>.key**) y el cliente remoto deberá tener la misma llave pública.

El campo “**host**” es un campo requerido el cual significa el nombre de la máquina o la dirección IP del servidor remoto con Asterisk. La especificación de puerto es opcional, si no se especifica por defecto usa el puerto 5036.

Configuración del cliente IAX.

La sección siguiente, luego de la sección general en el fichero **IAX.conf**. Define los usuarios, pares o amigos. Un usuario es alguien que se nos conecta al servidor. Un par es alguien al cual se puede conectar y un amigo simplemente es taquigrafía para crear un usuario y un par con parámetros idénticos (ejemplo: alguien que nos contacte y que se pueda contactarse).

[identifier]: Sección donde identificador puede ser una cadena alfanumérica.

type = [user|peer|friend]: Esta línea le dice a Asterisk como interpretar la entidad.

Campos de usuarios:

context = <context>: una o más líneas de contexto pueden ser especificadas como usuarios dándole así acceso al usuario a acceder o poner llamadas en contextos dados. Los contextos son utilizados por Asterisk para dividir el plan de discado en unidades lógicas cada cual con la habilidad de tener números interpretados diferentemente; tener su propio modelo de seguridad, manipulación

de conmutación auxiliar e incluir otros contextos. La mayoría de los usuarios se le da acceso al contexto por defecto y los usuarios acreditados se le brindan acceso al contexto local.

permit = <ipaddr>/<netmask>

deny = <ipaddr>/<netmask>: reglas de denegación o autorización pueden ser aplicadas a los usuarios, permitiéndole conectarse desde ciertas direcciones IP y denegando las que se especifiquen.

Ejemplo:

permit = 0.0.0.0/0.0.0.0

deny = 192.168.0.0/255.255.255.0 denegará cualquiera que se quiera conectar desde la dirección **192.168.0.0** con máscara 24 bits clase C **255.255.255.0**

Nota: Si se utiliza de esta forma

deny = 192.168.0.0/255.255.255.0

permit = 0.0.0.0/0.0.0.0

No denegará a nadie debido a que la regla final (permit) le permite a cualquiera conectarse sin tener en cuenta la regla anterior. Debe tenerse en cuenta el orden de las reglas.

Si no son listadas reglas de denegación o autorización se asume que cualquiera puede conectarse desde cualquier parte.

callerid = <callerid>: se podrá ignorar la información de Caller*ID pasada por el usuario al servidor.

auth = [md5|plaintext|rsa]: se podrá seleccionar cual método de autenticación es permitido al usuario para autenticarse con el sistema. Se pueden especificar múltiples métodos separados por coma. Si se utiliza md5 o plaintext se le debe proveer una clave al sistema. Si se usa autenticación RSA entonces se deben especificar una o más llaves. Si no se especifica clave ni se especifica método de autenticación entonces no se requiere autenticarse con el servidor.

secret = <secret>: La línea "secret" especifica la clave secreta para los métodos de autenticación "md5" o "texto plano". No se sugiere el uso de "texto plano" (texto plano) solamente si se está debugueando.

inkeys = key1[:key2...]: La línea "inkeys" especifica cual clave se puede usar para autenticar el par remoto. Los ficheros de llaves están en el fichero **/var/lib/asterisk/keys/<name>.pub** los cuales son *llaves públicas*. Las llaves públicas nos son encriptadas con DES3 de esta manera no necesitan inicialización

Configuración del Par.

allow = [gsm|lpc10|g723.1|adpcm|ulaw|alaw|mp3|slinear|all]

disallow = [gsm|lpc10|g723.1|adpcm|ulaw|alaw|mp3|slinear|all]:

Permiten habilitar o deshabilitar los codificadores específicos soportados por las conexiones por pares.

host = [<ipaddr>|dynamic]: Especifica el número IP o el nombre de la máquina remota.

defaultip = <ip*addr>: si la maquina remota usa registro dinámico para conectarse, Asterisk de todas formas le da un ip fijo dado para cuando no se pueda ejecutar la conexión dinámica.

Se pueden adicionar más secciones especificando el contexto, el nombre y la clave usada por la conexión. Se permite además las limitaciones por IP "**permit**" y "**deny**".

Al igual que en las otras secciones se soportan tres métodos de autenticación: md5, texto plano y RSA.

Codificadores Codecs

Los siguientes codificadores son válidos a utilizar con **allow/disallow**.

g723.1
gsm
ulaw
alaw
g726
slinear
lpc10
adpcm
g729
speex
ilbc
h261
h263
all

TOS (tipo de servicio)

Alternativamente se pueden activar estos valores de bits

tos=0x18 ; donde los bits del TOS son combinaciones individuales.

Ejemplo:

tos=0x10 = low delay
tos=0x08 = high throughput
tos=0x04 = high reliability
tos=0x02 = ECT bit set
tos=0x01 = CE bit set

Pueden ser múltiples bits, como:

tos=0x18 ; para activar a ambas low delay y high throughput.

Opciones de calidad

Algunos clientes IAX incluyendo el **firefly**, **diax**, **IAXphone**, **IAXComm** no parecen ser monitoreados con la opción **qualify=1000**. Si se tiene esta opción se mostrará el siguiente mensaje en la consola:

Peer '2004' is now UNREACHABLE!

Para estos tipos de clientes es conveniente desactivar la opción de calidad ejemplo: **qualify=no**.

Marcado sin duplicar información en el fichero **extensions.conf**

Es posible simplificar los parámetros que se le pasan al comando Dial si se guardan en el fichero **IAX.conf** el nombre de usuario, la clave, el número de puerto, el contexto de destino y demás. El comando simplemente se usaría de esta forma:

Dial(IAX2/<peer>/<exten>)

Por ejemplo para llamar a un teléfono IAXtel se puede crear la siguiente entrada en el fichero **iax.conf**

```
[iaxtel-outbound]
type=peer
username=<username>
secret=<secret>
peercontext=iaxtel
host=iaxtel.com
```

Entonces el comando Dial se usaría de la siguiente forma:

exten => _1700XXXXXX,1,Dial(IAX2/iaxtel-outbound/\${EXTEN})

notas:

- Los comandos de registro para conectarse con otro servidor IAX deben estar en la sección **[general]** del fichero **iax.conf**.
- Para habilitar IAX en modo trunk, adicionar la línea **"trunk=yes"** en la definición del fichero IAX.conf.
- No usar a la misma vez **"host=dynamic"** y **"defaultip=172.20.11.25"**, asegúrese de que utilizar uno o el otro solamente.
- Si no usa control de acceso basado en IP para todas las entradas de tipos de usuarios, entonces su fichero iax.conf debe contener una entrada **"guest"** sin clave especificada.

En los anexos se muestran ejemplos de configuración de IAX.

2.4.4 Configuración del servicio de colas.

Las colas de llamadas en Asterisk consisten en:

- Las llamadas entrantes que son colocadas en una cola.
- Agentes que responden a las colas.
- Estrategias de manipulación de la cola y división de la misma en agentes.
- Música a escuchar mientras se espera en la cola.

Las colas se definen en el fichero **queues.conf**.

Agentes.

Los agentes son personas (extensiones) que se dedican a responder llamadas. Asterisk envía las llamadas a cada agente que este libre de recibir llamadas.

Los agentes pueden ser: canales directos, teléfonos conectados a Asterisk, etc.

Los Agentes son definidos en el fichero **agents.conf**.

Los Agentes se registran en extensiones especiales que usa la aplicación **agentlogin**.

Estrategias de manipulación de las colas

ringall: timbrarán en todos los canales disponibles hasta que alguien conteste.

roundrobin: timbrarán por turnos cada canal disponible.

leastrecent: timbrará la última interfaz recientemente llamada por la cola.

fewestcalls: timbrará el canal con pocas llamadas en la cola.

random: timbrará los canales aleatoriamente

Menú para el usuario.

Se podrá definir menús para utilizar mientras el usuario está en espera. Para la creación de este tipo de menú, se podrá usar extensiones de un solo dígito. Para habilitar esta opción se podrá definir el contexto del menú en la configuración de la cola.

En las nuevas versiones de Asterisk se adicionaron opciones de penalidad prioridad, etc.

La nueva adición provee la habilidad de operar las colas con un nivel de prioridad a los actuales métodos FIFO (first in first out). La opción **priority** de poner una llamada en cualquier lugar de la cola menos en el final, en dependencia de su prioridad. La prioridad de una llamada está determinada por una variable de canal especial llamada **QUEUE_PRIO**, donde los valores altos que puede tomar significan una prioridad mayor. Si no se habilita esta variable, entonces todas las llamadas tendrán la misma prioridad, (0) por defecto en (FIFO).

Ejemplo:

```
; clientes Importantes  
exten => 111,1,Playback(welcome)  
exten => 111,2,SetVar(Queue_PRIO=10) ; note el valor de la prioridad  
exten => 111,3,Queue(support)
```

```
; clientes menos importantes  
exten => 112,1,Playback(welcome)  
exten => 112,2,SetVar(Queue_PRIO=5) ; menor prioridad  
exten => 112,3,Queue(support)
```

Configuración del fichero queues.conf

El ACD distribuye las llamadas entrantes en el orden en que arriban al primer canal disponible. El sistema responderá a cada llamada inmediatamente si es necesario las mantendrá en la cola hasta que la pueda direccionar hacia el próximo agente disponible. Mediante este método se puede balancear la carga de trabajo entre los agentes asegurándose que cada usuario reciba un aviso y un servicio profesional.

Asterisk soporta múltiples colas de llamadas, estas pueden ser definidas en el fichero **queues.conf** y pueden ser referenciadas como argumentos a la aplicación **Queue** en el fichero **extensions.conf**.

Los agentes se definen en el fichero **agents.conf**.

En los anexos se muestran ejemplos del servicio de colas configurado.

Configuración de los agentes. agents.conf

Los agentes son seres humanos, personas las cuales sus extensiones son utilizadas para manipular las llamadas. Los agentes se configuran en el fichero **agents.conf** permitiendo así el uso de estos en las colas de llamadas. A un agente se le pueden asignar varias colas y se le puede permitir al agente registrarse desde cualquier extensión.

Ejemplo

Fichero **agents.conf**

```
[agents]
```

```
agent => 1001,4321, Yadel Rodríguez;      extensión, clave, nombre de usuario
```

En el fichero **queues.conf**

```
[cola1]
```

```
member => Agent/1001
```

En el fichero **extensions.conf**

```
exten => 28,1,AgentLogin(1001)
```

```
exten => 29,1,Queue(colas1)
```

2.4.5 Distribución automática de llamadas (ACD)

El ACD es utilizado para realizar la distribución de las llamadas entrantes.

[agents]

Define el tiempo en segundos de auto desconexión si es apropiado. Significa cuanto tiempo el teléfono tendrá que timbrar sin que sea respondida la llamada antes de que el agente automáticamente lo desconecte. Note que si este valor es mayor que el valor de suspensión en la cola, el agente no será removido.

autologoff=15

Define el **ackcall** para requerir un reconocimiento por “#” cuando un agente se registra en el agentcallpark. Por defecto es “yes”.

ackcall=yes

Define la cantidad mínima de tiempo para cuando después de desconectado el agente, pueda recibir nuevas llamadas. Este valor es en milisegundos.

wrapuptime=5000

Define la música en espera por defecto para los agentes.

musiconhold => music_class

musiconhold => default

Beep de sonido opcional para cuando se conecten los agentes.

custom_beep=beep

Agrupación de los agentes

group=3

group=1,2

group=1

Opciones de grabación.

Esta sección esta dedicada a los agentes de grabación de llamadas. Las palabras claves son globales al controlador de canal **chan_agent**.

Habilitar la dirección de grabaciones a los agentes. Por defecto esta desactivada.

recordagentcalls=yes

Configuración del formato que será utilizado para grabara las llamadas (wav, gsm, wav49) por defecto es "wav".

recordformat=gsm

Inserta dentro del campo de usuario CDR un nombre de la grabación creada. Por defecto esta desactivado "off".

createlink=yes

El texto a ser adicionado al nombre de la grabación. Permite formar un link URL

urlprefix=http://host.domain/calls/

Directorio opcional para salvar en el las conversaciones. Por defecto es: /var/spool/asterisk/monitor

savecallsin=/var/calls

Definición de los Agentes.

La última sección contiene la definición de los agentes.

Sintaxis: **agent => agentid,agentpassword,name**

agent => 3000,1234,Samuel Montejo Sanchez

agent => 4002,1234,Ana Maria Casas

agent => 4003,1234,Carlos Rodríguez López

2.4.6 Parqueo de llamadas.

Por defecto la extensión 700 es usada para el parqueo de llamadas. Mientras se está en una conversación, puede presionar # para iniciar la transferencia, entonces dígame la extensión 700. Asterisk le anunciara la extensión de parqueo, más probable la 701 o la 702. Entonces cuelgue, el usuario que llama permanecerá en espera en la extensión de parqueo. Se podrá mover a cualquier teléfono y marcar la extensión 701 reanudando de esta forma la conversación.

Si el usuario que llamó se deja en la extensión un tiempo mayor al especificado, Asterisk entonces volverá a timbrar en la extensión que originalmente se marcó.

Para utilizar en un simple plan de discado edite el fichero features.conf y coloque esta línea en el fichero **extensions.conf**:

include => parkedcalls

si se tiene un plan de discado más complejo y se desea habilitar el comando Goto(), para ir a un manipulador más elaborado de parqueo de llamadas, entonces necesitará incluir el manipulador para la prioridad para poner las llamadas en el parqueo. Ejemplo

exten => somecontext,3,Goto(parkinglot,\${ARG1},1)

...

[parqueo]

exten => s,1,NoOp(once a parked call times out it will resume here)

...

include => parkedcalls

exten => i,1,Playback(pbx-invalidpark)

exten => i,2,Hangup

Configuración del parqueo de llamadas de Asterisk.

En el fichero **features.conf** incluir las siguientes líneas.

Nota el fichero **parking.conf** ha sido renombrado a **features.conf**

parkext => 700; extensión a marcar para parquear una llamada.
parkpos => 701-720 ; en cuales extensiones se parquearán las llamadas.
context => parkedcalls ; en cual contexto se encuentran las llamadas
 parqueadas y necesita incluirse este contexto en el fichero **extensions.conf**
parkingtime => 45 ; Número de segundos en que una llamada puede ser
 parqueada. Por defecto son 45 segundos.

Nota: se debe reiniciar Asterisk después de hacer los cambios en el fichero **features.conf** un simple RELOAD no es suficiente.

2.4.7 Servicio LastCaller.

El servicio de marcado al último llamante puede ser implementado en el fichero **extensions.conf** mediante los siguientes comandos.

```
exten => 1011,1,DBget(lastnumber=lastcaller/${CALLERIDNUM})
exten => 1011,2,Answer
exten => 1011,3,SayDigits(${lastnumber})
exten => 1011,4,Macro(call,${lastnumber})
```

La primera prioridad de la extensión 1011 es asignarle a la variable lastnumber el ID de la última persona que llamó.

La tercera prioridad es decir los dígitos guardados en la variable lastnumber y finalmente en la prioridad 4 hacen uso de la macro call para llamar al número de la última persona que llamó.

2.4.8 Repetición del último número marcado.

Si utiliza esta disposición podrá marcar *5 para repetir el último número que usted marcó. La configuración está basada en varios pasos:

Paso 1: Separar el contexto de marcado para teléfonos/usuarios del “marcado actual” de la macro que será definida en el paso 2.

```
[pbx]
include = apps
include = dialpbx

[dialpbx]
exten = _X.,1,Macro(dial)
[router]
```



```
include = local
include = apps
include = outbound
exten = t,1,Busy
```

Paso 2: Enrutar todos los marcados excepto para las aplicaciones de servicios de códigos verticales, a través de la macro.

[macro-dial]

```
exten = s,1,DBput(RepeatDial/${CALLERIDNUM}=${MACRO_EXTEN})
exten = s,2,Dial(Local/${MACRO_EXTEN}@router)
exten = s,3,Busy
```

nota: esta macro lee valores guardados en la base de datos de Asterisk.

La macro ejecuta el comando 'Dial(Local/<number>@router)' esto implica que el número que se selecciona es marcadable desde el contexto de ruteo como se muestra en el paso 1. Usted podrá cambiar esto si lo desea, si desea redireccionar para afuera puede utilizar Dial(Zap/1/<number>).

Si el comando DBget no encuentra la llave, abortará con un código de resultado igual a 101 siendo el próximo paso la prioridad actual + 101.

Paso 3: adicionar el código de redireccionamiento.

[apps]

```
; repetir el último número marcado
exten = *5,1,DBget(temp=RepeatDial/${CALLERIDNUM})
exten = *5,2,Dial(Local/${temp}@router) ; Last known
; si no encontró la llave de marcado.
exten = *5,102,Congestión
```

nota: Este contexto implementa la funcionalidad para activar el rediscado y los usuarios deben poder acceder a el como se muestra en el paso 1.

2.4.9 Servicio de música en espera.

Edición de los ficheros para habilitar la música en espera.

Mediante este servicio se crea una estrategia para mantener el usuario mientras esta en algún tipo de cola o en espera de algún servicio.

Para configurar este servicio deberá verificar que se tiene instalado el mpg123, una guía detallada de la instalación de este software se encuentra en los anexos. Para configurar el servicio de música en espera edite el fichero **/etc/asterisk/zapata.conf**, y adicione la línea **"musiconhold=default"** en el contexto de los **[canales]**.

En el fichero **/etc/asterisk/musiconhold.conf**, elimine el signo de comentario a la línea que dice **"default => mp3:/var/lib/asterisk/mohmp3"**

Deberá reiniciar Asterisk para recargar los ficheros de configuración **musiconhold.conf**.

Especificando la música.

Los ficheros de ejemplo de música en espera (**/etc/asterisk/musiconhold.conf**) contienen:

[classes]

;default => quietmp3:/var/lib/asterisk/mohmp3

;loud => mp3:/var/lib/asterisk/mohmp3

;random => quietmp3:/var/lib/asterisk/mohmp3,-z

Quítele el comentario a la línea **'default =>'**.

Entonces los ficheros en el directorio: **/var/lib/asterisk/mohmp3** serán escuchados con la música en espera. Si existen múltiples ficheros en el directorio, estos serán escuchados secuencialmente.

Definiendo su propia música en espera.

Si desea tener música de su elección disponible para el servicio de música en espera, deberá crear un subdirectorio en el camino **var/lib/asterisk/mohmp3**, nombrado con la clase del tipo de música que tenga.

Por ejemplo: crear una carpeta con nombre **Tiny_Tim** en el camino **/var/lib/asterisk/mohmp3** y copiando los ficheros MP3 en esta carpeta; quedando de esta forma: **/var/lib/asterisk/mohmp3/Tiny_Tim** luego adicionar al fichero **musiconhold.conf** la siguiente línea

Tiny_Tim => quietmp3:/var/lib/asterisk/mohmp3/Tiny_Tim

Para usar la música en espera, se debe usar el comando **SetMusicOnHold()** antes de la declaración del comando **Dial()**. Ejemplo:

exten => 100,1,SetMusicOnHold(Tiny_Tim)

Exten => 100,2,Dial(Zap/2)

En algunas situaciones es mejor evitar el **MusicOnHold()** cuando:

- Se originan llamadas particulares.
- Se conecta con una extensión particular.
- Se atraviesa una extensión particular cara.
- Se conecta a una conferencia.

2.4.10 Servicio de redireccionamiento de llamadas.

Si usa esta configuración con el teléfono podrá marcar ***21**número*>** para redireccionar inmediatamente la llamada o ***61**número*>** para redireccionar la llamada con espera, también podrá cancelar lo anteriormente expuesto marcando **#21#** or **#61#**.

[macro-stdexten]

; Macro de extensiones estándar con redireccionamiento de llamadas.

; \${ARG1} - Extensión

; \${ARG2} – Dispositivo(s) a timbrar

exten=s,1,DBget(temp=CFIM/\${ARG1}); Toma la clave CFIM, si no existe, ir a la línea 102

exten=s,2,Dial(Local/\${temp}@pbx/n) ; Redireccionamiento incondicional.

exten=s,3,Dial(\${ARG2},20) ; 20 segundos de tiempo espera.

exten=s,4,DBget(temp=CFBS/\${ARG1}); toma la clave CFBS, si no existe ir a la 105

exten=s,5,Dial(Local/\${temp}@pbx/n); Redireccionar en ocupado o no disponible.

; No hay clave CFIM

exten=s,102,Goto(s,3)

; No existe clave CFBS

exten=s,105,Busy

Notas

Esta macro lee variables guardadas en la base de datos de Asterisk.

Esta macro ejecuta el comando **'Dial(Local/<number>@pbx)'** si encuentra un número a redireccionar. Esto implica que el número sea marcadable por el contexto pbx. Si desea cambiar esta configuración para una extensión de salida puede utilizarlo de la siguiente forma **Dial(Zap/1/<number>)** si el comando DBget no encuentra la clave, terminará con el código de resultado 101 haciendo de esta forma que el próximo paso en el plan de discado será la prioridad + 101.

[apps]

; Redireccionamiento incondicional de llamadas

exten => _*21*X.,1,DBput(CFIM/\${CALLERIDNUM})=\${EXTEN:4})

exten => _*21*X.,2,Hangup

exten => #21#,1,DBdel(CFIM/\${CALLERIDNUM})

exten => #21#,2,Hangup

; redireccionamiento de llamadas cuando está ocupado o no disponible.

exten => _*61*X.,1,DBput(CFBS/\${CALLERIDNUM})=\${EXTEN:4})

exten => _*61*X.,2,Hangup

exten => #61#,1,DBdel(CFBS/\${CALLERIDNUM})

exten => #61#,2,Hangup

No podrá usar "#" para adaptadores SIP tales como Grandstream Handytone porque estos no enviarán este carácter. Estos tipos de teléfonos usan este carácter para finalizar el número de teléfono por tanto no se tiene que esperar por el tiempo de finalización.

Por tanto podrá usar algo como el ejemplo siguiente en este tipo de caso.

```
[apps]
; redireccionamiento incondicional de llamada
; crear el redireccionamiento de llamada
exten => _*21*X.,1,Gotolf(${EXTEN:-1} = #?2:3)
exten => _*21*X.,2,StripLSD(1)
exten => _*21*X.,3,DBput(CFIM/${CALLERIDNUM}=${EXTEN:4})
exten => _*21*X.,4,Hangup
;
; borrar el redireccionamiento de llamada.
exten => **21,1,ODBCdel(CFIM/${CALLERIDNUM})
exten => **21,2,Hangup
;
; borrar el redireccionamiento (con #)
exten => **21#,1,Goto(**21,1)
```

Si usa esta forma, entonces cualquier tipo de adaptador SIP podrán trabajar en la misma forma (terminando la cadena con el carácter "#").

Notas

Este contexto implementa la funcionalidad de activar o cancelar el redireccionamiento de llamadas. Necesita ser incluido en el contexto usado para las llamadas de salida.

La variable \${EXTEN:4} toma la extensión actual y le quita los cuatro primeros caracteres.

```
[pbx]
exten=7000,1,Macro(stdexten,7000,MGCP/aaln/1@<myphone>) ; IP10S
exten=7001,1,Macro(stdexten,7001,SIP/oej) ; SIP connection
```

2.4.11 Servicio de Conferencias.

Para configurar el módulo de conferencias de Asterisk deberá editar el fichero **meetme.conf**

Ejemplo

```
[rooms]  
; el uso es conf => Numero_de_conferencia,pincode  
conf => 9000  
conf => 101,123456
```

Salón de Conferencia 101 como se definió anteriormente en el fichero **meetme.conf** con la clave (PIN) 123456

En el plan de discado podrá utilizar lo siguiente:

```
exten => 500,1,MeetMe(101||123456)
```

Salón de conferencia autenticado

```
exten => 18,1,Answer  
exten => 18,2,Wait(1)  
exten => 18,3,Authenticate(5678)  
exten => 18,4,MeetMe(18|p)  
exten => 18,5,Playback(vm-goodbye)  
exten => 18,6,Hangup
```

En los salones de conferencias dinámicos, el usuario deberá entrar el número de salón para crearla

```
exten => 9999,1, Wait(1)  
exten => 9999,2,MeetMe(|Md) ; si adiciona la opción 'e', Asterisk seleccionará el  
número de salón de conferencias para usted.  
; cambie la opción 'd' a 'D' si desea un número de PIN para la conferencia.
```

Capítulo 3



Capítulo 3

Interfaz con la PSTN y prueba de servicios

3.1 El hardware para Asterisk.

Asterisk está diseñado para que se le puedan añadir fácilmente nuevas tecnologías. Asterisk nos permite el procesamiento de los TDM a nivel de software debido a que el hardware TDM utilizado tradicionalmente es muy caro. Para cumplir con las funciones de gateway contra la PSTN se utiliza el hardware Zaptel. Una lista de hardware diseñado para Asterisk se encuentra referenciada en Asterisk Hardware (2005).

3.1.1 Hardware Zaptel.

El hardware Zaptel es comercializado por la firma Digium. El sistema PBX Asterisk está diseñado para trabajar con este tipo de dispositivos por lo cual este hardware esta completamente soportado. Los controladores para el hardware de Digium le permitirán integrar muchos tipos de equipamiento de telefonía tales como dispositivos E1/T1, PSTN, FXO, FXS. En IAFX (2005) se encuentra una lista de tarjetas actualmente disponibles.

En los anexos se muestran los ficheros de configuración de la interfaz ZAPTEL.

3.1.2 X100P y X101P.

Los X100P y las X101P son dispositivos FXO los cuales permitirán conectar a la PBX Asterisk con una línea de la Red de Telefonía Pública.

La X100P se puede conectar a un puerto simple PCI y soporta señalización (LoopStart) FXS y ("Kewlstart") (Loopstar con desconexión y supervisión en puntos lejanos).

Con la X100P Asterisk soporta, llamadas de entrada y de salida además de detección de timbrado y colgado remoto.

Nota: estas tarjetas no pueden ser usadas como dispositivos FXS para conectar teléfonos analógicos a la PBX Asterisk.

3.1.3 S100U – Interfaz de Puerto simple FXS USB.

LA S100U es una interfaz FXS de puerto simple para conectarle al servidor Asterisk un teléfono analógico por el puerto USB. Esto le permitirá utilizar cualquier teléfono analógico para VoIP cuando esté conectado a Asterisk.

3.1.4 TDM 400P.

La TDM400P es una tarjeta PCI 2.2 de longitud media que permite conectar líneas analógicas y teléfonos analógicos estándar con la computadora. La tarjeta utiliza pequeños módulos que activan cuatro puertos en la tarjeta. Los módulos usados determinan el tipo de interfaz FXO o FXS.

La señalización soportada por esta interfaz es configurable por software.

La FXO ha sido configurada y probada en la conexión con un puerto de estación de la PBX **GoldStar GHX-36** y se probó además con la línea de la Red Telefónica Pública Conmutada.

3.1.5 T400P, E400P, TE410P y TE405P – interfaz T1/E1.

Los T400P y E400P son cuatro versiones de puertos de los T100P y los E100P respectivamente. El TE410P es un T1 o E1, seleccionable por tarjeta o por puerto, permitiéndole hacer ambos tipos de formato de señalización en una simple tarjeta. Esta tarjeta soporta bus PCI 3.3v disponible en las nuevas computadoras o en la arquitectura del bus PCI de 64 bits.

3.1.6 TE110P, – interfaz Simple T1/E1.

Es una tarjeta PCI de un solo puerto, de longitud media con características similares a la TE410P, de modo seleccionable para los T1/E1

3.1.7 Valoración del hardware a usar en la solución universitaria.

Aunque la tarjeta TDM400P ha sido satisfactoriamente probada como elemento que cumple con las funciones de gateway contra la PSTN, por razones técnicas y económicas no es el tipo de interfaz que debe cumplir con esta función, esto se debe a que la TDM400P solo soporta 4 puertos, por lo tanto, un servidor Asterisk tendría capacidad para 12 puertos (tres tarjetas) en una solución típica. Esta cantidad de puertos es insuficiente si la PBX Asterisk debe cursar todo el tráfico hacia la PSTN. Por otra parte cuando la cantidad de troncos supera el número de ocho es aconsejable desde el punto de vista económico valorar el uso de un multiplex TDM en la interconexión. De lo anterior se concluye que la interfaz ideal para este escenario puede ser una tarjeta TE110P o una TE410P. La primera opción es válida desde el punto de vista económico hasta 60 canales, para más de 60 canales es preferible el uso de una tarjeta TE410P.

3.2 Prueba de servicios.

De la lista de servicios mencionados en el capítulo 1, han sido configurados y probados satisfactoriamente los siguientes:

- Music on Hold (Música en espera)
- Call Parking (Parqueo de llamadas)
- Call Pickup: (captura de llamada)
- Conferencing (servicio de conferencias):
- IVR (Respuesta de voz interactiva)
- Call Hold: (retención de llamada)
- Unconditional Call Forwarding (Redireccionamiento incondicional)
- No Answer Call Forwarding: (Redireccionamiento por No respuesta)
- Busy Call Forwarding: (Redireccionamiento por ocupado)
- Automatic Redial: (rediscado automático)
- Manual Redial (rediscado manual)
- Call waiting indication: (indicador de llamada en espera)
- Auto attendant (atendido automático)
- Queues (Servicio de colas)

En el capítulo dos y en los anexos aparecen ejemplos de ficheros de configuración para la habilitación los servicios antes mencionados.

Conclusiones



CONCLUSIONES

Los siguientes puntos reflejan de forma precisa el cumplimiento de los objetivos trazados:

- *La PBX Asterisk se encuentra actualmente instalada en el entorno universitario.*
- *El servicio Correo de voz actualmente está funcionando en la UCLV.*
- *Se instaló y realizaron pruebas con el protocolo IAX conectándose con servidores en diferentes puntos fuera de la universidad.*
- *Se hicieron configuraciones y pruebas de servicios y se creó el plan de numeración.*
- *Se instaló y probó la interfaz ZAPTEL TDM400P.*

Recomendaciones



RECOMENDACIONES

De acuerdo con la investigación llevada a cabo se proponen las siguientes recomendaciones:

- Atendiendo al potencial para uso docente que tiene la plataforma de telefonía creada a partir de la PBX Asterisk, se recomienda la preparación de laboratorios que tributen a la asignatura conmutación telefónica.
- Aunque en la literatura consultada aparecen algunos artículos que analizan la capacidad de procesamiento de tráfico de Asterisk, estos son en realidad poco profundos y muy dependientes de una solución particular. Se recomienda realizar dicho análisis para la solución universitaria.
- Se recomienda la realización de pruebas con la interfaz TE410P
- Se recomienda el uso de Asterisk para soportar el tráfico interno de la universidad y valorar su potencial en el tráfico de larga distancia en la conexión con la PSTN atendiendo siempre a las regulaciones legales que limitan esta aplicación.

Bibliografía



Referencias Bibliográficas

1. Asterisk (2005). "Asterisk - the Open Source PBX!" [en línea]. [Consultado 21/4/2005] Disponible en <http://www.asterisk.org/>
2. Asterisk Hardware (2005). "Asterisk Hardware". [en línea]. [Consultado 12/5/2005] Disponible en <http://asterisk.org/index.php?menu=hardware>
3. IAFX, Solutions (2005) "Available Products". [en línea]. [Consultado 4/6/2005] disponible en http://shop.ifax.com/index.php?cPath=32_26_29
4. LinuxMall (2005) "Que es Asterisk" [en línea]. [Consultado 21/4/2005] Disponible en <http://www.linuxmall.cl/asterisk.html>
5. Manesh, Nasser K. (2004) "Asterisk: A Non-Technical Overview" [Consultado 4/5/2005] Disponible en <http://www.millenigence.com/articles/asterisk-non-technical-review.pdf>
6. Powel, Andy. (2003). "Getting Started with Asterisk Version 0.1a" [en línea]. [Consultado 28/5/2005]. Disponible en <http://www.automated.it/guidetoasterisk.htm>
7. Spencer, Mark. (2002) "Introduction to the Asterisk Open Source PBX" [Consultado 6/5/2005] Disponible en http://www.digium.com/downloads/asterisk_whitepaper.pdf
8. Spencer, Mark. (2003) "The Asterisk Handbook Version 2" [Consultado 28/4/2005] Disponible en <http://www.digium.com/handbook-draft.pdf>
9. VoiceTronix (2002) "Why Open Source for Computer Telephony" [en línea]. [Consultado 12/5/2005] Disponible en <http://www.voicetronix.com/open-source.htm>
10. Voip-info. (2005) "Asterisk introduction: An overview for new Asterisk administrators - THE PLACE TO START!" [en línea]. [Consultado 26/5/2005] Disponible en: <http://www.voip-info.org/wiki-Asterisk+introduction>

Anexos



Anexo I

Instalación de Asterisk en Linux RedHat 9.

Asterisk puede ser instalado en cualquier distribución estable de Linux. En este anexo se utilizará la distribución Linux RedHat 9 y se hará referencia a la instalación de los paquetes de Linux necesarios para el desempeño correcto de Asterisk completando los siguientes pasos, todo esto asumiendo que el usuario este familiarizado con la instalación y manipulación de Linux.

Requisitos previos.

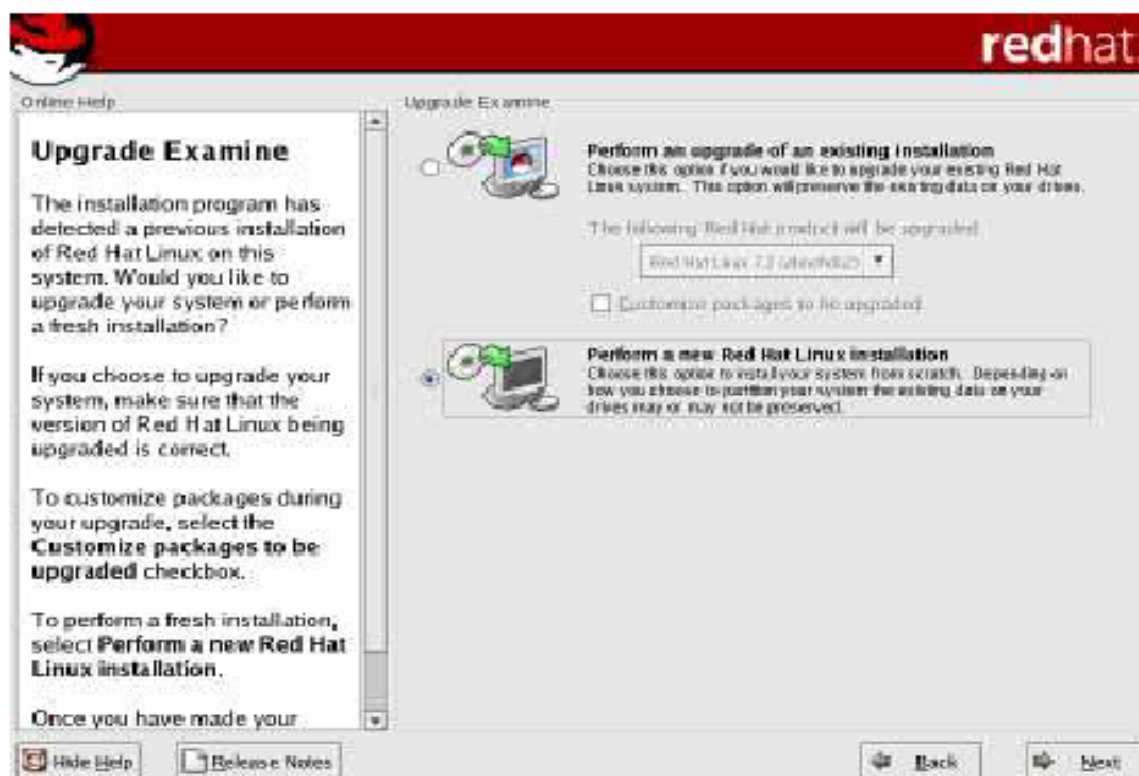
- 1 Discos de instalación de Red Hat Linux 9.
- 2 PC (Como mínimo un PII 350 MegHertz con 64 MB RAM, 2GB de HDD)
- 3 Código fuente del software PBX Asterisk.

Versión del núcleo de Linux (Kernel).

Asterisk está diseñado para trabajar con el núcleo de Linux 2.4, sin embargo consta de algún soporte para el núcleo 2.6, no obstante se recomienda utilizar el núcleo 2.4 si se desea crear un sistema PBX estable con Asterisk.

Instalación los paquetes necesarios para Asterisk en Linux Red Hat 9.

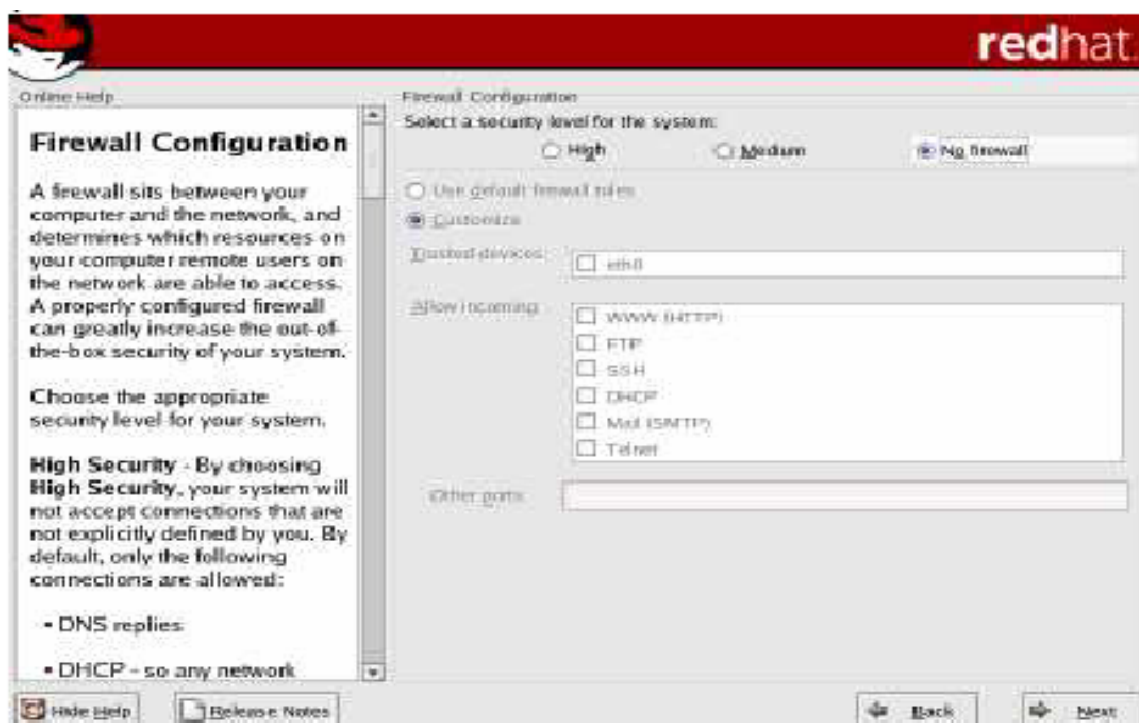
- 1 Despierte la PC con el CD de instalación de Red Hat Linux y proceda a través de los pasos de de instalación seleccionando el teclado, el idioma y el mouse.
- 2 Si en el sistema se detecta una instalación previa de Red Hat se observará la próxima pantalla, seleccione la opción de realizar una nueva instalación



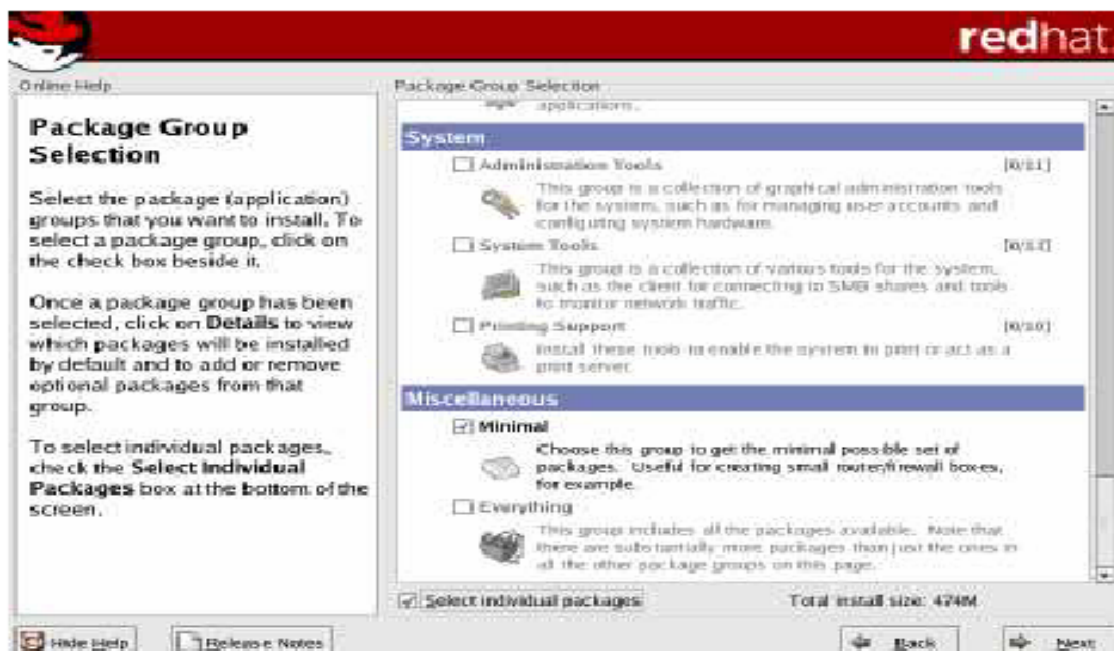
Escoja instalación personalizada (Custom Installation)



- 3 Configurar las particiones del disco, el cargador inicial (boot loader) así como la configuración de la red que su máquina necesite.
- 4 En la pantalla de firewall seleccione **No firewall**.



- 5 Configure el lenguaje, la zona de tiempo, la clave de administración (password de root), y las configuraciones de autenticación de su sistema.
- 6 En el grupo de selección paquetes seleccione la instalación mínima y asegúrese de marcar la opción de paquetes individuales.



7 Buscar en los paquetes individuales

- **Development Tools**
- **Kernel Development**
- **Text Based Internet (optional but advised)**

8 Seleccione los siguientes paquetes:

- **Bison** (development tools)
- **Cvs** (development tools)
- **Gcc** (development languages)
- **Kernel-source**
- **Libtermcap-devel** (developments libraries)
- **Newt-devel** (developments libraries)
- **Ncurses-devel** (developments libraries)
- **readline-devel** (developments libraries)
- **ncurse C++ devel** (developments libraries)
- **Openssl-devel** (developments libraries)
- **Ncurses4**
- **Openssl096b** (system environments libraries)
- **Readline41** (system environments libraries)
- **SOX**

9 En la siguiente pantalla se mostraran las dependencias requeridas por los paquetes, instale los paquetes de las dependencias. Para comenzar la instalación click en <next>.

10 Al completar la instalación el CD automáticamente será expulsado, click en <next> para reiniciar.

Instalaciones adicionales.

Es posible que se desee el servicio de música en espera, para ello se debe instalar el siguiente paquete: **mpg123/mpg123-0.59r.tar.gz**

Luego de copiarlo lo puede descompactar en /usr/src/ y lo puede compilar de la siguiente forma con los comandos:

```
Make linux
Make install
```

Procedimiento de instalación del hardware de Zaptel y la PBX Asterisk.

1. Entre al sistema como root y corra los siguientes comandos: (El carácter ~ significa que el comentario no es parte de la línea de comandos.)
2. Copie el código fuente en el subdirectorio `/usr/src/zaptel`
3. Para compilar el código fuente corra los siguientes comandos:

```
# cd zaptel
# make install      ~ aproximadamente un minuto
# cd ../libpri
# make install      ~ aproximadamente menos de un minuto
# cd ../asterisk
# make install      ~ aproximadamente un 10 minutos dependiendo del procesador
# make samples      ~ aproximadamente menos de un minuto
# make config       ~ aproximadamente menos de un minuto
```

- Una vez terminado este proceso, si no hay problemas usted tiene Asterisk compilada y lista para correr.
- Todo lo que se tiene para configurar su hardware de telefonía y los ficheros de configuración se encuentran localizados en `/etc/Asterisk`
- Si desea tener scripts de inicio instalados para que el hardware de zaptel sea cargado automáticamente en tiempo de booteo, corra los siguientes comandos.

```
#cd /usr/src/zaptel
#make config
```
- Para chequear que el hardware esta correctamente cargado, y para diagnósticos, use el comando `./ztcfg -v` o `./zttool`

Anexo II

Instalación de Asterisk en Linux Knoppix basado en Debian 3.0.

Knoppix es una distribución demo de un sistema GNU/Linux bastante interesante basado en una DEBIAN 3.0 cuya principal característica es que se puede ejecutar totalmente desde CD, sin utilizar nada del software instalado en el sistema en el que se está ejecutando. De esta manera se acaba con la "difícil" operación de instalación de Linux. Además esto supone una gran portabilidad, ya se puede mover el sistema Linux personalizado a donde se necesite.

Otra de las características remarcables de Knoppix es que autodetecta gran parte del Hardware cuando se inicia, con lo que se ahorra bastante trabajo (y tiempo). También cabe destacar el paquete de aplicaciones que se incluye en Knoppix. Es francamente completo y variado.

Se podrá encontrar desde soluciones ofimáticas completas, a herramientas de programación y desarrollo, juegos, multimedia... Además cabe destacar la posibilidad de crear un Knoppix personalizado.

Pero ahora viene lo mejor. Algunos usuarios, tras probar Knoppix, deciden instalarlo permanentemente en el su sistema. Este HOWTO da un ejemplo sencillo de como llevar a cabo este proceso de instalación paso a paso, con la gracia añadida de que correrá más rápido que desde CD.

Requerimientos del sistema

Para instalar y ejecutar Knoppix en tu ordenador, necesitarás:

Un procesador de la clase Pentium, preferiblemente a 300MHz+ 64MB RAM

Una partición vacía en tu disco duro, aproximadamente de 3GB + espacio para crear la partición Swap que generalmente es el doble de tamaño de la RAM instalada en su PC.

Instalar Knoppix en el Disco Duro.

Cargue el CD de Knoppix.

Cuando aparezca el prompt de carga, seleccione el idioma. En nuestro caso sería español, por lo tanto se escribirá:

boot: knoppix lang=es

Espera hasta que el sistema haya sido cargado totalmente, incluido el escritorio KDE.

Pulse CTRL-ALT-F1, para obtener una consola de root (en Knoppix no existen contraseñas de root).

Escribe el comando: **knx-hdinstall**

Ahora no hay más que seguir los pasos de instalación guiada en un menú. Esto incluye:

Crear una partición de Linux (al menos 3 GB)

Crear una partición swap (como máximo 256 MB) o el doble de la cantidad de memoria RAM instalada en la PC.

Montar la partición de Linux como root.

Inicializar la partición swap.
Copiar todos los archivos requeridos (se hace automáticamente)
Configurar los parámetros de red.
Fijar los passwords de root y un primer usuario del sistema.
Configurar el bootloader (LILO o Grub)
Reiniciar sin el CD.

Cuando se haya recargado Knoppix desde el disco duro, haga click en el icono KDE Control Center de la barra de tareas en la parte inferior de la pantalla (un icono de un monitor en color con una tarjeta encima)
Una vez en el Centro de Control, haga click en Personliche Einstellungen (personalizar distribución)
Haz click en Land (país) und Sprache (idioma)
Elige tu idioma
Andwenden en la parte inferior de la ventana
Cierre y reinicie el Centro de Control.
Haga click en Peripherals (periféricos), después Keyboard (teclado), elige su tipo de teclado. Click OK.
Presiona CTRL-ALT-F2 para tener otra consola de root, y entra como root (usando la palabra clave que elegiste durante la instalación)
(De forma opcional) ==> escribe apt-get update (y presiona ENTER). Esto actualizará la lista de paquetes que tengas, llevará en torno a 5- 10 minutos (debes tener conexión a Internet).

Listo, ya Linux está instalado.

Luego de instalado el Knoppix en su HD se procederá a copiar y compilar los ficheros de Asterisk como se describe en el anexo anterior.

Anexo III

Configurando un cliente X-Lite para Asterisk.

1) Adicionar una entrada de cliente para la configuración SIP de Asterisk's.
Para editar el fichero de configuración SIP el cual se encuentra en:
/etc/asterisk/sip.conf utilice el siguiente comando:
#sudo vi /etc/asterisk/sip.conf

Para adicionar una entrada para cada cliente X-Lite siga el siguiente ejemplo:

```
[12345]                                ;Cliente X-Lite 12345
type=friend
secret=blah                          ; palabra clave a utilizar
auth=md5                             ; autenticación que se utilizará
nat=yes                              ; asumimos que los clientes están usando NAT
host=dynamic                         ; asignación dinámica de IP
reinvite=no
canreinvite=no                       ; siempre que vaya a través de Asterisk
qualify=1000
dtmfmode=inband
callerid="Oswald Caballero A" <12345>
disallow=all
allow=gsm                            ; adicionar cualquiera de los codecs que se tenga
context=Estudiantes                 ; usar el entorno existente
```

2) Adicionar extensiones para los clientes de X-Lite.

Para editar los ficheros de extensión buscar el fichero
/etc/asterisk/extensions.conf
#sudo vi /etc/asterisk/extensions.conf

Para entrar extensiones, por ejemplo:
[Estudiantes] ; Nuestro entorno para los clientes X-Lite.

```
; Tomar los 5 números sin tomar el cero
exten => _[123456789]XXXX,1,NoOp("call for "${EXTEN}")
exten => _[123456789]XXXX,2,Dial(SIP/${EXTEN},60,tr)
exten => _[123456789]XXXX,3,Congestion
```

Se puede entrar alternativamente otra extensión para cada cliente.
Salvar los cambios.

3) Recargue la nueva configuración con el siguiente comando en la consola de Asterisk
switch1*CLI> reload

Anexo IV

Variables de canal Predefinidas.

Existen algunas variables de canal predefinidas por Asterisk a las cuales se le puede referenciar en la definición el plan de discado. Las variables predefinidas de Asterisk son sensibles a la diferencia entre mayúsculas y minúsculas. Generalmente estas se escriben en mayúsculas para diferenciarlas.

Ejemplo:

\${CONTEXT}: El nombre del contexto actual.

\${DATETIME}: Fecha y hora actual en el formato: YYYY-MM-DD_HH:MM:SS

\${TIMESTAMP}: Fecha y hora actual en el formato: YYYYMMDD-HHMMSS

\${EPOCH}: Época actual estilo UNIX(números de segundos desde 1ero de enero de 1970)

\${EXTEN}: Extensión actual.

\${INVALID_EXTEN}: Extensión solicitada al redireccionar a la extensión inválida.

\${LANGUAGE}: Lenguaje actual

\${PRIORITY}: Prioridad actual.

\${CALLINGPRES}: Variable de presentación para llamadas entrantes

\${CALLERID}: Identidad del llamante actual (número y nombre).

\${CALLERIDNAME}: El nombre del llamante.

\${CALLERIDNUM}: El número del llamante.

\${CHANNEL}: Nombre de canal actual.

\${RDNIS}: El redireccionador DNIS actual, identificador de llamante para redireccionar la llamada.

\${SIPDOMAIN}: Dominio destino del contexto SIP de las llamadas entrantes.

\${SIP_CODEC}: Usado para habilitar el codificador SIP para una llamada.

\${SIPCALLID}: Encabezado del identificador de llamante SIP.

\${SIPUSERAGENT}: Encabezado del agente de usuario SIP.

\${UNIQUEID}: identificador de llamada SIP exclusivo.

\${HANGUPCAUSE}: Causa de colgado, código de retorno del comando Hangup

\${DNID}: identificador de número discado.

\${DIALEDPEERNAME}: nombre del par llamado.

\${DIALEDPEERNUMBER}: número del par llamado.

\${DIALEDTIME}: tiempo de discado del número marcado.

\${ANSWEREDTIME}: tiempo de demora de las llamadas respondidas.

\${DIALSTATUS}: Estado de la llamada.

\${ACCOUNTCODE}: Código de cuenta.

\${MEETMESECS}: Cantidad de segundos en que un usuario participa en una conferencia MeetMe.

\${TXTCIDNAME}: Resultado de la aplicación TXTCIDName

Anexo V

Ejemplo de configuración de la interfaz ZAPTEL.

```
# La línea build-out (o LBO) es un entero, de la siguiente tabla:
# 0: 0 db (CSU) / 0-133 feet (DSX-1)
# 1: 133-266 feet (DSX-1)
# 2: 266-399 feet (DSX-1)
# 3: 399-533 feet (DSX-1)
# 4: 533-655 feet (DSX-1)
# 5: -7.5db (CSU)
# 6: -15db (CSU)
# 7: -22.5db (CSU)

# El entramado es uno de "d4" o "esf" para T1 o "cas" o "ccs" para E1
# Nota: "d4" puede ser referenciado como "sf" o "superframe"

# El codificador es uno de "ami" o "b8zs" para T1 o "ami" o "hdb3" para E1

# Los E1 pueden tener claves "crc4" adicionales para habilitar el chequeo CRC4

#span=1,0,0,esf,b8zs
#span=2,1,0,esf,b8zs
#span=3,0,0,ccs,hdb3,crc4

# Definición dinámica de extensión en la forma: dynamic=,,,

# usar "0" para deshabilitar la fuente de tiempo o priorizarlo como primario, secundario
etc. Note que debe tener un dispositivo ZAPTEL real si no está usando el tiempo externo.

# dynamic=eth,eth0/00:02:b3:35:43:9c,24,0

# Definición para el uso de canales.
# los dispositivos válidos son:

# "e&m" : Canal(es) que son señalizados usando señalización E&M
# "fxsls" : Canales que son señalizados usando protocolo FXS Loopstart.
# "fxsgs" : Canales que son señalizados usando protocolo FXS Groundstart.
# "fxsks" : Canales que son señalizados usando protocolo FXS Koolstart
# "fxols" : Canales que son señalizados usando protocolo FXO Loopstart.
# "fxogs" : Canales que son señalizados usando protocolo FXO Groundstart
# "fxoks" : Canales que son señalizados usando protocolo FXO Koolstartl.

# La lista de canales es una lista separada por coma de los rangos de canales, ejemplo
# 1,3,5 (canal uno, tres y cinco)
# 16-23, 29 (canal 16 al 23, y canal 29)

# Ejemplo completo
# e&m=1-12
# nethdlc=13-24
```

```
# fxsls=25,26,27,28
# fxols=29-32
#fxoks=1-24
#bchan=25-47
#dchan=48
#fxols=1-12
#fxols=13-24
#e&m=25-29
#nethdlc=30-33
#clear=44
#clear=45
#clear=46
#clear=47
#fcshdlc=48
#dacs=1-24:48
#dacsrbs=1-24:48
```

se puede precargar algunas zonas de tonos para prevenir que las sobrescriban otros
*#usuarios (si se le permite que usuarios no root abran la interfaz /dev/zap/**
#El formato es "loadzone=" donde la zona es una palabra de dos letras para el código
#del país. Se puede especificar además una zona por defecto de la forma:
#"defaultzone=" una lista de zonas se encuentra en el fichero e zaptel/zonedata.c

```
fxols=1
fxsls=4
loadzone = us
```

```
#loadzone = us-old
#loadzone=gr
#loadzone=it
#loadzone=fr
#loadzone=de
#loadzone=uk
#loadzone=fi
#loadzone=jp
#loadzone=sp
#loadzone=no
defaultzone=us
```

Anexo VI

Ejemplo de Configuración del correo de Voz en la UCLV.

[general]

```
; formato por defecto para escribir los ficheros de voz.
format=wav49|gsm|wav
serveremail=asterisk
; el correo será adjuntado.
attach=yes
; longitud máxima del mensaje de voz.
maxmessage=180
; longitud máxima de los mensajes de saludos.
maxgreet=60
; milisegundos para saltar forward/back cuando este en rew/ff
skipms=3000
; cantidad de segundos de silencio antes de terminar de grabar el mensaje.
maxsilence=10
; umbral de silencio
silencethreshold=128
; número máximo de intentos de registros fallidos
maxlogins=3
; evitar la cadena "[PBX]:" en los títulos de mensajes.
pbxskip=yes
; cambiar la cadena From: en el correo
fromstring=Correo de Voz
; cambiar el cuerpo del mensaje, usando variables: VM_NAME, VM_DUR,
VM_MSGNUM, VM_MAILBOX, VM_CALLERID, VM_DATE

emailbody=Estimado(a) ${VM_NAME}:\n\n\tSolamente se quiere que usted
sepa que tiene un mensaje de ${VM_DUR} de duración (numero
${VM_MSGNUM})\n en el buzón de voz ${VM_MAILBOX} de
${VM_CALLERID}, con fecha ${VM_DATE} así que usted podrá chequearlo
en cualquier oportunidad. Gracias!\n\n\t\t\t\t\t--Servicio de Correo de
Voz\n\n\t\t\t\t\t Facultad de Ing. Eléctrica\n
```

[zonemessages]

```
eastern=America/New_York|'vm-received' Q 'digits/at' IMp
central=America/Chicago|'vm-received' Q 'digits/at' IMp
central24=America/Chicago|'vm-received' q 'digits/at' H 'digits/hundred' M
'hours'
```

; cada buzón de correo será listado de la forma:

<mailbox>=<password>,<name>,<email>,<pager_email>,<options>

[default]

;1234 => 4242,Example Mailbox,root@localhost

9000 => 1111,Roberto Hiribarne Guedes,hiribarne@localhost.localdomain

9001 => 1111,Vitalio Alfonso Reguera,vitalio@localhost.localdomain

9002 => 1111,José Chaljub Duarte,chaldjub@localhost.localdomain

9003 => 1111,Carlos Alberto Rodríguez Lopez,carlos@localhost.localdomain

9004 => 1111,Roberto Jiménez,jiménez@localhost.localdomain

9005 => 1111,Hiram del Castillo,Hiram@localhost.localdomain

9006 => 1111,Héctor Cruz, hector@localhost.localdomain

9007 => 1111,Maria del Carmen Casas,mary@localhost.localdomain

9008 => 1111,Oswald Caballero Avendaño,oswaldc@localhost.localdomain

9009 => 1111,SamuelMontejoS,samuel@localhost.localdomain

9010 => 1111,Alberto Taboada,taboada@localhost.localdomain

9011 => 1111,Alain Yumar,Alain@localhost.localdomain

9012 => 1111,Julio Antonio Rodríguez, Julio@localhost.localdomain

En el fichero **sip.conf** se debe adicionar las líneas siguientes de los usuarios:

[9008]

type=friend

host=dynamic

defaultip=172.20.12.1

username=oswaldc

secret=oswaldc

dtmfmode=rfc2833

mailbox=9008 ; número de buzón de correo

context=sip

callerid="Oswald Caballero Avendaño" <9008>

[9009]

type=friend

host=dynamic

defaultip=172.20.11.31

username=Samuel

secret=samuel

dtmfmode=rfc2833

callgroup=1

pickupgroup=1,2

mailbox=9009

context=sip

callerid="Samuel Montejo Sanchez" <9009>

;

Anexo VII

Ejemplo de configuración de IAX

Fichero **iax.conf**

[general]

port=5036

bandwidth=high

disallow=lpc10

tos=lowdelay

[guest]

type=user

context=from-iaxtel

callerid="John Todd"

[nufone] ; Nufone es un servicio experimental de IAX.

type=friend

secret=password

context=NANPA

host=switch-1.nufone.net

disallow=all

allow=gsm

Otro ejemplo

En el fichero **iax.conf**

[name] ; contexto del servidor iax que será llamado

type=friend ; tipo define si recibe llamada, solo puede llamar o si emite y recibe
; llamadas friend es el último caso

host=202.193.55.251 ; IP del servidor iax que será llamado

regexten=1234

secret=moofoo

context=sip

permit=0.0.0.0/0.0.0.0

trunk=yes

En el fichero **extensions.conf**

exten => _1700XXX,1,Dial(IAX2/name/\${EXTEN:4}@sip)

; en este caso será llamada la extensión de cuatro dígitos que suceda a 1700

; Ej. 17004444, será llamado en el servidor iax 202.193.55.251 la extensión 4444

; perteneciente al contexto sip

exten => _1800X.,1,Dial(IAX2/name/4444@sip)

; este realizará lo mismo de un modo menos elegante

Anexo VIII

Ejemplo de configuración SIP

Cada cliente SIP que se conecta a Asterisk necesita una definición en el fichero **SIP.CONF**.

[grandstream1]

```
type=friend           ;puede ser "friend" (peer+user), "peer" o "user"
context=from-sip
username=grandstream1 ;usualmente se corresponde con el título de sección sip.
fromuser=grandstream1 ; requerido por FWD
callerid=John Doe <1234> ; ID del usuario.
host=192.168.0.23      ; dirección IP estática del usuario SIP.
nat=no                ;no hay cortafuegos NAT entre el teléfono y Asterisk
canreinvite=yes       ;permite tráfico de voz RTP
dtmfmode=info         ;se usa cualquiera RFC2833 o INFO para el tono Budge.
mailbox=1234@default  ;buzón de correo de voz para el contexto "default".
disallow=all          ;necesita deshabilitar todo antes de usar allow=
;allow=alaw            ;codecs soportados por asterisk.
;allow=g723.1          ;Asterisk solo soporta g723.1
;allow=g729
```

[xlite1]

```
;Desactivar la supresión de silencio en el cliente X-Lite ("Transmit Silence"=YES).
type=friend
username=xlite1
callerid="Jane Smith" <5678>
host=dynamic
nat=yes               ; en este caso el cliente X-Lite está detrás de un ruteador NAT
canreinvite=no        ; se deshabilita si se usa ruteador NAT.
disallow=all
allow=gsm              ; el GSM consume menor ancho de banda que el ulaw.
allow=ulaw
allow=alaw
```

[user1_snomsip]

```
type=friend
secret=blah
host=dynamic
dtmfmode=inband       ; puede ser inband, rfc2833, o info
defaultip=192.168.0.59
mailbox=1234,2345      ; buzón de voz
disallow=all
allow=ulaw
allow=alaw
```

Luego de definir estas cuentas de clientes SIP, en el fichero sip.conf, usted será capaz de registrarse con el servidor Asterisk y realizar llamadas. Para recibir llamadas usted debe configurar las extensiones en el fichero extensions.conf.

Ejemplo:

exten => 1010,1, Dial(SIP/xlite1,10,t)

Si alguien llama a la extensión 1010, el cliente SIP registrado como xlite1 es marcado y listo para recibir la llamada.

Notas

La variable **\${VXML_URL}** podrá ser usada para adicionarle artículos adicionales al encabezado To: los valores serán adicionados al encabezado To: del SIP después del punto y coma.

La variable **\${ALERT_INFO}** puede ser usada para crear un nuevo encabezado llamado Alert-Info el cual puede ser usado para crear llamadas distintivas en los dispositivos telefónicos Cisco versión 6.0. Para los Cisco versión 7940/60 Alert-Info puede tener los siguientes valores:

- **Bellcore-BusyVerify**
- **Bellcore-Stutter**
- **Bellcore-MsgWaiting**
- **Bellcore-dr1**
- **Bellcore-dr2**
- **Bellcore-dr3**
- **Bellcore-dr4**
- **Bellcore-dr5**

Anexo IX

Ejemplo de configuración del servicio de colas.

Fichero **queues.conf**

[general]

[default]

; ejemplo simple cola de llamada

[markq]

; Musiconhold especifica cual música aplicar para esta cola de llamadas particular.

musiconhold = default

; se puede especificar un anuncio el cual será anunciado al miembro tan pronto

; como responda la llamada, típicamente para indicar en que cola será respondida

; esta llamada

announce = queue-markq

; estrategia de timbrado a usar

strategy = ringall

; puede especificarse un contexto en el cual el usuario puede marcar una

; extensión de un solo dígito mientras está en la cola para salir de la cola y

; alcanzar la extensión del contexto especificado.

context = qoutcon

; cuanto tiempo se especificará para que el teléfono timbre antes de considerar

; tiempo fallido en segundos.

timeout = 15

; cuanto tiempo esperar antes de tratar con todos los agentes otra vez.

retry = 5

; después de una llamada satisfactoria, cuanto tiempo se esperara antes de

; enviarle a un potencial miembro libre otra llamada (por defecto es 0, no espera).

wrapuptime=15

; Número máximo de personas esperando en la cola. (se active en 0 para usar

; número indeterminado).

maxlen = 0

; con que frecuencia anunciar la posición de la cola o el tiempo estimado al usuario. (0=off)

announce-frequency = 90

; usar tiempo estimado de espera en los anuncios de posicionamiento

; el tiempo de espera será anunciado como un tiempo estimado menor que 2 minutos

;announce-holdtime = yes|no|once

; redondeo de tiempo para los segundos

; si es un valor distinto de cero, entonces se anunciarán los segundos

; redondeados al número definido.

announce-round-seconds = 10

; use estos ficheros de sonido para hacer los anuncios de posicionamiento.

; los ficheros por defecto son listados debajo, estos podrán ser cambiados a conveniencia del idioma utilizado si lo necesita.

queue-youarenext = queue-youarenext ; usted es el próximo en la línea.

queue-thereare = queue-thereare ; hay en la pila tantos usuarios.

queue-callswaiting = queue-callswaiting ; llamadas esperando.

queue-holdtime = queue-holdtime ; el tiempo estimado es:

queue-minutes = queue-minutes ; minutos en la cola

queue-seconds = queue-seconds ; segundos

queue-thankyou = queue-thankyou ; gracias por su paciencia.

queue-less-than = queue-less-than ; menos que ;

; las llamadas pueden ser grabadas usando el monitor de recursos de Asterisk.

; se puede activar esta opción desde dentro de la aplicación queue para comenzar

; a grabar cuando una llamada es realizada satisfactoriamente. No se grabará

; mientras se esta escuchando la música en espera.

; se puede especificar el nombre de fichero de monitoreo con la variable:

SetVar(MONITOR_FILENAME=foo)

; de otra forma usará la variable \${UNIQUEID}

monitor-format = gsm|wav|wav49

; si se desea que los ficheros se unan cuando finalicen las llamadas active la

; siguiente opción:

monitor-join = yes

;

;

; joinempty = yes

;

; si se desea eliminar usuarios de la cola si no existe agentes presentes.

leavewhenempty = yes

; Asterisk puede generar eventos AgentCalled cuando un agente esta trabajando.

eventwhencalled = yes

; cada miembro de esta cola de llamadas es listado en líneas separadas en la

; forma tecnología/cadena_de_discado. Miembro significa un usuario normal de la

; cola. Se le puede especificar después de la coma una penalidad.

member => Zap/1
member => Zap/2
member => Agent/1001
member => Agent/1002
member => Agent/@1 ; cualquier agente del grupo 1
member => Agent/:1,1 ; cualquier agente del grupo 1, esperando por el
; primero disponible pero considerando la penalidad.

Nota: en el parámetro de penalidad puede tener agentes que son menos probables de tomar llamadas ejemplo en una cola de ventas, se tienen a los vendedores sin penalidad, a la recepcionista con penalidad 1 y al equipo técnico con penalidad 2.

Si incluyen grupos en la definición de la cola la llamada se enruta en el orden del grupo independiente de la estrategia especificada.

Ejemplo

member=Agent/@1 ; un grupo
member=Agent/501 ; un simple agente.

Anexo X

Configuración del servicio música en espera.

Para configurar el servicio de música en espera antes deberá verificar que tenga instalado el programa mpg123. Si no lo tiene instalado podrá bajar una copia del sitio <http://www.mpg123.de/cgi-bin/sitexplorer.cgi?mpg123/>. Una versión de este programa que esta probada con Asterisk es la versión 0.59r.

Una vez en la máquina debe proceder a descomprimirlo con el comando:

tar -zxvf mpg123-<version>.tar.gz

Luego deberá posicionarse en la carpeta mediante el comando `cd /mpg123`

Una vez en este directorio, utilice el comando “**make**” para ver la lista de las plataformas de hardware soportado así como los sistemas operativos. Luego utilice el comando “**make**” con la opción que necesite. Ejemplo **make Linux**.

Si el comando “**make**” falla entonces trate primero con la opción “**make clean**” luego pruebe la opción “**make linux-devel**” (esta opción fue probada exitosamente en un linux Debian 2.4.26-1-686).

Si la compilación resulto exitosa entonces deberá usar el comando “**make install**” para instalar los ficheros binarios y las páginas de manuales en el directorio `/usr/local`

Deberá crear los enlaces en el directorio `/usr/bin/` con el comando **ln -s /usr/local/bin/mpg123 /usr/bin/mpg123** este enlace es necesario para que Asterisk trabaje con el mpg123.

Ejemplos:

Configurar las extensiones en el fichero **extensions.conf** para probar la música en espera.

[local]

exten => 6601,1,Answer

exten => 6601,2,WaitMusicOnHold(30)

Nota: recuerde ejecutar el comando `answer` antes de usar el `MusicOnHold()`.

Fichero musiconhold.conf

[classes]

default => custom:/var/lib/asterisk/mohmp3/music,/usr/bin/mpg123 -q -r 8000 -f 50000 -b 2048 --mono -s

;default => mp3:/var/lib/asterisk/mohmp3/music

;music => quietmp3:/var/lib/asterisk/mohmp3/music

;loud => mp3:/var/lib/asterisk/mohmp3/music

;random => quietmp3:/var/lib/asterisk/mohmp3,-z

;unbuffered => mp3nb:/var/lib/asterisk/mohmp3

;quietunbuf => quietmp3nb:/var/lib/asterisk/mohmp3

;manual => custom:/var/lib/asterisk/mohmp3,/usr/bin/mpg123 -q -r 8000 -f 8192 -b 2048 --mono -s

Abreviaturas



Abreviaturas

ACD	(Automatic Call Distribution)
ADPCM	(Adaptive Differential Pulse Code Modulation)
ADSI	(Analog Display Service Interface)
AGI	(Asterisk Gateway Interface)
API	(Application Programming Interface)
ASCII	(American Standard Code for Interchange Information)
AU	(Audio file in UNIX)
CAPI	(Common Application Programming Interface)
CFBS	(Call Forward when Busy database family)
CFIM	(Call Forward database family for labels)
CD	(Call Deflection)
CD	(Compact Disk)
CES	(Circuit Emulation Service)
CLIP	(Calling Line Identification Presentation)
CLIR	(Calling Line Identification Restriction)
CVS	(Concurrent Version System)
DID	(Direct Inward Dialing)
DOS	(Disk Operating System)
DND	(Do not Disturb)
DTMF	(Dual Tone Multi Frequency)
E1	(European PDH Signal level)
ECT	(Explicit Call Transfer)
FXO	(Foreign eXchange Office)
FXS	(Foreign eXchange Station)
FWD	(Forward)
GNU	(Gnu's Not Unix)
GSM	(Global System Mobile)
GUI	(Graphical User Interface)
HDD	(Hard Disk Driver)

HTML	(Hyper Text Markup Language)
IAX	(Inter Asterisk eXchange)
IP	(Internet Protocol)
ISDN	(Integrated Services Digital Network)
IVR	(Interactive Voice Response)
LILO	(LIinux LOader)
MD5	(Message Digest encryption algorithm)
MGCP	(Media Gateway Control Protocol)
MP3	(Motion Picture Expert Group Audio Stream layer 3)
MWI	(Message Waiting Indication)
NAT	(Network Address Translator)
PBX	(Private Branch eXchange)
PC	(Personal Computer)
PCI	(Peripheral Component Interconnect)
PDH	(Plesiosynchronous Digital Hierarchy)
PERL	(Practical Extraction and Report Language)
PHP	(Personal Home Page)
PSTN	(Public Switched telephone Network)
RAM	(Random Access Memory)
RSA	(Rivest Shamir Adelman public key encryption algorithm)
SIP	(Session Initiation Protocol)
TDM	(Time Division Multiplexing)
TOS	(Type Of Service)
URL	(Uniform Resource Locator)
USB	(Universal Serial Bus)
VOIP	(Voice over Internet Protocol)
WAV	(Waveform Audio)
ZAP	(Zapata interface)