

**UNIVERSIDAD NACIONAL DE INGENIERÍA**  
**FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**



**TÍTULO:**

**ESTUDIO Y ANÁLISIS DE LA IMPLEMENTACIÓN DE LA  
PLATAFORMA DE COMUNICACIONES DE CALL CENTER  
BASADO EN UNA RED VoIP CON SOFTWARE LIBRE  
ASTERISK**

**PARA OPTAR EL TÍTULO DE:  
INGENIERO ELECTRÓNICO**

**PRESENTADO POR:  
FERNANDO ENRIQUE GIBSON CHILET NÁJERA**

**PROMOCIÓN:  
2002-II**

**LIMA – PERÚ  
2008**

**ESTUDIO Y ANÁLISIS DE LA IMPLEMENTACIÓN DE LA  
PLATAFORMA DE COMUNICACIONES DE CALL CENTER  
BASADO EN UNA RED VoIP CON SOFTWARE LIBRE  
ASTERISK**

## **DEDICATORIA**

---

Agradezco de manera especial a mis padres por haber formado y apoyado mi vocación por las ciencias, a mis hermanas por su apoyo moral en todo momento y a todos los amigos con los que compartí las aulas de estudio.

---

## **SUMARIO**

La tecnología busca generar más rentabilidad para los negocios y es dentro de ese marco que una clase de negocio en particular emerge como uno de los más prometedores, se trata de los Call Centers o Contact Centers. Las empresas que brindan servicios de Call Center y Contac Center buscan ofrecer la mayor cantidad de ventajas con las que actualmente cuentan las plataformas de comunicaciones y al mismo tiempo tratan de emplear el menor presupuesto posible para gastos en tecnología de las telecomunicaciones.

Es por esta necesidad que surge una plataforma de comunicaciones telefónica que brinda todas las características exigidas para este tipo de negocios y que no requiere más que un mínimo gasto de hardware y software además de conocimientos no complejos de programación nativa sobre esta. Se trata de Asterisk, una Plataforma de comunicaciones que has sido desarrollada sobre software libre y cuya finalidad principal es reducir los costos de la migración de la telefonía convencional hacia la Telefonía IP que es la tendencia actual en todo negocio.

La Plataforma de Comunicaciones Asterisk está tomando un porcentaje importante del mercado de las telecomunicaciones y es que una empresa que se inicia en el negocio de los de Call Centers, comenzará desplegando una pequeña pero significativa cantidad de posiciones de atención y pensar en adquirir Centrales Telefónicas como Nortel, Avaya o Cisco escapan de su presupuesto. Con esta premisa, la solución lógica a implementar es Asterisk, central telefónica con la que pueden obtener casi todos los beneficios que obtendrían con otra marca de renombre por un bajo costo.

Lo que hace a Asterisk bajo en costos es la utilización de la telefonía IP sumándole soluciones particulares para la telefonía tradicional, y además, debido a que se trata de un Plataforma casi en su totalidad implementada en software de código abierto el hardware requerido para su despliegue es mínimo.

## ÍNDICE

<b>INTRODUCCIÓN</b>	<b>2</b>
<b>CAPÍTULO I</b>	
<b>PLANTEAMIENTO DE INGENIERÍA DEL PROBLEMA</b>	<b>4</b>
1.1. Descripción del Problema.	4
1.2. Objetivos del trabajo.	4
1.3. Evaluación del problema.	5
1.4. Limitaciones de trabajo.	5
1.5. Síntesis del trabajo.	6
<b>CAPÍTULO II</b>	
<b>MARCO TEÓRICO CONCEPTUAL</b>	<b>8</b>
2.1. Marco Histórico	8
2.2. Base Teórica	9
2.2.1. Definición de Call Center	9
2.2.2. Teoría de encolamiento.	9
2.2.3. Tecnología base para Call Center	10
2.2.4. Elementos adicionales para el funcionamiento de un Call Center.	13
2.2.5. Fundamentos de Voz sobre IP.	15
2.2.6. Protocolos VoIP.	18
2.2.7. Protocolo de Transporte en tiempo real RTP.	18
2.2.8. El estandar H.323.	19
2.2.9. El estandar SIP.	21
2.2.10. Calidad de Servicio.	25
2.2.11. Fundamentos de Asterisk.	27
2.3. Términos y Configuraciones en Asterisk.	30
2.3.1. Configuración inicial de Asterisk.	30
2.3.2. Especificaciones de IAX.	34
2.3.3. Fundamentos del Dial Plan.	35

<b>CAPÍTULO III</b>	
<b>METODOLOGÍA PARA EL DISEÑO DE LA SOLUCIÓN</b>	<b>44</b>
3.1. Estudio de la solución.	44
3.1.1. Selección del servidor para la PBX Asterisk.	44
3.1.2. Aspectos de performance.	45
3.1.3. Selección de Procesador.	46
3.1.4. Selección de Placa Madre.	46
3.1.5 Hardware de telefonía.	46
3.2. Instalando los componentes.	48
3.2.1. Instalando Asterisk.	48
<b>CAPÍTULO IV</b>	
<b>ANÁLISIS Y DISEÑO PARA LA IMPLEMENTACIÓN</b>	<b>52</b>
4.1. Análisis descriptivo del caso de estudio.	52
4.1.1. Gestión de llamadas Inbound.	52
4.1.2. Gestión de llamadas OUT.	53
4.1.3. Grabación de voz.	53
4.1.4. Aspectos técnicos.	53
4.2. Análisis de la configuración de la PBX Asterisk para la atención.	54
4.3. Presentación de la solución.	57
4.3.1. Esquema de la Red de datos.	57
4.3.2. Esquema de la red de voz.	58
4.3.3. Software y Programación.	59
4.4. Presupuesto y tiempo de ejecución.	69
<b>CONCLUSIONES</b>	<b>72</b>
<b>ANEXO A</b>	<b>74</b>
<b>HOJA DE ESPECIFICACIONES TARJETAS DIGIUM E1</b>	<b>75</b>
<b>ANEXO B</b>	<b>77</b>
<b>HOJA DE ESPECIFICACIONES TARJETAS DIGIUM 410 ANALÓGICAS</b>	<b>78</b>
<b>BIBLIOGRAFIA</b>	<b>79</b>

## **PRÓLOGO**

En el mercado actual de las telecomunicaciones, las empresas medianas o grandes pueden justificar gastos relativamente grandes para hacerse de Centrales telefónicas de reconocidas marcas. Sin embargo las empresas que no disponen de un presupuesto tan grande, deben hacer esfuerzos por tener los mismos servicios de una Central Telefónica. Si nos referimos a los Call Centers, se necesita que como mínimo la central telefónica pueda manejar ACD (Distribución automática de llamada), IVR (Recursos de voz interactivo) Buzón de voz entre otros.

La Plataforma asterisk puede brindar todas las características anteriormente mencionadas por un costo accesible y justamente esto es lo que finalmente puede decidir la opción a la hora de elegir. Pero como hace asterisk para poder brindar las mismas ventajas de la Telefonía actual que Centrales de marcas reconocidas. Asterisk es una Plataforma que trabaja netamente con el protocolo IP lo que hace que no requiera grandes gastos en Hardware para la telefonía tradicional TDM, sin embargo asterisk también puede manejar la telefonía tradicional lo que lo hace aún más atractivo.

Para poder entender el funcionamiento y características de Asterisk, es necesario comprender como trabaja esta Plataforma utilizando voz sobre IP, y para esto se debe tener un marco teórico adecuado luego de lo cual se debe comprender la lógica de funcionamiento que también será detallado a lo largo de este trabajo.

## INTRODUCCIÓN

El presente informe tiene por finalidad brindar los conocimientos necesarios para diseñar y soportar la plataforma tecnológica de un Call Center basada en Asterisk. Un Call Center se define como una oficina centralizada cuyo propósito es recibir y/o transmitir un gran volumen de requerimientos y/o consultas a través de líneas telefónicas. La tecnología de Voice over IP (VoIP), cuyo origen se remonta muchos años atrás, ha cobrado una importancia tremenda en los últimos años, dando lugar a grandes esfuerzos e inversiones que, sin duda, están revolucionando las redes telefónicas y, sobre todo, los servicios y aplicaciones que ésta nos ofrece.

Asterisk es un proyecto Open Source basado en licencia GNU (Libre), que conforma una plataforma Private Branch eXchange (PBX) e Interactive Voice Response (IVR) híbrida Time Division Multiplexing (TDM) y voz sobre paquetes. Es probablemente el software de telecomunicaciones más poderoso, flexible y extensible disponible.

Es necesario recalcar que Asterisk es una solución casi ideal para pequeños y medianos despliegues pero debido a que esta solución es software no es recomendada para altas exigencias a diferencia de Marcas reconocidas cuyo Hardware está pensando para grandes esfuerzos que en algunos casos implican redundancia, supervivencia, procesamiento descentralizado entre otros.

La estructura del presente trabajo busca comprender como trabaja Asterisk y como se puede aprovechar las bondades de esta Plataforma para la implementación de un Call Center. Para tal efecto, se ha planificado una estructura que facilite la comprensión de esta tecnología. Como inicio, se cubrirán todos los aspectos a tener en cuenta sobre la evolución de la telefonía, de la tradicional a la telefonía IP, con sus respectivas implicancias, haciendo énfasis en los protocolos usados en voz sobre IP que permitirán obtener la base de conocimientos necesarios para entender la tecnología de Asterisk.



Seguidamente, se trataran todos los aspectos de la Plataforma asterisk, desde la forma de instalación y los paquetes a instalarse hasta los patrones y programación para obtener configuraciones básicas para el uso en despliegues de Call Centers y otros.

Se plantea un caso de estudio en la que se proyecta los recursos a usarse para la implementación de un Call Center con requerimientos establecidos. Aquí se procura cubrir las necesidades más comunes para un Call Center, desde la programación de su IVR hasta el Hardware necesario para el despliegue de la solución tecnológica completa. Además se presentan cuadros de costos referenciales para tener una visión económica de la accesibilidad de la solución.

Luego del análisis se detalla a pleno la solución en Asterisk. Las tarjetas que se requieren, las configuraciones necesarias para la atención de llamadas, los planes de numeración entre otros. Esto también incluye la proyección de la red de datos y la red de voz sobre la que se desplegará el Call Center con PBX Asterisk.

Finalmente, se detallan las conclusiones obtenidas del diseño y la proyección de esta solución. Lógicamente, hay que tener en cuenta que la solución se despliega dentro de un marco teórico y de la experiencia obtenida en instalación, configuración y administración de Centrales Telefónicas, en especial de la PBX Asterisk. Por lo que queda como proyecto aparte la implementación real basada en este trabajo.

Las fuentes de información sobre Asterisk es abundante y variada, esto se debe a que como se trata de un proyecto de código abierto, todos los implicados en el mundo de las telecomunicaciones en especial los ingenieros que ven telefonía, Call Centers, o programación depositan su grano de arena para contribuir a la continua mejora de Asterisk. Sin embargo es importante tener muy en cuenta que Asterisk no pertenece a una marca en particular y con esto quiero decir que a diferencia de recibir soporte de fabricante cuando nos enfrentamos a un problema de alto nivel en nuestra Central telefónica de una reconocida marca, en el caso de Asterisk nadie se hará responsable por los problemas que puedan generarse, solo consultar o buscar información en foros o sitios web donde hay bancos de casos y problemas será el recurso que se tenga hábil. Quizás este tema sea uno de los puntos débiles de Asterisk pero no por eso deja de ser una buena oportunidad de reducir los altos costos de telecomunicaciones.

## **CAPÍTULO I**

### **PLANTEAMIENTO DE INGENIERÍA DEL PROBLEMA**

#### **1.1. Descripción del Problema.**

La mayoría de las empresas en la actualidad requieren de sistemas que les brinden las mismas facilidades de una central telefónica (y más aún) pero de manera privada. Esto es, un equipo que les permita establecer llamadas dentro de una misma área: llámese edificio, nave industrial, ó bodega; a distancias considerables y al mismo tiempo tengan el acceso a realizar llamadas a través de la red pública telefónica. Justamente esos equipos son los que se denominan PBX (Private Branch Exchange – rama privada de intercambio. Un tipo particular de empresas que se dedican a brindar servicios de Call Center requieren de las funcionalidades avanzadas de las PBX y es aquí donde se plantea el inconveniente, ¿Qué central se adecúa mejor al trabajo de un Call Center?.

#### **1.2. Objetivos del trabajo.**

El presente trabajo tiene como objetivos, los que se describen a continuación:

- Analizar el entorno de la telefonía tradicional y la telefonía IP, estableciendo las ventajas de la migración a esta última. Dar a conocer las bases sobre las que se ha desarrollado la Plataforma de comunicaciones Asterisk así como también las fuentes necesarias para su implementación.
- Detallar las funcionalidades que se pueden tener sobre una PBX Asterisk así como la forma en que se programan la mayoría de ellas.
- Establecer los parámetros de diseño a tener en cuenta en una solución de Voz sobre IP con Asterisk teniendo como caso de estudio un Call Center.
- Dar a conocer los factores que se deben tener en cuenta para la correcta selección de una Plataforma telefónica Asterisk.
- Describir los pasos necesarios para la implementación de un sistema Asterisk y la configuración asociada a esta para un proyecto en particular.

Presentar referencias sobre hardware de telefonía que se emplea con la Plataforma Asterisk para completar la integración con los sistemas de telefonía tradicional.

### **1.3. Evaluación del Problema**

Las centrales telefónicas han ido desarrollando mejoras en su sistema de conmutación de llamadas o matriz de conmutación. Además de añadir partes o componentes que pueden integrarse a redes de voz IP lo cual los hace sistemas completos. Cuando se trata de una empresa de tipo Call Center, la problemática radica en seleccionar la PBX (Central Telefónica) que disponga de ciertas características mínimas como: gran flexibilidad, escalabilidad, operabilidad y sobre todo que no sea muy costosa. Como ya se ha indicado, la elección en este tipo de casos se inclina por la Plataforma Asterisk. Entonces, la evaluación a realizarse es: ¿Puede Asterisk cumplir con todos los requerimientos que necesita un Call Center?, ¿Cómo puedo implementar Asterisk como la Plataforma telefónica de un call Center?

La correcta evaluación de estos puntos será soportada en el presente trabajo y consistirá en los siguientes puntos:

- Realizaremos el enfoque sobre un marco teórico para el entendimiento de la telefonía IP que permitirá conocer por que Asterisk puede trabajar de una manera simple con el uso de Voz sobre IP.
- Se presentarán los conceptos básicos relacionados a Asterisk que comprenden su propia definición y su estructura.
- Se profundizarán los conocimientos sobre la forma en que se puede implementar y administrar Asterisk.
- Se cubrirán los parámetros de diseño para la red de datos y la red de voz.
- Se presenta la programación necesaria para configurar la atención del Call Center según las especificaciones requeridas.

La figura 1.1 presenta un esquema de Plataforma de Call Center.

### **1.4. Limitaciones del trabajo**

El presente trabajo tiene algunas limitaciones que se presentan a continuación:

- El trabajo en su totalidad se presenta dentro de un marco teórico quedando la implementación del mismo como otro proyecto.
- La inclinación por la implementación de un Call Center fue determinada por la experiencia cursada en ese campo donde las implementaciones con Asterisk

han sido constantes, sin embargo, como en todo trabajo teórico, es posible que sea necesario realizar ajustes sobre los cálculos teóricos.

El caso de estudio en sí tiene cierto nivel de complejidad pero no se trata de una implementación a gran escala.

Debido a que no se puede cubrir todos los temas con el detalle necesario, se han incluido los puntos más importantes de cada uno, haciendo énfasis en la forma de implementar y administrar la PBX Asterisk, lo cual deja solo unas descripciones y consideraciones de diseño para el tema de la red de datos.

Sobre el tema eléctrico, no se ha dado alcances ya que prácticamente se trataría de un proyecto en paralelo.

Solo se han realizado algunos cálculos aproximados con respecto al costo de la implementación sin entrara a un detalle minucioso.

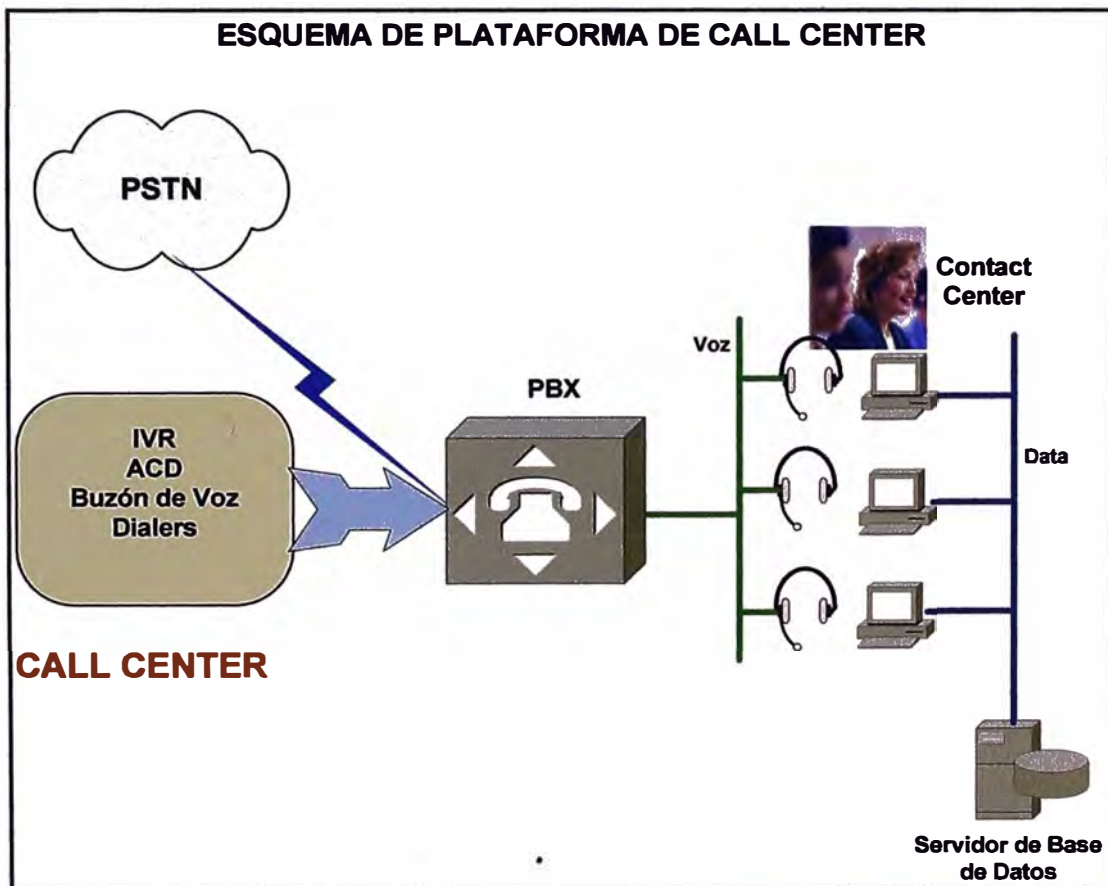


Fig. 1.1 Esquema de Plataforma de Call Center

## 1.5. Síntesis del Trabajo

Este trabajo se puede resumir en el diseño de comunicaciones de VoIP mediante una PBX Asterisk con el objetivo de cumplir los requerimiento de un Call Center. Este proyecto de diseño, surgió a partir del trabajo que he realizado en el marco de la

práctica profesional llevada a cabo en una empresa dedicada al rubro de Call Center que vio oportunidad de reducir costos con el uso de la Plataforma Asterisk.

Para tales efectos, el trabajo recuerda los principales conceptos de la Telefonía IP cubriendo los protocolos H.323, SIP e IAX. Estos conceptos son los que permiten entender el funcionamiento de la Plataforma Asterisk y es en esta parte donde se conoce la estructura de este sistema así como su modo de funcionamiento y operación. Debido a que nuestra meta final es cumplir con los requerimientos de un Call Center, es necesario entender que es un Call Center y cuales son sus necesidades. Estas pasan por requerir las mismas funcionalidades que puede brindar una PBX de un fabricante reconocido, es decir (Distribución Automática de llamadas, Respuesta Interactiva de Voz, Buzón de Voz, etc.). Con esta premisa, enfocamos como es que Asterisk puede dar este tipo de tratamientos usando sencillos comandos de programación. Finalmente, aplicamos estos conocimientos para proceder a implementar el caso de estudio en los que se selecciona el Hardware necesario que depende de los requisitos y del análisis de tráfico realizado. Así mismo se presenta el diseño de la red de datos y de Voz para soportar la solución.

Queda claro en el siguiente trabajo que Asterisk no es un sistema muy complejo para realizar diseños, sin embargo permite obtener diversas soluciones perfectamente ajustables a las necesidades de comunicaciones de una empresa.

Desde el punto de vista económico, una solución Asterisk pasa a ser rentable cuando se explotan sus aplicaciones avanzadas, ya que en cuanto a hardware no puede competir con la fabricación de equipamiento a gran escala.

## **CAPÍTULO II**

### **MARCO TEÓRICO CONCEPTUAL**

#### **2.1. Marco Histórico.**

A pesar de que la tecnología de Voz sobre IP ha existido desde hace bastante tiempo, es en los últimos años donde se ha convertido en una tecnología más que viable frente a la alternativa de la Telefonía tradicional. Costos más atractivos han repercutido para que el mercado se oriente en la dirección de la tecnología VoIP.

Hoy en día, los protocolos de VoIP se han desarrollado para ofrecer una gran variedad de prestaciones, escalabilidad y estandarización con respecto a lo que ofrecía hace algunos años.

Las principales razones por la que VoIP se antepone a las demás tecnologías es que tiene un menor costo efectivo debido a que no se necesita interactuar con las redes de conmutación de circuitos y conmutación de paquetes al mismo tiempo y un mismo personal técnico puede supervisar la red de voz y datos. Otra razón es que la convergencia de voz y datos sobre una misma red IP provee gran flexibilidad en términos de cambiar, quitar o agregar con facilidad nodos en la red. Adicionalmente, la tecnología VoIP promete brindar aún mayores características tales como protocolos para ruteo de llamadas avanzado, mensajes unificados, integración con otro tipo de servicio Multimedia como video y otros.

Ante esto, el mercado actual de la telefonía busca establecerse utilizando Telefonía IP y lógicamente las empresas siguen la tendencia de adquirir un buen producto con la tecnología mas actual a un buen precio. Si hablamos específicamente de los Call Centers, cuyo negocio es brindar facilidades telefónicas a sus clientes, el tema es verdaderamente crítico. Los Call Center tienen su razón de ser en la central telefónica que usan y en el capital humano que en ellos trabajan. Por lo tanto un Call Center necesariamente debe elegir una Central telefónica que le brinde todas las características de Centrales telefónicas de última generación y obviamente esta debe

ser una que soporte la tecnología de voz sobre IP. Pero, ¿Por que elegir Asterisk?. Las empresas tienen un referente en los gastos que se hacen por telecomunicaciones y es por ese motivo que Asterisk entra a tallar. Para desplegar una PBX Asterisk no se necesitará más hardware que un Servidor, algunas tarjetas de telefonía y en algunos casos equipos telefónicos (aunque esto último se puede obviar si se usa Softphones). Con el detalle anterior, Asterisk se convierte en una plataforma muy accesible desde el punto de vista económico y además brinda las funcionalidades de cualquier Central telefónica de reconocidas marcas.

## **2.2. Base teórica**

### **2.2.1. Definición de Call Center**

Un Call Center es una Oficina central usada para propósitos de hacer y recibir un gran volumen de requerimientos y/o gestiones vía telefónica. Es operado por una empresa o Compañía para realizar soporte remoto de productos o información requerida por clientes. Las llamadas de salida son realizadas con propósitos de Telemarketing (Venta a distancia), fidelización e incluso para comunicaciones sobre deudas pendientes.

Un Call Center es usualmente operado a través de un gran espacio físico de trabajo, con estaciones de trabajo que incluyen una computadora, un teléfono conectado a una Central Telefónica Privada, Headsets para habilitar manos libres y adicionalmente algunas estaciones de supervisión de la gestión. La operación de un call center puede ser manejada independientemente como parte de una red jerárquica. Adicionalmente, las redes de Voz y datos de un Call Center están ligados a través de nuevas tecnologías llamadas CTI (Computer Telephony Integration).

Otra definición complementaria es la siguiente: Un **Call Center** o Centro de atención de llamados entrantes (INBOUND) o salientes (OUTBOUND) es una herramienta de comunicación y relación con los Clientes que utiliza el TELEFONO como medio de comunicación básico gestionado por "PERSONAS HUMANAS" en conjunto a los recursos humanos, físicos y tecnológicos necesarios y disponibles, basados en metodologías de trabajo y procesos determinados y adecuados, para atender las necesidades y dar servicio a cada "CLIENTE UNICO" con el objeto de atraerlos y fidelizarlos con la organización y permitir su viabilidad.

### **2.2.2. Teoría de encolamiento**

Un call Center puede ser visto desde un punto de vista operacional como una red de encolamiento. El Call Center más simple, que consistiría en un solo tipo de clientes y



similar característica de servidores, puede ser visto como un encolamiento simple. La teoría de colas es una rama de las matemáticas en el que los modelos de sistemas de encolamiento han sido desarrollados. Estos modelos, en su turno, son usados para soportar fuerza de trabajo planificado y administración, por ejemplo ayudando a responder interrogantes comunes como: parámetros de nivel de servicio, lo que es el mínimo número agentes telefónicos que es requerido para alcanzarlo. (El nivel de servicio mínimo que prevalece en la actualidad es de 80% dentro de 20 segundos, lo que significa que como mínimo el 80% de clientes que llaman deben ser atendidos en menos de 20 segundos).

Los modelos de encolamiento también ayudan a comprender de forma cualitativa los eventos, por ejemplo, identificando circunstancias bajo las cuales economías de escala prevalecen, dando a conocer que un único y gran Call Center es más efectivo en la atención de llamadas que varios pequeños Call Centers distribuidos, o que la Venta cruzada (Cross-Selling) es muy beneficiosa. Recientemente, los modelos de encolamiento también han sido usados para planificar y operar el enrutamiento de las llamadas basados en distribución de skills, lo cual implica el análisis de sistemas con varios tipos de clientes y varios skills de agente.

### **2.2.3. Tecnología base para call-center**

Entre las tecnologías tradicionales que se ocupan en un call-center están: la infraestructura telefónica (conmutador o PBX, teléfonos, diademas o cintillos), la infraestructura de datos (computadoras, bases de datos, CRM), el distribuidor automático de llamadas entrantes (ACD), un sistema de respuesta interactiva de voz (IVR), un grabador de llamadas (que muchas veces también graba las pantallas de los agentes), y si el call-center gestiona llamadas de salida, un marcador o discador, asistido, progresivo o predictivo.

Para recolectar los valores de los parámetros que necesitan ser cuantificados por un Call Center y tener una medida del desempeño, es fundamental que la tecnología usada para su implementación cumpla con un mínimo de requerimientos que incluyen principalmente los listados a continuación:

#### **a) ACD (Distribución automática de llamadas)**

El ACD es un proceso por el cual se hacen distribuciones programadas de las llamadas que llegan a una Central telefónica. Frecuentemente, los ACD residen en PBX privadas usados para repartir tráfico a los sistemas de atención o agentes.



Debido a que se trata de un proceso programable, se pueden definir diferentes modos de distribución de llamadas que pueden incluir los criterios siguientes: Primero el agente con mayor tiempo en estado disponible, Primero el agente mas experto, primero el agente con menor cantidad de llamadas atendidas, etc. El ACD es el Proceso más importante para un sistema de atención de llamadas, ya que es el motor que permite hacer llegar las llamadas a los agentes o tratamientos automáticos finales.

#### **b) IVR (Respuesta interactiva de voz)**

Los IVR's son sistemas de respuesta interactiva incorporados ala Centrales telefónicas permitiendo, a través del teléfono, que el cliente pueda entregar y/o capturar información automatizada producto de la interacción con grabaciones de Voz. Frecuentemente, los sistemas IVR realizan consultas internas a Bases de Datos con la finalidad de almacenar o brindar datos al cliente que interactúa con ellos. Estos sistemas pueden consistir en grabaciones automáticas, menú de opciones, reconocimiento de voz, transacciones financieras, Etc.

#### **c) CTI (Integración teléfono - computadora)**

El CTI es un sistema que realiza la integración de manera coordinada del proceso de la llamada telefónica y la computadora. Esta integración permite implementar una variedad de funciones como por ejemplo: Información del ANI y DNIS (número de origen y número marcado), transferencia de llamadas con datos adjuntos, control del estado del agente, monitorización de la calidad de servicio, estados de agente para grabación de llamadas, Etc.

#### **d) Marcadores (Progresivo o predictivo - Outbound)**

Los marcadores son sistemas que pueden ser implementados en Software o hardware y cuya función es realizar llamadas de salida de forma automática desde el puesto de los agentes. Este automatismo es el que ahorra el tiempo empleado por el agente para realizar un marcado manual. Estos sistemas se basan en una fuente de datos de donde obtienen los números telefónicos a ser discados. Los marcadores usados por los Call Centers recaen principalmente en 2 categorías:

Los marcadores progresivos se basan en que necesariamente deben recibir el evento de finalización de una llamada para iniciar el discado de un nuevo número telefónico el cual será pasado al agente desde el momento en el que se inicia la llamada.

Los marcadores predictivos se basan en algoritmos de análisis de gestión y reconocimiento de señales para soportar la generación de una llamada en una probabilidad.

#### **e) CRM (Customer relationship management)**

La administración de la relación con los clientes, CRM, es parte de una estrategia de negocio centrada en el cliente. Una parte fundamental de su idea es, precisamente, la de recopilar la mayor cantidad de información posible sobre los clientes, para poder dar valor a la oferta. La empresa debe trabajar para conocer las necesidades de los mismos y así poder adelantar una oferta y mejorar la calidad en la atención. Cuando hablamos de mejorar la oferta nos referimos a poder brindarles soluciones a los clientes que se adecuen perfectamente a sus necesidades, y no como rezan muchos opositores a estas disciplinas generarles nuevas necesidades. Por lo tanto, el nombre CRM hace referencia a una estrategia de negocio basada principalmente en la satisfacción de los clientes, pero también a los sistemas informáticos que dan soporte a esta estrategia.

#### **f) Buzón de voz**

El buzón de voz es un sistema requerido no solo en Call Centers sino también en todo ámbito donde se use la telefonía. Como tal es ampliamente conocido que este sistema permite dejar un mensaje pregrabado de voz en caso de no obtener respuesta en el punto remoto con el que se deseaba establecer una llamada telefónica. Generalmente los buzones de voz se activan después de un tiempo o cantidad de timbrados establecidos sin obtener respuesta, luego de lo cual se recibe un mensaje de información con indicaciones de cómo dejar el mensaje de voz. Este tipo de sistemas es fundamental en los Call Centers con operaciones InBound ya que en caso de no poder atender una llamada, las llamadas de los clientes son enrutadas hacia el buzón de voz que posteriormente es revisado para poder realizar gestiones OutBound y contactar a los clientes que no pudieron acceder

#### **g) Grabación de voz**

Los sistemas de grabación de voz han venido tomando gran importancia debido a las transacciones telefónicas. Muchos Call Centers realizan Telemarketing y como tal, es necesario tener un documento que acredite la venta realizada, es justo aquí donde la grabación de audio entra a tallar. Existen diferentes clases de normativas dependiendo

del País en donde se realice Telemarketing, sin embargo en todos estos países es necesario tener una prueba reconocible de la venta, la cual viene a ser la grabación de voz. Los sistemas de grabación pueden ser del tipo hardware o software, sin embargo para almacenar la grabación física como tal es necesario disponer de espacio físico. Además la forma de grabación dependerá de la tecnología usada para la telefonía (Analógica, Digital, IP)

#### **h) CDR (registro de detalle de llamadas)**

Los CDR o registros de llamadas son los archivos con la información de los eventos por los que pasaron las llamadas telefónicas, es decir, en estos archivos reside información tal como: Fecha y hora de inicio de llamada, duración, ANI, DNIS, Extensión que atendió la llamada, etc. Estos archivos son la base para extracción de reportes de atención, que finalmente son los que permiten llevar el control de la gestión.

#### **i) Estadísticas de desempeño**

La Parte fundamental para el desempeño de un Call Center son los reportes de gestión. Estos pueden abarcar desde reportes de logueo de agentes hasta reportes de tiempo promedio de abandono de llamadas. Justamente los Call Center basan su negocio en la información de estos reportes, los cuales muestran indicadores de desempeño de todos los elementos del Call Center lo cual permite realizar los ajustes que sean necesarios.

### **2.2.4. Elementos adicionales para el Funcionamiento de un Call Center**

#### **a) Teléfonos**

Los Teléfonos vienen a ser la interfaz que permite establecer el proceso de llamada. Este puede tener 2 formas. La Primera es como equipo Telefónico es decir un Hardware diseñado para poder establecer y procesar llamadas usando la PBX como medio. La segunda forma es de tipo Softphone, es decir una aplicación software que se ejecuta sobre un sistema operativo de PC y que emula todas las características de un Teléfono físico. En la figura 2.1 se muestra un Teléfono IP.



**Fig. 2.1 Teléfono IP Syslink**

#### **b) PCs**

Las PCs son los elementos que permitirán la integración Teléfono Computadora (CTI). Esta integración hace trabajar a aplicaciones encargadas de mostrar y recoger datos con los eventos de la llamada (Respuesta, corte, BackOffice, Etc.).

#### **c) Bases de Datos**

Las bases de datos son elementos adicionales pero necesarios en toda gestión de Call Center. Estas permiten el almacenamiento de la información recogida por los agentes en el proceso de conversación con clientes finales. Además también sirve como plataforma de registros de datos que incluyen números telefónicos que serán discados.

#### **d) Headsets**

Elemento necesario para el ahorro de tiempo por el agente ya que al usar las 2 manos, aumenta la velocidad de gestión en procesamiento de datos.

#### **e) Red de datos**

Sin lugar a duda es uno de los elementos que mas en cuenta se debe tener al momento de optar por una solución de Telefonía IP. Esto incluye el diseño de una red de datos separa de la red de voz y previniendo todos los posibles factores que puedan causar inconvenientes tales como Jitter, echo o delay que son de los mas comunes

sobre las redes que no fueron pensadas para telefonía IP. Sobre este punto se analizará con mayor detalle mas adelante.

### **2.2.5. Fundamentos de Voz sobre IP**

Años atrás se descubrió que mandar una señal a un destino remoto podía hacerse también de manera digital: antes de enviar la señal se debía digitalizar con un ADC (analog to digital converter), transmitirla y en el extremo de destino transformarla de nuevo a formato análogo con un DAC (digital to analog converter).

VoIP funciona de esa manera, digitalizando la voz en paquetes de datos, enviándola a través de la red y reconvirtiéndola a voz en el destino. Básicamente el proceso comienza con la señal análoga del teléfono que es digitalizada en señales PCM (pulse code modulación) por medio del codificador/decodificador de voz (codec). Las muestras PCM son pasadas al algoritmo de compresión, el cual comprime la voz y la fracciona en paquetes que pueden ser transmitidos para este caso a través de una red privada WAN. En el otro extremo de la nube se realizan exactamente las mismas funciones en un orden inverso.

Dependiendo de la forma en la que la red este configurada, el enrutador o el Gateway puede realizar la labor de codificación, decodificación y/o compresión. Por ejemplo, si el sistema usado es un sistema análogo de voz, entonces el enrutador o el Gateway realizará todas las funciones mencionadas anteriormente.

Si, por otro lado, el dispositivo utilizado es una PBX digital, es entonces éste el que realiza la función de codificación y decodificación, y el enrutador solo se dedica a procesar las muestras PCM que le ha enviado el PBX.

Es importante tener en cuenta también que todas las redes deben tener de alguna forma las características de direccionamiento, enrutamiento y señalización. El direccionamiento es requerido para identificar el origen y destino de las llamadas, también es usado para asociar clases de servicio a cada una de las llamadas dependiendo de la prioridad. El enrutamiento por su parte encuentra el mejor camino a seguir por el paquete desde la fuente hasta el destino y transporta la información a través de la red de la manera más eficiente, la cual ha sido determinada por el diseñador. La señalización alerta las estaciones terminales y a los elementos de la red su estado y la responsabilidad inmediata que tienen al establecer una conexión.

Para comprender la tecnología VoIP, es importante tener en cuenta sus componentes básicos:

- Señalización.
- Servicios de Base de Datos.
- Conexión y Desconexión de llamadas.
- Codecs de Operación.

#### **a) Señalización**

La señalización es la forma en la que los dispositivos se comunican dentro de la red, activando y coordinando la variedad de componentes necesarios para completar una llamada. En una red VoIP, la señalización VoIP es completada mediante el intercambio de datagramas IP entre los componentes IP.

#### **b) Servicios de Base de Datos**

Los servicios de base de datos son una forma de localizar un Terminal y trasladar la dirección que 2 redes usan. La PSTN, usa números de Teléfono para identificar los terminales. Una red VoIP usa direcciones IP y números de puerto para identificar un Terminal.

#### **c) Conexión y Desconexión de llamadas**

La conexión de una llamada es hecha por 2 terminales abriendo una sesión de comunicación entre una y otra. En la PSTN, el switch privado o público conecta canales de señal digital a través de la red para completar llamadas. En las implementaciones VoIP, esta conexión es una trama multimedia (audio, video o ambos) transportados en tiempo real. Esta conexión es canal limpio.

#### **d) Codecs**

La Comunicación de voz es análoga, mientras los datos de Red son digitales como resultado, la red necesita una forma para ser capaz de convertir la voz en un formato que se pueda transportar. Entonces se hace necesaria la paquetización de la voz. El proceso de convertir las señales análogas a información digital es realizado con un Codificador-Decodificador también conocido como Vocoder. Hay muchas formas en las que una señal de voz análoga puede ser transformada todas ellas son gobernadas por una serie de estándares.



Algunos de los CODEC's mas usados se muestran en la figura 2.2

Packet size	10 ms	20 ms	30 ms	40 ms	50 ms	60 ms
G.711	102	83	77	74	72	71
G.729	46	27	21	18	16	15
G.723	NA	NA	19	NA	NA	13
G.726	67	50	43	40	39	38

Fig 2.2 Consumo de ancho de banda para diversos Codecs

El equivalente del concepto de conmutador (central local, central de grupo, etc.) en VoIP es el *Media Gateway Controller (MGC)*. Éste es un conjunto de productos, protocolos y aplicaciones capaz de permitir que cualquier dispositivo acceda a los servicios de Internet y servicios de Telecomunicaciones sobre las redes IP. Este elemento es la pieza central en la red de telefonía IP, ya que es capaz de manejar inteligentemente las llamadas en la plataforma de servicio de los ISP's (Internet Service Provider). Por otro lado, sirven como plataformas de integración para aplicaciones e intercambio de servicios y son capaces de transportar tráfico de voz, datos y video de una manera más eficiente que los equipos existentes.

El Media Gateway Controller es eficiente gracias a su interacción con el Media Gateway y el Signaling Gateway. Las funciones principales son: control de llamada, protocolos de establecimiento de llamadas como H.323 y SIP, protocolos de control de media por ejemplo MGCP y H.248, control sobre la calidad y clase de servicio, conocimiento del enrutamiento, plan de marcado local, detalle de las llamadas para facturación, control de manejo del ancho de banda, crear un puente entre la señalización SS7 y VoIP, asignación del canal DS0, entre muchas otras más.

Algunos dispositivos adicionales facilitan la interconexión de la Red IP con la de la telefonía tradicional, tal es el caso de los módulos FXO (Foreign Exchange Office) y el FXS (Foreign Exchange Station). El primero conecta VoIP con Centrales de Conmutación de la PSTN mientras el segundo convierte de VoIP a canal analógico a nivel línea de abonado.

Así mismo puede observarse que la red de VoIP tiene las mismas características que la red IP. Las redes pueden ser inalámbricas, de área local o de amplia cobertura, y

contienen elementos como routers, servidores, en conjunto con los elementos particulares de VoIP como el *Media Gateway Controller*, los IP-PBX, las interfaces de salida hacia otras redes y usuarios de servicios especiales de VoIP como el Chat.

### 2.2.6. Protocolos VoIP

La red de voz sobre IP es muy grande y eficiente. Esta es una red de paquetes conmutados. La voz es empaquetada y enviada por diferentes rutas, siempre buscando la ruta más corta y/o menos congestionada. Es ahí cuando los protocolos entran en acción.

Los protocolos que se utilizan en las redes de voz sobre IP son: SGCP, MGCP, H.323 y SIP, entre otros; todos definidos por instituciones y organismos reguladores con normativas de control como: la ITU-T, la IETF, el ETSI o el EIA-TIA. Estos protocolos tienen interfaces abiertas y estándares definidos, y cuentan con una buena infraestructura de paquetes. En la figura 2.3 se esquematiza las capas y protocolos.

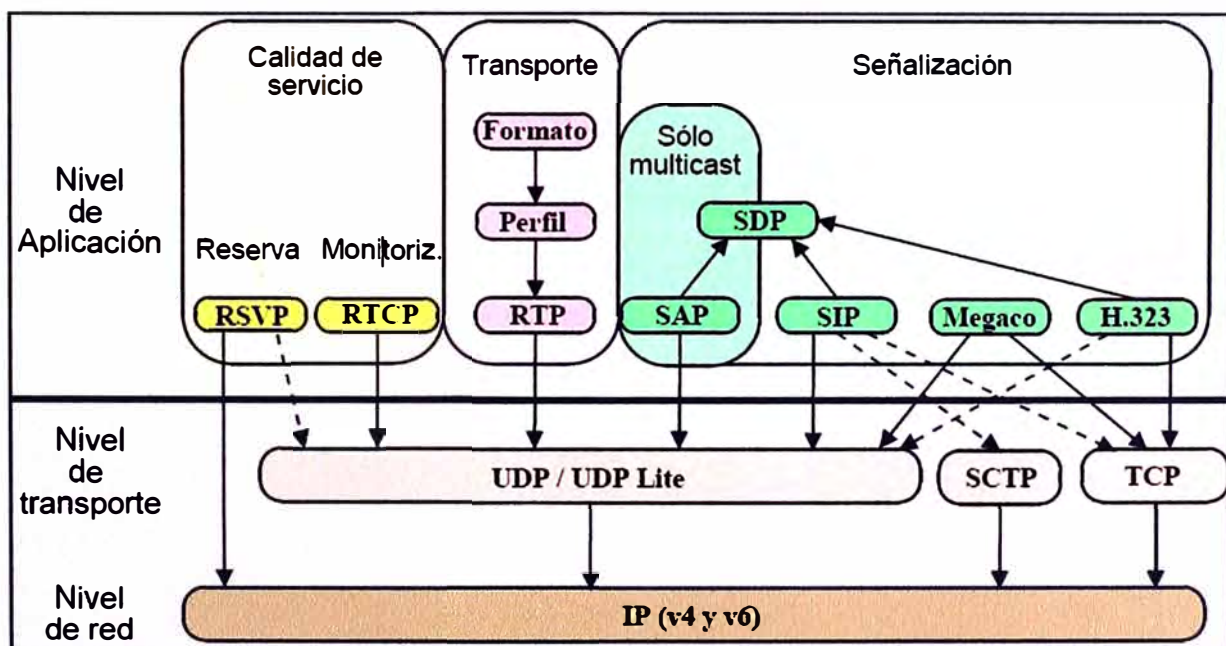


Fig. 2.3 Esquematización de la Telefonía IP

### 2.2.7. Protocolo de Transporte en tiempo real RTP

RTP (Real-time Transport Protocol) o protocolo de transporte en tiempo real, es un protocolo que como su nombre lo indica, está orientado a la transmisión de información en tiempo real, como la voz o el video. Este es un protocolo de las capas superiores de usuario que funciona sobre UDP (User Datagram Protocol) haciendo uso de los servicios de checksum y multiplexación, para proporcionarle a los



programas que generan este tipo de datos, un manejo de transmisiones en tiempo real a través de difusiones unicast o multicast. En UDP se cambia confiabilidad por velocidad, lo cual es básico para manejo de transmisiones en tiempo real como la VoIP.

Aunque RTP no es suficientemente confiable por si solo, este proporciona relaciones con protocolos y aplicaciones de capas inferiores y recursos proporcionados por los switches y enrutadores para garantizar confiabilidad. Los paquetes RTP no contienen campo de longitud, ya que al funcionar sobre UDP, este protocolo es quien encapsula la voz comprimida en datagramas.

La herramienta de la que se vale RTP para lograr transmisiones en tiempo real es el RTCP (Real-Time Control Protocol ) que proporciona un feedback acerca de la calidad de distribución y la congestión. Con esto, la empresa que ofrece el servicio puede monitorear la calidad y puede diagnosticar los problemas que pueda presentar la red, además de esto, RTCP sincroniza el audio y el video, conoce el número de usuarios presentes en una conferencia y con esto calcula la tasa a la cual deben ser enviados los paquetes.

### **2.2.8. El Estándar H.323**

El estándar H.323 posibilita en intercambio de servicios multimedia (audio, video y datos en tiempo real) sobre redes de paquetes, lo que incluye las redes IP y forma parte de un conjunto de recomendaciones de la ITU-T llamadas H.32x que proveen comunicaciones multimedia sobre todo tipo de redes.

#### **a) Componentes de H.323**

H.323 especifica los siguientes componentes que posibilitan la comunicación multimedia:

- Terminales
- Gateways
- Gatekeepers
- MCU (Unidades de Control Multipunto)

**Terminales:** Se utilizan para comunicaciones multimedia bidireccionales de tiempo real. Sirve para comunicaciones de audio y opcionalmente puede usarse para video y datos. Puede tratarse tanto de una PC como de un dispositivo corriendo una aplicación multimedia utilizando H.323. Estos terminales H.323 pueden también ser usados en conferencias multipunto.

**Gateways:** Se utilizan para interconectar redes disímiles. Conecta en definitiva, redes H.323 con redes que no son H.323. Puede conectar, por ejemplo, terminales de una red H.323 y redes PSTN. Para lograr esta interconexión, se traducen protocolos de una red hacia la otra usando el gateway. Dentro de una misma red H.323 no es necesario utilizar un gateway.

**Gatekeepers:** Se lo puede considerar como un portero en una red H.323. Aquí se efectúan las tareas de direccionamiento, autenticación tanto de terminales como gateways, gestión de ancho de banda y contabilidad. Pueden también proveer servicios de encaminamiento de llamadas.

**MCU o Unidades de Control Multipunto:** Éstas proveen soporte para realizar conferencias. A tal efecto los terminales que deseen participar deberán conectarse al MCU. Así, el MCU gestionará los recursos negociando entre terminales. Gatekeepers, Gateways y MCU pueden ser componentes que físicamente estén integrados en un solo equipo, aunque de manera lógica implementen funciones totalmente separadas.

## b) Protocolos especificados por H.323

Los protocolos especificados por H.323 son:

- Codecs de audio
- Codecs de video
- Registro, Admisión y Estado H.225 (RAS)
- Señalización de llamadas H.225
- Señalización de control H.245
- Protocolo de transferencia en tiempo real (RTP)
- Protocolo de control en tiempo real (RTCP)

H.323 es independiente de la red de paquetes y de los protocolos de transporte sobre los cuales corre y no establece especificaciones sobre estos. La figura 2.4 esquematiza la pila de protocolos H.323.

**Codec de audio:** permite la codificación de las señales de audio que ingresan al terminal H.323 para ser transmitidas y decodifica el audio recibido para que sea reproducido por el parlante. Todos los terminales H.323 deben soportar al menos un codec de audio como se especifica en las recomendaciones ITU-T G.711 (i.e., codificación a 64kbps). Puede opcionalmente soportar otros tales como G.722, G.723, G.728 y G.729.

**Codec de video:** permite codificar la señal de video que ingresa al terminal transmisor y decodificar los datos de video codificado recibidos para enviarlos al monitor. El

soporte de este codec es opcional dado que también lo es el soporte de video en H.323.

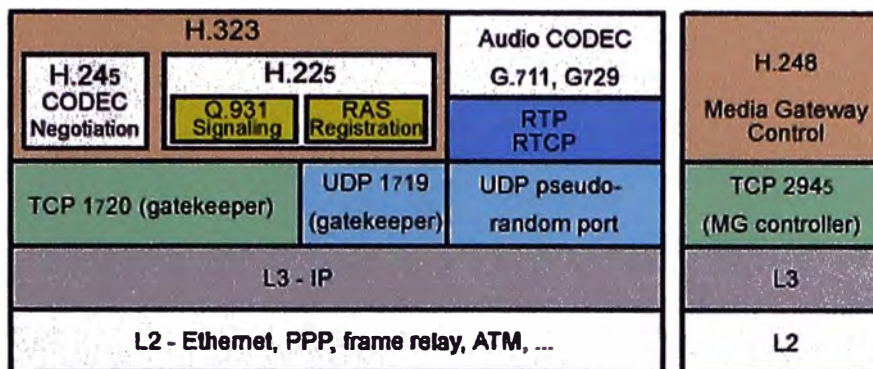


Fig. 2.4 Pila de protocolos H.323

**RAS, registro, admisión y estado H.225:** se trata de un protocolo usado entre los terminales y gateways (puntos terminales) y el gatekeeper. RAS es usado para las funciones de registro, control de admisión y gestión de ancho de banda entre puntos terminales y gatekeeper, entre otras funciones. Para los mensajes RAS se utiliza lo que se denomina un canal RAS, que es un canal de señalización que está abierto entre una Terminal y el gatekeeper antes del establecimiento de cualquier otro canal.

**Señalización de llamadas H.225:** esta señalización se utiliza para el establecimiento de las conexiones entre dos puntos terminales H.323, mediante el intercambio de mensajes sobre el canal de señalización. Éste canal existe entre dos puntos terminales H.323 o entre un terminal y un gatekeeper.

**Señalización de Control H.245:** esta señalización se usa para manejar la operación de los puntos extremos H.323 intercambiando mensajes de control que transportan la siguiente información: Intercambio de capacidades, Apertura y cerrado de canales lógicos usados para transportar flujos multimedia, Mensajes de control de flujo, Comandos e indicaciones generales

### 2.2.9. El estándar SIP

SIP tuvo como objetivo inicial de la IETF que se pudieran invitar usuarios en sesiones Mbone (parte de Internet que soporta multicast). Se desarrolló así el protocolo y en la actualidad se usa como protocolo de inicio de sesiones de todo tipo, incluyendo multicast. SIPv1 estaba basado en texto y utilizaba SDP (Protocolo de Descripción de Sesión) para describir sesiones, usaba UDP como transporte. Tenía la particularidad de proveer cierta movilidad a los usuarios a partir del uso de servidores de direcciones. Intervenia solo para establecer las sesiones y finalizaba su intervención una vez que la

Sesión era establecida. Protocolo de Invitación de Conferencia Simple: SCIP En 1996 IETF emitió un draft que describía un mecanismo para invitar usuarios a sesiones punto a punto y sesiones multicast, llamado Protocolo de Conferencia Simple (SCIP). Este se basaba en HTTP y por ello utilizaba TCP como protocolo de transporte, y también era basado en texto. SCIP utilizaba direcciones de e-mail para identificar a los usuarios, con el objetivo de proveer un identificador universal para comunicaciones asincrónicas y sincrónicas. La señalización de CSIP se mantenía después de que la sesión se establecía.

**Protocolo de Iniciación de Sesión: SIPv2:** Producto de fusionar SIP y SCIP se produjo el avance a SIPv2. El significado de la sigla se cambió a Protocolo de Iniciación de Sesión. SIP está en continuo crecimiento y adaptación. Su funcionamiento tiene ciertos elementos de http y smtp lo que lo hace muy versátil para funcionar. Sus ventajas principales residen en las características de simplicidad, escalabilidad, funcionalidad distribuida y versatilidad con los protocolos Internet.

**Movilidad del usuario:** SIP no puede enviar una descripción de una sesión a un potencial participante hasta que no haya sido localizado. Frecuentemente un usuario debe ser alcanzado en diferentes ubicaciones. Por ejemplo, en los laboratorios de la facultad un alumno podría utilizar diferentes máquinas, y debido a ello se ubicaría en diferentes direcciones IP.

Por ello SIP soporta de forma transparente el mapeo de nombres y la redirección de servicios, permitiendo la implementación de servicios de redes inteligentes de telefonía.

Estas facilidades también permiten movilidad personal, es decir, la habilidad de los usuarios finales de originar y recibir llamadas y de acceder a servicios de telecomunicaciones en cualquier lugar. La movilidad personal está basada en el uso de una identificación personal única.

La movilidad personal complementa a la movilidad del terminal, por ejemplo, roaming. Los usuarios en un ámbito SIP son identificados por un URL SIP (Localizador Uniforme de Recursos). El formato de un URL SIP es similar a una dirección de e-mail. Generalmente consiste en un nombre de usuario y un nombre de dominio. Los URL's SIP son utilizados en los mensajes para indicar: el que llama (From), el destino (Request-URI) y el que es llamado (To) en una solicitud SIP, y para especificar la dirección de redirección (Contact).

Un URL SIP sigue la guía del RFC 2396. A continuación se muestran algunos ejemplos:

Sip: usuario@host

Sip: usuario:contraseña@host

Sip: usuario@direccionIP:puerto

Sip: usuario@host;parametros-url

Donde:

usuario: El nombre del usuario.

contraseña: Clave de seguridad del usuario.

host: El host donde se encuentra el usuario. Puede ser un nombre o una dirección IP.

puerto: Numero de puerto para enviar una solicitud.

parámetros URL: Los URL's SIP pueden definir parámetros específicos de la solicitud.

**Entidades SIP.** UAC: Agente del Usuario Cliente, es el que llama: Un UAC es un cliente a nivel aplicación que inicia una solicitud SIP.

UAS: Agente del usuario Servidor. Un UAS es un servidor a nivel aplicación que contacta al usuario cuando una solicitud de SIP es recibida y retorna una respuesta en nombre del usuario. La respuesta a la solicitud es aceptada, rechazada o redireccionada.

UA: El agente del usuario es una aplicación que contiene al UAC y al UAS. Cuando un usuario desea hablar con otro, ejecuta un programa que contiene un UA. El usuario interactúa a través de la interface. Cuando el primer usuario oprime el botón de llamar Al segundo, el UA dispara un mensaje SIP apropiado para establecer la llamada. Todas las interacciones entre los usuarios y el protocolo SIP son mediante los UA's. Sin embargo algunos sistemas que utilizan SIP no están directamente conectados con el usuario. Por ejemplo se puede redireccionar todas las llamadas recibidas a la hora de la siesta al contestador automático. El contestador automáticamente establecerá la sesión para grabar los mensajes.

Cuando la sesión describe una sesión de audio/video, esta no puede ser establecida sin un UA, una herramienta de audio y una herramienta de video. Estas pueden ir o no incluidas en una misma aplicación, así como otros tipos de herramientas de sesión, pero la separación de estos elementos hace que SIP tenga más flexibilidad, y permita el establecimiento de cualquier tipo de sesión.

**Servidores Redireccionadores.** Los servidores Redireccionadores ayudan a localizar UA's SIP proveyendo ubicaciones alternativas donde el usuario puede ser alcanzado. Por regla general el servidor redireccionador no siempre retorna la dirección de un UA, sino que retorna la dirección de otro servidor que posee mayor conocimiento acerca de

la ubicación del usuario. Los servidores redireccionadores pueden también ser utilizados para implementar grupos de direcciones. El servidor redireccionador puede retomar diferentes direcciones dependiendo, por ejemplo, de la hora del día.

**Servidores Proxy** Los servidores Proxy aceptan sesiones hechas por un UA y solicitan la información de direccionamiento del UA de destino al Servidor de Registro. Luego, redireccionan la invitación directamente al UA destino, si se localiza en el mismo dominio, o se dirigirá a otro Servidor Proxy en caso que el UA destino resida en otro dominio.

**Proxy bifurcador** Cuando un servidor Proxy emite un INVITE de una misma sesión a diferentes puntos, se dice que está bifurcando la invitación. Los servidores Proxy bifurcadores pueden ejecutar búsquedas paralelas o secuenciales dependiendo de su configuración. Una búsqueda paralela consiste en el intento de las posibles ubicaciones al mismo tiempo y una búsqueda secuencial consiste en la búsqueda de cada ubicación individualmente.

**Grupo de direcciones** Los servidores Proxy también pueden crear grupos de direcciones e intentar contactar a todas las personas del grupo hasta encontrar una persona que esté disponible. Durante el establecimiento de la sesión, es común que los dos tipos de servidores (Proxy y redireccionadores) estén juntos. El término servidor SIP se refiere a ambos tipos de servidores.

**Registradores** Los registradores son los servidores SIP que aceptan invitaciones. Un servidor registrador está usualmente ubicado junto a un servidor redireccionador o un servidor Proxy.

**Servidores de localización** Los servidores de localización no son entidades SIP, pero son una parte importante de cualquier arquitectura que use SIP. Un servidor de localización guarda y retorna posibles localizaciones de usuarios. Este puede utilizar información de otras bases de datos. La mayoría de los registradores "suben actualizaciones de localizaciones a un servidor de localización.

**Transacciones Cliente/Servidor SIP** se basa en el protocolo HTTP, es un protocolo de solicitud/respuesta. Un cliente es una entidad SIP que genera solicitudes. Un servidor es una entidad SIP que recibe solicitudes y devuelve respuestas.

SIP se basa en los mismos procedimientos. Siguiendo la misma terminología cuando dos UA's intercambian mensajes SIP, el UA que envía solicitudes es el UAC y el UA que retorna respuestas es el UAS. Una solicitud SIP junto con la respuesta que genera, es referida como una transacción SIP. En la figura 2.5 se esquematiza los pasos de una transacción SIP.



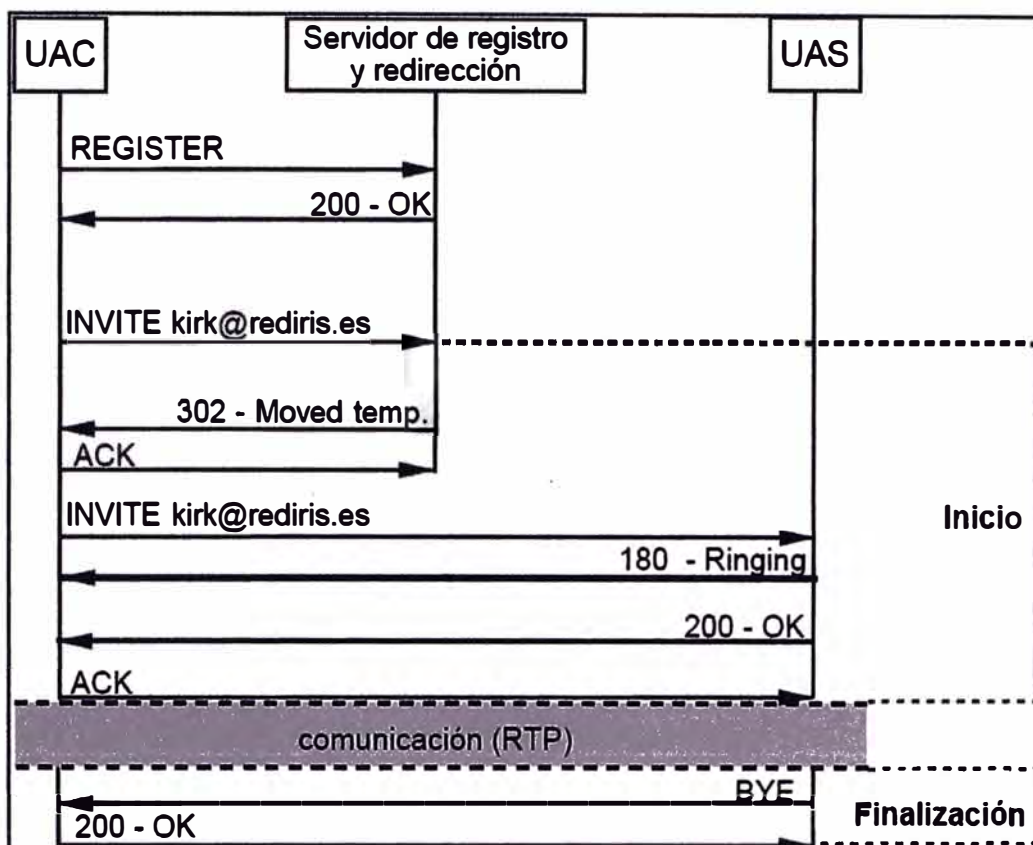


Fig. 2.5 Establecimiento de una transacción SIP (Asterisk PBX Guía de Configuración –Flavio M.)

### 2.2.10. Calidad de Servicio

La calidad del Servicio o QoS (Quality of Service) como es más popularmente llamado, se refiere al desafío de la entrega de una corriente de datos sensible al tiempo a través de una red que fue diseñada para entregar datos haciendo el mejor esfuerzo. Aunque no haya ninguna regla difícil, es generalmente aceptado que si se puede entregar el sonido producido por el altavoz al oído del oyente dentro de los 300 milisegundos, es posible transportar un flujo de conversación normal. Cuando la tardanza excede los 500 milisegundos, se hace difícil evitar la interrupción entre los participantes de la llamada. Más allá de un segundo, una conversación normal se hace muy torpe. Además de la necesidad de llegar a tiempo, es también esencial asegurar que la información transmitida llegue intacta. Demasiados paquetes perdidos impedirán reproducir de forma completa el audio, y huecos en los datos serán oídos como estática o, en casos severos, palabras u oraciones enteras perdidas.

Si se va a enviar datos a través de una red de IP, estos serán transportados usando uno de los tres protocolos de transporte descritos a continuación.

**Protocolo de Control de Transmisión** El Protocolo de Control de Transmisión (TCP) casi nunca es usado para VoIP, ya que mientras posee mecanismos para asegurar la entrega, no tiene intrínsecamente ninguna prisa en lograr esto. A menos que se tenga una interconexión de sumamente baja latencia entre los puntos extremos, TCP va a tender a causar más problemas de los que va a solucionar. El objetivo de TCP es garantizar la entrega de paquetes. En el mundo de VoIP, llevar rápidamente los paquetes al punto extremo es supremo, y 20 años de telefonía celular nos han entrenado para tolerar algunos paquetes perdidos.

El alto procesamiento de saturación del TCP, la administración por estados y el reconocimiento de llegada de los paquetes, son buenos a la hora de transmitir cantidades grandes de datos, pero simplemente no es bastante eficiente para comunicaciones de tiempo real.

**Protocolo de Datagrama de Usuario** A Diferencia de TCP, el Protocolo de Datagrama de Usuario (UDP) no ofrece ninguna clase de garantía de entrega. Los paquetes son colocados en el vínculo tan rápidamente como es posible y liberados en el mundo para encontrar el camino a sus destinos finales, sin comunicar atrás si realmente llegan o no. Ya que UDP en sí mismo no ofrece ninguna clase de garantías de que los datos llegarán, consigue eficacia gastando muy poco esfuerzo en lo que está transportando.

**Protocolo de Transmisión de Control de Flujo** Aprobado por el IETF como un estándar propuesto en RFC 2960, SCTP es un protocolo de transporte relativamente nuevo. Fue diseñado para trabajar sobre los defectos tanto de TCP como de UDP, especialmente los relacionados con los tipos de servicios que se usan para entregar datos en redes de telefonía de conmutación de circuitos. Algunos objetivos de SCTP eran:

- Mejores técnicas para evitar la congestión
- Estricta entrega de datos en secuencia
- Baja latencia para transmisiones de tiempo real mejoradas

Lógicamente, las necesidades de la telefonía IP han terminado escogiendo al protocolo UDP como el indicado para transmitir paquetes con la mayor velocidad posible.



### 2.2.11. Fundamentos de Asterisk

Asterisk es un PBX implementada en software que permite que terminales clientes se conecten a él. Una vez conectados, los usuarios pueden transmitir voz y vídeo en tiempo real utilizando cualquiera de los protocolos y códecs soportados por Asterisk.

Asterisk es una plataforma abierta y sus características son fáciles de agregar y muchas nuevas características se agregan continuamente. Cualquier característica agregada a Asterisk por un usuario puede estar disponible para otro en su sistema.

El software libre vive gracias a las comunidades de usuarios que lo mantienen. Asterisk tiene una comunidad que crece exponencialmente. El 90% de este crecimiento viene dado por los intereses de empresas interesadas como: Empresas de telefonía (proveedores IP), Empresas de tele-gestión, Desarrolladoras de software de Call Centers, Desarrolladoras de software de comunicaciones, Empresas distribuidoras de productos de VoIP y en general, cualquier empresa que necesite ahorrar y actualizar sus telecomunicaciones. En la figura 2.11 se ilustra un esquema conceptual para la definición de Asterisk.

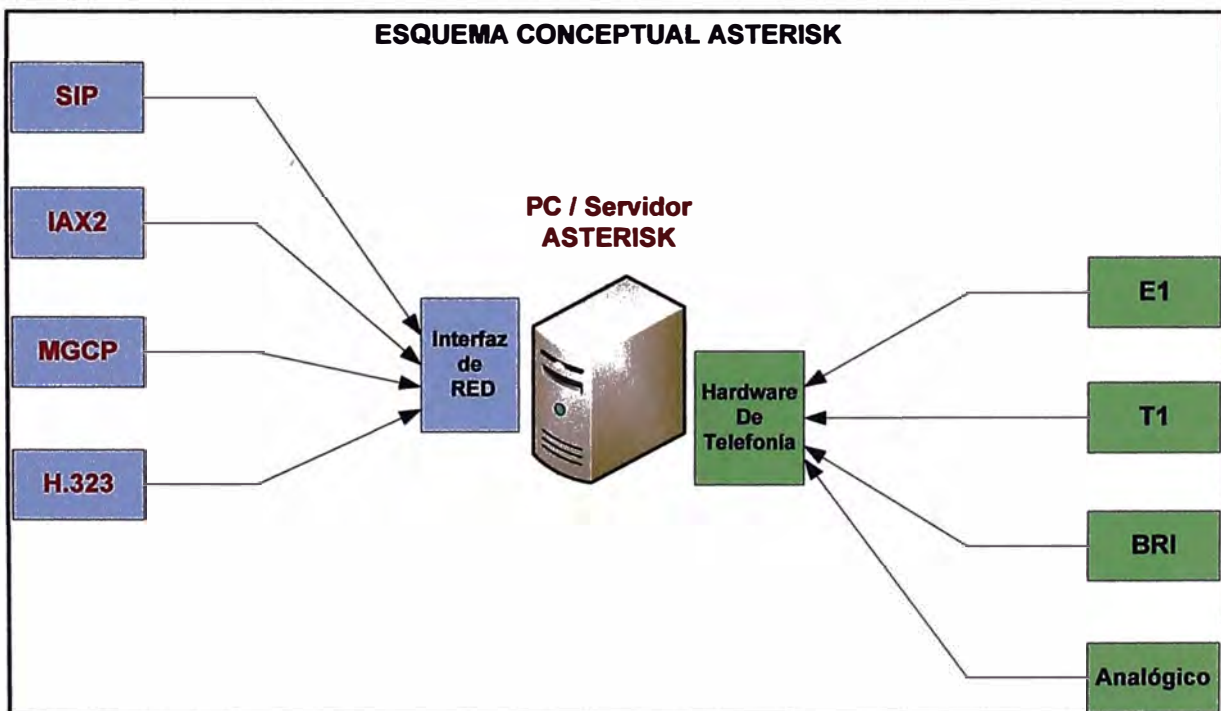


Fig. 2.6 Esquema conceptual de Asterisk

#### a) Estructura de la Plataforma Asterisk

El asterisco (\*) es un símbolo utilizado en muchas aplicaciones y es el nombre perfecto para esta PBX por varias razones, una de las cuales es la enorme cantidad de interfaces con las que Asterisk puede conectarse. Entre éstas se incluyen:

- Interfaces analógicas, como la de la línea de teléfono
- Circuitos digitales, como líneas T1 o E1
- Protocolos VoIP, como SIP e IAX

Asterisk no necesita hardware especial (ni siquiera una placa de sonido). Las placas de canales que conectan Asterisk con teléfonos análogos o líneas de teléfono están disponibles pero no son imprescindibles. Esto es así, dado que se puede conectar a Asterisk usando soft-phones (teléfonos basados en software) que están disponibles para Windows, Linux y otros sistemas operativos sin necesidad de ningún hardware especial. Se puede conectar incluso con cualquier teléfono IP que soporte SIP o IAX2. Por otro lado, en caso de no conectar la central a una línea telefónica de salida, se pueden cursar las llamadas por Internet a un proveedor de servicio telefónico.

## **b) Directorios utilizados por Asterisk**

Asterisk usa muchos de los directorios del sistema Linux para manejar varios aspectos de sistema, como grabaciones de voicemail, archivos de configuración, etc. A continuación veremos los directorios necesarios, los cuales se crean durante la instalación y se configuran en el archivo asterisk.conf.

El directorio `/etc/asterisk` contiene los archivos de configuración de Asterisk. Un archivo sin embargo, `zaptel.conf`, está en el directorio `/etc/`. El hardware Zaptel fue originalmente desarrollado por Jim Dixon del Grupo de Telefonía Zapata como una forma de vincular de manera razonable y asequible los equipos de computación telefónicos al mundo.

El directorio `/usr/lib/asterisk/modules` contiene todos los módulos cargables de Asterisk. En este mismo directorio también hay varias aplicaciones, codecs, formatos y canales utilizados por Asterisk. Por defecto, Asterisk carga todos estos módulos al iniciarse.

El directorio `/var/lib/asterisk` contiene el archivo `astdb` y varios subdirectorios. El archivo `astdb` contiene la información de la base de datos local de Asterisk, que es algo así como el Registro de Microsoft Windows.

El directorio `/var/spool` de Asterisk contiene varios subdirectorios, incluyendo `outgoing/`, `qcall/`, `tmp/` y `voicemail/`. Asterisk monitorea los directorios `qcall` y `outgoing` en busca de archivos de texto que contienen pedidos de información. Estos archivos permiten al usuario generar una simple llamada copiando o moviendo el archivo de estructura

correcto dentro del directorio outgoing/. El viejo método qcall generaba llamadas utilizando una simple línea de texto con el archivo de llamada. Los archivos de llamada para usarlos con el directorio qcall tomaban la siguiente forma:

Dialstring CAller-ID Extensión Maxsecs [Identifier] [Required-response]

Esto limitaba un poco lo que se podía hacer con el archivo de llamada, y que tipo de información se podía pasar a Asterisk. De este modo, un nuevo método se desarrolló en Asterisk usando el directorio outgoing. Los archivos de llamada que estén en este directorio pueden contener mucha mas información, como ser un contexto, extensiones, y prioridades donde la llamada debe comenzar, o simplemente una aplicación y sus argumentos.

Todos los voicemail y saludos de usuarios están en el directorio voicemail/.Extensiones configuradas en voicemail.conf que tiene que haber sido logeadas por lo menos una vez se crean como subdirectorio de voicemail/.

El directorio /var/run contiene la información de procesos ID (pid) para todos los procesos activos en el sistema, incluyendo Asterisk.

En el directorio /var/log/asterisk se guarda la información de logs. Se puede controlar el tipo de información a ser guardada por los archivos editando logger.conf situada en el directorio /etc./asterisk. La estructura de Asterisk se esquematiza en la figura 2.7

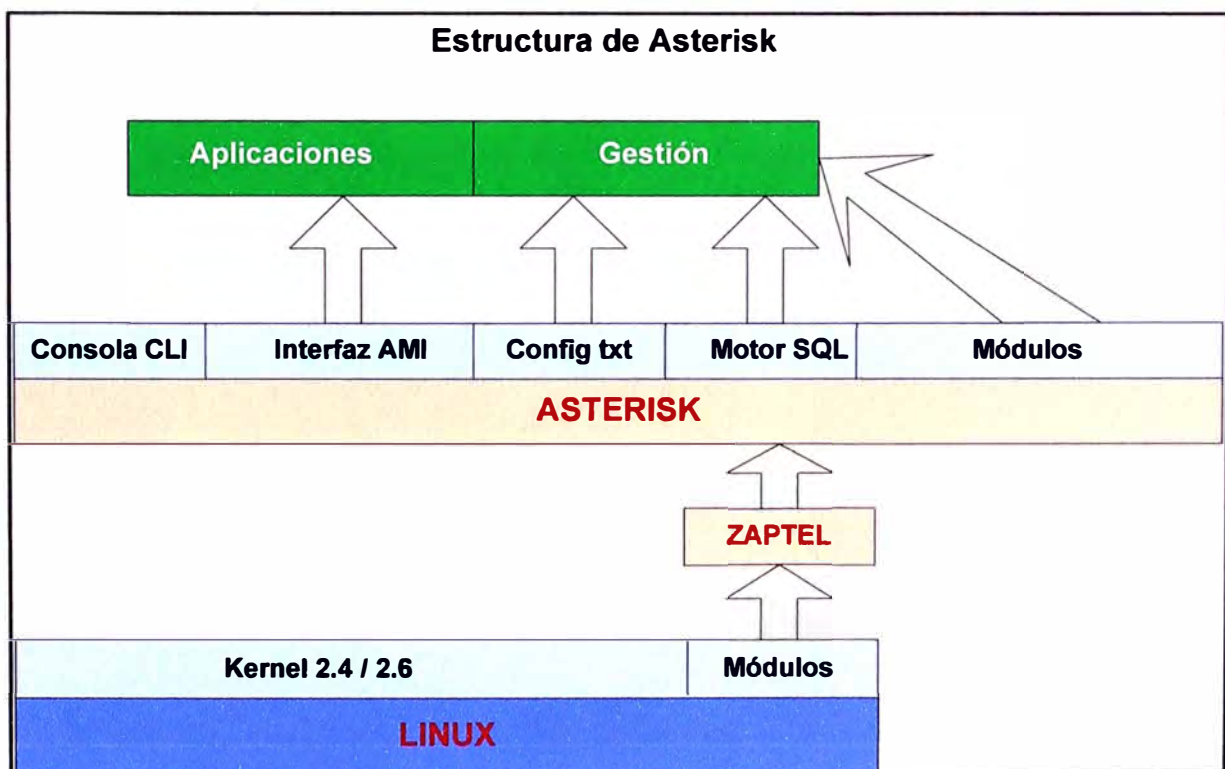


Fig. 2.7 Estructura de la Plataforma Asterisk

## **2.3. Términos y Configuraciones en Asterisk**

### **2.3.1. Configuración Inicial de Asterisk**

Los archivos sobre los que se trabajará para la configuración de Asterisk son:

**zaptel.conf:** aquí se configuran las opciones de bajo nivel de la interfaz de hardware.

Se configuran los canales FXO y FXS.

**zapata.conf:** en este archivo se configura la interfaz de asterisk con el hardware.

**extensions.conf:** es donde se arma el dial-plan de Asterisk.

**sip.conf:** aquí se configura el protocolo SIP.

**iax.conf:** aquí se configuran los canales de entrada y salida de IAX.

Luego de editar los archivos hay que recargarlos para que los cambios surtan efecto.

Luego de editar el archivo `zaptel.conf`, va a ser necesario ejecutar `/sbin/ztcfg -vv` (-vv para salidas de depuración mas completas) para que se actualicen las configuraciones del hardware.

#### **a) Canales FXO y FXS**

La diferencia entre un canal FXO y uno FXS es simplemente qué extremo de la conexión provee el tono de discado. Un puerto FXO no genera tono, acepta uno. Un ejemplo es el tono provisto por la compañía telefónica. Un puerto FXS provee tanto el tono de discado como el voltaje necesario para alertar cuando hay una llamada entrante. Si el servidor Asterisk tiene un puerto compatible FXO, se puede conectar allí la línea telefónica analógica proveniente de la PSTN (ej. línea que nos da Telecom). Asterisk puede usar esta línea tanto para recibir como para efectuar llamadas. De la misma manera, si Asterisk tiene un puerto FXS, se puede conectar un teléfono analógico en el servidor Asterisk de manera que pueda llamar y conectar llamadas por el teléfono analógico conectado.

Los puertos están definidos en la configuración por la señalización que usan de manera opuesta a lo que son físicamente. Por ejemplo, un puerto físico FXO estará definido en la configuración con señalización FXS y un puerto físico FXS estará con señalización FXO.

Cuando se configura el archivo `zaptel.conf` y se cargan los módulos, Asterisk no se entera de nada de lo que se ha configurado. El hardware no necesita ser usado por Asterisk, puede perfectamente ser usado por otra pieza del software, que hace de interfaz con los módulos de Zaptel. Se le informa a Asterisk sobre el hardware y sobre el control de las opciones asociadas a través del archivo `zapata.conf`:

```
[trunkgroups]
```

```

; define any trunk groups
[channels]
; hardware channels
; default
usecallerid=yes
hidecallerid=no
callwaiting=no
threewaycalling=yes
transfer=yes
echocancel=yes
echotraining=yes
; define channels
context=incoming ; Incoming calls go to [incoming] in extensions.conf
signalling=fxs_ks ; Use FXS signalling for an FXO channel
channel => 2 ; PSTN attached to port 2

```

La sección [trunkgroups] es para conexiones NFAS y GR-303. Si se requiere este tipo de funcionalidad, en el archivo zapata.conf.sample hay más información. La sección [channels] determina el método de señalización y sus opciones para los canales del hardware. Una vez que se define una opción, ésta es heredada hacia el resto del archivo. Un canal se define usando channel =>, cada definición de canal hereda las opciones definidas hacia arriba de esa línea. Si se desea configurar diferentes opciones para diferentes canales, éstas deben ser configuradas antes de colocar channel => Con usecallerid=yes se ha activado la función de caller ID, y se especifica que no se ocultará para llamadas salientes con hidecallerid=no. En una línea FXO se desactiva la llamada en espera con callwaiting=no. Si se activa el llamado de tres vías con threewaycalling=yes una llamada activa puede ponerse en espera con el botón flash.

#### **b) Configuración de un canal FXO/ FXS**

Se procederá de manera similar a la configuración de un canal FXO. A continuación se muestra una configuración mínima para un canal en una placa TDM400P. La configuración para el canal FXO es idéntica, pero con el agregado de fxoks=1. Recordando que se utiliza la señalización opuesta para canales FXO y FXS, entonces se configurará señalización FXO para el canal FXS. En el ejemplo siguiente se

configura el canal 1 para usar señalización FXO con el protocolo de señalización kewlstart:

(Sobre el archive de configuracion zapata.conf

fxoks=1

fxsks=2

loadzone=us

defaultzone=us

### **c) Configurando SIP**

SIP comúnmente usado en teléfonos VoIP, se encarga del establecimiento y liberación de llamadas junto con las renegociaciones durante llamadas. Básicamente ayuda a los puntos extremos a hablarse mutuamente y de ser posible de manera directa. SIP no se encarga de transportar datos multimedia, para ello hace uso del RTP una vez que la llamada se ha establecido.

SIP es un protocolo de capa de aplicación que usa un puerto bien conocido, 5060, para las comunicaciones. SIP puede ser transportado por los protocolos por User Datagram Protocol (UDP) o Transmission Control Protocol (TCP). Asterisk no maneja la implementación de TCP para transportar los mensajes SIP, pero es posible que futuras versiones lo soporten. SIP se utiliza para establecer, modificar y terminar las sesiones multimedia tales como llamadas telefónicas. SIP no transporta datos multimedia entre los puntos finales.

RTP se usa para transportar datos multimedia (ej. voz) entre los puntos finales. RTP usa números altos para sus puertos que van del 10000 al 20000 por defecto en Asterisk. Cuando un punto quiere comunicarse con el otro, primero contacta su servidor Proxy y éste último trata de contactar al usuario destino. Una vez que los teléfonos comienzan la llamada se comunican de ser posible de manera directa uno con el otro, de tal forma que los datos no tienen que ser cursados por los Proxy.

Hay otros protocolos de VoIP tales como H.323, MGCP, IAX y otros, pero SIP parece ser el que está en apogeo. La ventaja de SIP reside en su amplia aceptación y flexibilidad (y tradicionalmente en su simpleza).

El siguiente es un archivo básico de configuración sip.conf:

[general]

context=default



**srvlookup=yes**

**[john]**

**type=friend**

**secret=123456**

**qualify=yes ; par calificado si no está a mas de 2000ms de distancia**

**nat=no ; Este teléfono no está detrás de un NAT**

**host=dynamic ; Este dispositivo se registra con nosotros**

**canreinvite=no ; Asterisk por defecto trata de redireccionar**

**context=internal ; El contexto internal controla lo que se puede hacer**

El archivo sip.conf comienza con la sección [general], que contiene las configuraciones de canal y las opciones por defecto para todos los usuarios (users) y pares (peers) definidos en sip.conf. Se puede redefinir las configuraciones por defecto de manera individual para cada usuario/par en la sección de definición respectiva.

Los registros del servicio de sistema de nombre de dominio (Domain Name System Service, DNS SRV records) son una manera de establecer una dirección lógica y resoluble del lugar en el que se puede ser alcanzado. Esto permite que las llamadas sean direccionadas a diferentes ubicaciones sin la necesidad de hacer cambios en direcciones lógicas.

Cada conexión se define como user, peer o friend (usuario, par o amigo). El tipo user se usa para autenticar llamadas entrantes, el tipo peer se usa para llamadas salientes y el friend se usa para ambas. El nombre de la extensión se define con corchetes ([]). En este caso definimos la extensión john como un friend.

Un secret es una contraseña para la autenticación. En este caso se definió como 123456. Se puede monitorear la latencia entre el servidor Asterisk y el teléfono con qualify= yes, de esta manera se determina si el dispositivo remoto es alcanzable. Por defecto, Asterisk considerará una extensión o interno como alcanzable si tiene una latencia menor a 2 segundos. Reemplazando yes por el número de milisegundo en qualify=yes le decimos a Asterisk el tiempo para determinar a un dispositivo como inalcanzable.

Si un interno está detrás de un dispositivo haciendo Network Address Translation (NAT), tal como un router o firewall, hay que configurar nat=yes para forzar a Asterisk a ignorar la información de contacto y usar la dirección de donde los paquetes están siendo recibidos.

Establecer host=dynamic va a requerir que el interno se registre de manera que Asterisk sepa como alcanzar cada teléfono. Para limitar a un punto extremo a una sola dirección IP o a un nombre de dominio plenamente calificado Fully Qualified Domain

Name (FQDN), reemplazar dynamic con el IP o nombre de dominio. Esto sólo limita a donde se hacen llamadas, ya que el usuario está permitido a hacer llamadas desde donde sea (asumiendo que se ha autenticado exitosamente). Si se coloca host=static, el dispositivo no necesita registrarse.

En SIP, invites (invitaciones) se usan para establecer llamadas y direccionar los datos. Cualquier invitación emitida luego de la inicial en el mismo diálogo se la denomina como reinvoke (re invitación). Por ejemplo, si se supone que dos partes están intercambiando tráfico, si uno de los clientes es colocado en espera y Asterisk se configura para reproducir MOH, Asterisk va a emitir una invitación al segundo cliente, diciéndole que redireccione su flujo de datos hacia la PBX. Asterisk entonces puede enviar la música o un anuncio al cliente que está en espera. El primer cliente entonces emite un comando de off-hold (salida de espera) en una reinvitación a la PBX, la que a su vez, emite una reinvitación a la segunda parte pidiendo que redirija su flujo de datos hacia la primera parte y por lo tanto finaliza la MOH y reconecta los clientes.

### **2.3.2. Especificaciones de IAX**

El protocolo de Intercambio Inter Asterisk (IAX) proporciona control y transmisión de datos multimedia sobre redes IP. IAX puede ser usado con cualquier tipo de datos multimedia incluso el vídeo, pero apunta principalmente al control de llamadas de VoIP. Los objetivos de diseño para IAX se derivan de la experiencia con los protocolos de VoIP existentes como SIP y MGCP para control, y RTP para la transmisión de datos multimedia.

Los objetivos de diseño primarios para el protocolo IAX son:

- Minimizar el uso de ancho de banda tanto para control como para datos multimedia, con especial énfasis en las llamadas de voz individuales.

- Proporcionar soporte transparente frente a un NAT.

IAX es un protocolo de datos multimedia y control par a par. Par a par significa que los puntos extremos mantienen máquinas de estados asociadas con las operaciones del protocolo. El componente de señalización del protocolo IAX es un protocolo de control de llamadas Maestro-Esclavo, más análogo al Protocolo de Iniciación de Sesión (SIP) que al Protocolo de Control de Entrada de Media (MGCP). Con respecto a datos multimedia, la información de secuencia y cronometraje es incluida en los marcos IAX.



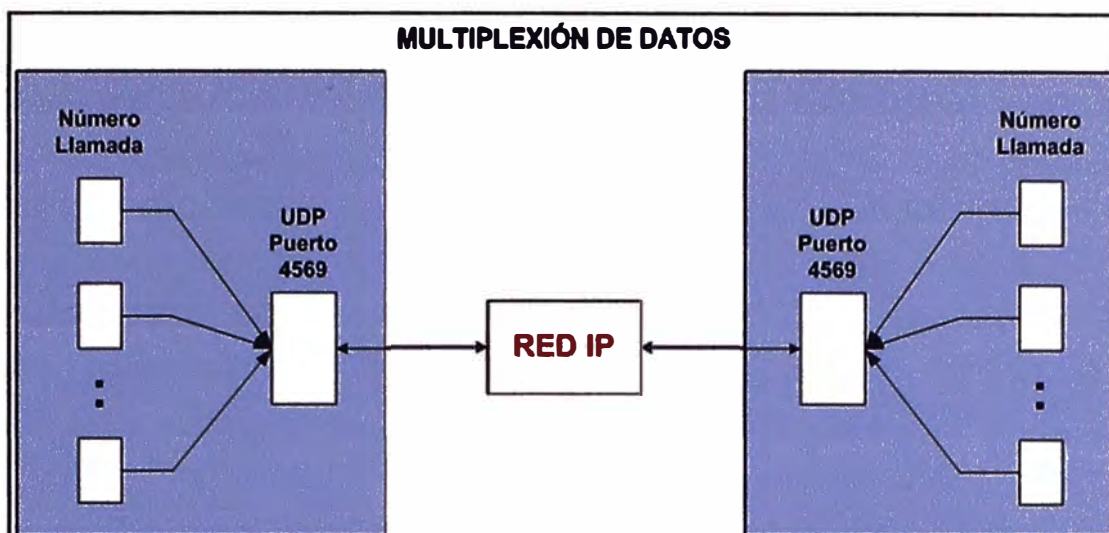


Fig. 2.8 Multiplexación de datos IAX

El transporte de multimedia no utiliza el Protocolo de Transporte en Tiempo real (RTP). El diseño básico fue realizado para la señalización IAX multiplexada y múltiples corrientes de datos multimedia sobre una única asociación UDP entre dos anfitriones de Internet. Es realmente dos protocolos en uno: un protocolo para sesiones de señalización y un protocolo para transportar las corrientes de datos multimedia. Esto difiere totalmente con la arquitectura de otros protocolos IETF que separan el control (MGCP y SIP) de la transmisión de datos (RTP/RTCP). Como la señalización y los datos multimedia comparten el mismo número de puerto UDP, IAX hace que no se sufran los problemas transversales relacionados con el uso de un NAT.

### 2.3.3. Fundamentos del dial plan

El dial plan está especificado en el archivo de configuración llamado `extensions.conf`. El mismo está ordenado en cuatro partes principales: contextos, extensiones, prioridades y aplicaciones. En las siguientes secciones, cubriremos cada una de estas partes y explicaremos como trabajan juntas para crear un dial plan. Después de explicar el papel que cada uno de estos elementos juega en el dial plan, entraremos en el proceso de crear un dial plan básico.

#### a) Contextos

Los dialplans están divididos en secciones llamadas contextos. Los contextos son grupos de extensiones. Una extensión que es definida en un contexto está completamente aislada de extensiones en cualquier otro contexto, a menos que la interacción sea expresamente permitida.

Como un ejemplo simple, imaginemos que tenemos dos compañías que comparten un servidor Asterisk. Si colocamos el menú de voicemail de cada compañía en su propio contexto, ellos han sido eficientemente separados del otro. Esto permite que se defina independientemente lo que pasa cuando, supongamos, la extensión 0 es marcada: las personas que presionen el 0 en el menú de voicemail de la Compañía A, hablarán con la recepcionista de la Compañía A, y visitantes que presionen 0 en el menú de voicemail de la Compañía B, hablarán con la recepcionista de la Compañía B. (Este ejemplo asume, por supuesto, que se ha configurado Asterisk para transferir las llamadas a los recepcionistas cuando los visitantes presionan 0).

Los contextos son denotados colocando el nombre del mismo entre corchetes ([ ]). El nombre puede ser cualquier cosa, utilizando desde la letra A hasta la Z (minúscula o mayúscula), los números de 0 a 9 y el guión bajo o alto. Por ejemplo, un contexto para las llamadas entrantes podría ser: [incoming]

Todas las instrucciones colocadas después de una definición de contexto son parte de aquel contexto, hasta que el siguiente contexto es definido. Al principio del dial plan, hay dos contextos especiales llamados [general] y [globals]. Más adelante en este trabajo los describiremos.

## **b) Extensiones**

Dentro de cada contexto, definimos una o varias extensiones. Una extensión es una instrucción que Asterisk seguirá, provocado por una llamada entrante o por dígitos marcados en un canal. Las extensiones especifican lo que le pasa a las llamadas cuando ellas recorren su camino por el dial plan. Aunque las extensiones puedan ser usadas para especificar extensiones telefónicas en el sentido tradicional (es decir, por favor llame a Juan en la extensión 153), ellas pueden ser usadas para mucho más en Asterisk.

La sintaxis para una extensión es la palabra `exten` seguida de una flecha formada por el signo igual y el signo *mayor que*: `exten =>`

Esto es seguido del nombre de la extensión. Tratando con sistemas telefónicos, tendemos a pensar en extensiones como los números que se marcarían para hacer que el otro teléfono suene. Sin embargo en Asterisk siempre se podrá utilizar números, letras, o combinaciones de estos. Una extensión completa está formada de tres componentes: El nombre (o número) de la extensión, La prioridad (cada extensión puede incluir pasos múltiples; el número de paso es llamado "prioridad") y La aplicación (o comando) que ejecuta alguna acción en la llamada. Estos tres componentes son separados por comas, como esto:

exten => nombre, prioridad, aplicación ( )

Aquí se muestra un ejemplo simple de cómo podría lucir una verdadera extensión:

exten => 123,1, Answer ( )

En este ejemplo, el nombre de extensión es 123, la prioridad es 1 y la aplicación es Answer ( ).

### c) Prioridades

Cada extensión puede tener pasos múltiples, llamados prioridades. Cada prioridad es numerada secuencialmente, comenzando con 1. Cada prioridad ejecuta una aplicación específica. Como ejemplo, la extensión siguiente contestaría el teléfono (en la prioridad número 1) y luego lo cuelga (en la prioridad número 2):

exten => 123,1, Answer ( )

exten => 123,2, Hangup ( )

Asterisk sigue las prioridades en orden numérico.

### d) Aplicaciones

Las aplicaciones son las piezas fundamentales del dial plan. Cada aplicación realiza una acción específica en el canal, como reproducir un sonido, aceptar una entrada marcada o la finalización de una llamada.

*Por ejemplo, el dial plan podría verse como lo siguiente:*

exten => 123,1, Answer ( )

exten => 123, n, hace algo

exten => 123, n, hace algo más

exten => 123, n, hace una última cosa

exten => 123, n, Hangup ( )

*Una etiqueta a una prioridad, simplemente añade texto dentro de un paréntesis después de la prioridad, como esto:*

exten => 123, n (etiqueta), hace algo

Algunas aplicaciones, como Answer ( ) y Hangup ( ), no necesitan ninguna otra instrucción para hacer su trabajo. Otras aplicaciones requieren información adicional. Estas informaciones, llamadas argumentos, pueden ser pasadas a las aplicaciones para indicar como deben ejecutar las acciones. Para pasar argumentos a una aplicación, se los deberá colocar entre paréntesis seguidos del nombre de la aplicación, separados por comas.

### e) La Extensión s

Cuando las llamadas entran en un contexto sin una extensión de destino específica (por ejemplo, que suene una línea FXO), ellas son manejados automáticamente por la extensión s. (La s significa *start*, ya que la mayor parte de las llamadas empiezan en la extensión s.) Ya que esto es exactamente lo que necesitamos para el dial plan, vamos a comenzar a unir los pedazos. Ejecutaremos tres acciones en la llamada (contestar, reproducir sonido y colgar), entonces necesitamos crear una extensión s con tres prioridades.

Colocaremos las tres prioridades dentro [entrantes], ya que todas las llamadas entrantes deberían comenzar en este contexto:

[incoming]

exten => s, 1, aplicación ( )

exten => s, 2, aplicación ( )

exten => s, 3, aplicación ( )

Ahora todo lo que tenemos que hacer es rellenar las aplicaciones y hemos creado el dial plan.

### f) Las aplicaciones Answer ( ), Playback( ) y Hangup( )

La aplicación Answer ( ) es usada para contestar un canal que está sonando. Como mencionamos antes, Answer ( ) no tiene argumentos. La aplicación Playback ( ) es usada para reproducir un sonido previamente grabado en un canal. Usando la aplicación Playback ( ), la entrada digitada por el usuario es simplemente ignorada.

Para usar Playback ( ), se debe especificar el nombre del archivo (sin una extensión de archivo) como argumento. Por ejemplo, Playback (nombre) reproducirá el archivo de sonido llamada nombre.gsm, asumiendo que está ubicado en el directorio de sonidos por defecto. Se puede incluir el camino completo al archivo si se quiere:  
Playback (/home/juan/sounds/nombre)

Este ejemplo reproducirá nombre.gsm desde el directorio /home/juan/sounds/. Se puede usar también caminos relativos del directorio de sonidos de Asterisk:

Playback (custom/nombre)

Este ejemplo reproducirá nombre.gsm desde el subdirectorio de sonidos por defecto custom/. Si el directorio especificado contiene más de un archivo con aquel nombre pero con extensiones de archivo diferentes, Asterisk automáticamente reproduce el

mejor archivo (el que utilizará menos el procesador de la computadora para convertirlo a su formato nativo).

La aplicación Hangup ( ) hace exactamente lo que su nombre implica: cuelga el canal activo.

### **g) Las aplicaciones Background ( ) y Goto ( )**

Una clave importante al edificar un sistema Asterisk interactivo es la aplicación Background( ). Como Playback ( ), reproduce un archivo de sonido registrado. A diferencia de Playback ( ) cuando el visitante presiona un dígito (o una serie de dígitos) en su teclado numérico telefónico, esto interrumpe la reproducción y dirige la llamada a la extensión que corresponde al dígito/s marcado. Si un visitante aprieta 5, por ejemplo, Asterisk dejará de reproducir el sonido y enviará el control de la llamada a la primera prioridad de la extensión 5.

Background ( ) tiene la misma sintaxis que Playback ( ):

exten => 123,1, Background (Bienvenida)

Otra aplicación útil es Goto ( ). Como su nombre lo indica, es usada para enviar la llamada a otro contexto, extensión o prioridad. La aplicación Goto ( ) hace fácil mover mediante configuración una llamada entre partes diferentes del dial plan. La sintaxis para Goto ( ) pide que se pase el contexto, la extensión o la prioridad de destino como argumentos a la aplicación:

exten => 123,1, Goto (contexto,extensión,prioridad)

### **h) La aplicación Dial ( )**

Uno de los rasgos más valiosos de Asterisk es su capacidad de conectar a diferentes usuarios entre ellos. Esto es especialmente útil cuando usan diferentes métodos de comunicación. La sintaxis de la aplicación Dial ( ) es un poco más compleja que las otras aplicaciones que hemos usado hasta ahora. Dial ( ) tiene hasta cuatro argumentos. El primero es el destino que se intenta llamar, compuesto por la tecnología (o transporte) que se utiliza para hacer la llamada, una barra oblicua y el recurso remoto (por lo general el nombre de un canal o número). Por ejemplo, asumamos que queremos llamar a un canal Zap llamado Zap/1, que es un canal FXS con un teléfono análogo. La tecnología es Zap y el recurso es 1. Del mismo modo, una llamada a un dispositivo SIP podría tener un destino SIP/1234, y una llamada a un

dispositivo IAX podría tener un destino IAX/Pedro. Si queremos que Asterisk haga sonar el canal ZAP/1 cuando la extensión 123 es alcanzada en el dial plan, añadiríamos la siguiente siguiente:

```
exten => 123,1, Dial (Zap/1)
```

Cuando esta extensión sea ejecutada, Asterisk hará sonar el teléfono conectado en el canal ZAP/1. Si aquel teléfono es contestado, Asterisk tenderá un puente sobre la llamada entrante con el canal Zap/1. También podemos marcar múltiples canales al mismo tiempo, concatenando los destinos con el signo &, de esta forma:

```
exten => 123,1, Dial (Zap/1&Zap/2&Zap/3)
```

La aplicación Dial ( ) tenderá un puente sobre la llamada entrante con cualquier canal destino que sea contestado primero.

El segundo argumento de la aplicación Dial ( ) es un intervalo de espera, especificado en segundos. Si un intervalo de espera es dado, Dial ( ) intentará llamar al destino durante aquel número de segundos antes de dejar la llamada y pasar a la siguiente prioridad en la extensión. Si no se especifica intervalo de espera, Discar ( ) continuará llamando al canal hasta que alguien atienda o el visitante cuelgue. Vamos a añadir un intervalo de espera de 10 segundos a la extensión:

```
exten => 123,1, Dial (Zap/1,10)
```

Si la llamada es contestada antes del intervalo de espera, se tiende un puente sobre los canales y el dial plan está ejecutado. Si el destino simplemente no contesta, Dial ( ) continúa con la siguiente prioridad en la extensión. Sin embargo, si el canal de destino está ocupado, Dial ( ) irá a la prioridad n+101, si esta existe (donde n es la prioridad desde donde la aplicación Dial es llamada). Esto permite que manejemos llamadas sin contestar de forma diferente a las llamadas cuyos destinos estaban ocupados.

El tercer argumento para Dial ( ) es una cadena de opciones. Puede contener uno o varios caracteres que modifican el comportamiento de la aplicación Dial ( ). Mientras que la lista de las posibles opciones es demasiado larga para cubrir aquí, la opción más popular es la letra r. Si se coloca la letra r como el tercer argumento, la persona que llama oirá un tono de llamada mientras el canal de destino está siendo notificado de una llamada entrante.

Un ejemplo del uso de las opciones de Dial se muestra a continuación:

```
exten => 123,1, Dial (Zap/1,10, r)
exten => 123,2, Playback (vm-no-contesta)
exten => 123,3, Hangup ( )
exten => 123,102, Playback (tt-ocupado)
exten => 123,103, Hangup ( )
```

### **i) Utilización de Variables**

Se puede pensar en una variable como un contenedor que puede tener solo un valor a la vez. De este modo, por ejemplo, podríamos crear una variable llamada JUAN y adjudicarle el valor Zap/1. De esta forma, cuando escribimos nuestro dial plan, podemos referirnos al canal de Juan por el nombre, en vez de recordar que Juan usa Zap/1. Para signar un valor a una variable, simplemente hay que escribir el nombre de la variable, un signo igual y el valor, como se muestra:

```
JUAN=Zap/1
```

Hay dos modos de referirse a una variable. Para referirse al nombre de la variable, simplemente se escribe el nombre de la variable, como JUAN. Si, por otra parte, se quiere referir su valor, se debe escribir un signo de dólar, abrir paréntesis, el nombre de la variable y cerrar paréntesis. Aquí se muestra como nos referiríamos a la variable dentro de la aplicación Dial ( ):

```
exten => 555,1, Dial ($ { JUAN },, r)
```

En nuestro dial plan, siempre que escribamos \${ JUAN }, Asterisk lo sustituirá automáticamente con cualquier valor que haya sido adjudicado a la variable llamada JUAN.

### **j) Correspondencia de Patrones**

Con la correspondencia de patrones, usamos diferentes letras y símbolos para representar los posibles dígitos que queremos corresponder. Los patrones siempre comienzan con un guión bajo (`_`). Este le dice a Asterisk que corresponderemos un patrón, y no el nombre de una extensión. Esto significa, por supuesto, que nunca se deberá comenzar los nombres de extensión con un guión bajo.



Después del guión bajo, se puede usar uno o varios de los siguientes caracteres:

X Corresponde con cualquier dígito de 0 a 9.

Z Corresponde con cualquier dígito de 1 a 9.

N Corresponde con cualquier dígito de 2 a 9.

[15-7] Corresponde con cualquier dígito o rango de dígitos especificados. En este caso, corresponde con 1, 5, 6, o 7.

El "." (punto) comodín de correspondencia; corresponde con uno o varios caracteres.

### **k) Permitiendo llamadas salientes**

Añadiremos una variable al contexto [globals] para definir qué canal será usado para llamadas salientes:

```
[globals]
```

```
Juan=Zap/1
```

```
Julia=SIP/julia
```

```
OUTBOUNDTRUNK=Zap/4
```

Después, añadiremos contextos a nuestro dial plan para la marcación saliente. Podríamos preguntarnos en este punto, por qué tenemos que separar contextos para llamadas salientes. Este es el modo a través del cual podemos regular y controlar quién tiene permiso para hacer llamadas, y qué tipos de llamadas salientes se permiten hacer.

Primero, vamos a hacer un contexto para llamadas locales. Para ser consecuente con los conmutadores telefónicos tradicionales, pondremos un 9 en el frente de nuestros patrones, de modo que los usuarios tengan que marcar 9 antes de llamar a un número exterior:

```
[outbound-local]
```

```
exten => _9NXXXXXX, 1, Dial (${OUTBOUNDTRUNK}/${EXTEN:1})
```

```
exten => _9NXXXXXX, 2, Congestion ( )
```

```
exten => _9NXXXXXX, 102, Congestion ( )
```

Vamos a examinar lo que acabamos de hacer. Hemos añadido una variable global llamada OUTBOUNDTRUNK, que controlará que canal se usará para llamadas salientes. Tenemos también añadido un contexto para llamadas locales. En la prioridad 1, tomamos la extensión marcada, sacamos el número 9 con la sintaxis \${EXTEN:1}, y luego se intenta marcar aquel número en el canal designado por la variable OUTBOUNDTRUNK. Si la llamada es exitosa, se tiende un puente entre quien

llama y el canal de salida. Si la llamada fracasa (porque el canal está ocupado o el número no puede ser marcado por la razón que sea), la aplicación Congestión ( ) es llamada, que reproduce una señal de ocupado (tono de congestión) para hacer saber al usuario que la llamada no se puede realizar.

## **CAPITULO III**

### **METODOLOGÍA PARA EL DISEÑO DE LA SOLUCIÓN**

#### **3.1. Estudio de la Solución**

En términos de requerimientos de recursos, Asterisk necesita, tal como los sistemas embebidos, aplicaciones de tiempo real. Esto es debido en gran parte a su necesidad de tener acceso prioritario al procesador y los buses del sistema. Es por lo tanto imperativo que cualquier función en el sistema que no esté directamente relacionada con la tarea de procesamiento de llamadas de Asterisk sea ejecutado con baja prioridad. En sistemas pequeños, esto no debería ser un aspecto problemático. Sin embargo, en sistemas de alta capacidad, una performance defectuosa se manifestará en problemas de calidad de audio para los usuarios, tales como eco o estática, entre otros. Los síntomas se parecerán a aquellos experimentados en un teléfono celular cuando se lo utiliza fuera de un área de cobertura, aunque las causas que los provocan serán diferentes. Mientras la carga se incrementa, el sistema tendrá mayores dificultades para mantener las conexiones.

Flexibilidad es la razón principal por la que Asterisk es extremadamente efectivo en relación a los costos, frente a negocios que crecen rápidamente. Si bien es posible algún tipo de escalabilidad en la mayoría de los sistemas telefónicos, todavía no se conoce alguno que pueda crecer sin generar grandes costos. Habiendo dicho esto, vale aclarar que sistemas distribuidos con Asterisk no son simples de diseñar, por lo que no sería una tarea para alguien nuevo en la utilización de este producto.

##### **3.1.1. Selección del Servidor para la PBX Asterisk**

La selección de un servidor es simple y complicado a la vez: simple porque, realmente, cualquier plataforma basada en x86 será suficiente; pero complicado porque la performance segura del sistema dependerá del cuidado que se tenga en el diseño de la plataforma. Cuando se seleccione el hardware, se debe considerar cuidadosamente el diseño global del sistema y que funcionalidad se necesita proveer. Esto ayudará en

la determinación de los requerimientos para el CPU, placa madre y abastecimiento de energía

### **3.1.2. Aspectos de Performance**

Entre otras consideraciones, cuando se selecciona el hardware para una instalación de Asterisk se debe tener en mente la siguiente pregunta crítica: ¿que tan poderoso debe ser el sistema?. Ésta no es una pregunta fácil de responder, porque la manera en que el sistema será usado jugará un rol importante en los recursos que consumirá. Se deberá por lo tanto comprender cómo Asterisk utiliza el sistema para tomar decisiones inteligentes sobre qué tipo de recursos se necesitarán.

Se deberán considerar varios factores, incluyendo: El máximo número de conexiones simultáneas que el sistema podrá proveer: Cada conexión incrementará la carga de trabajo en el sistema. El porcentaje de tráfico que requerirá un procesamiento intensivo de codecs de compresión (como el G.729 o GSM) por parte del Procesador de Señal Digital DSP: el trabajo de procesamiento de señales digitales que Asterisk realiza mediante software puede tener un fuerte impacto en el número de llamadas simultáneas que proveerá. Un sistema que puede fácilmente manejar 50 llamadas G.711 simultáneas, puede caer frente al requerimiento de atender 10 canales comprimidos mediante G.729.

Lógica de escritura del dial plan: Cada vez que Asterisk tiene que pasar el control de las llamadas a un programa externo, existe una penalidad en la performance. La mayor lógica posible debe ser construida en el dial plan. Si scripts externos son utilizados, éstos deben ser diseñados teniendo a la performance y eficiencia como metas importantes.

Si bien estos factores deben tenerse en consideración, el impacto en la performance de los mismos todavía no es conocido con exactitud. Los efectos de cada uno son conocidos en términos generales, pero no se ha definido un método de calcular la performance preciso. Esto es en parte debido a que los efectos de cada componente del sistema dependen de numerosas variables, como el poder de procesamiento del CPU, la calidad de la placa madre y el chipset en general, la carga total de tráfico sobre el sistema, la optimización del kernel de Linux, el tráfico de red, el número y tipo de interfaces PSTN, y el tráfico PSTN (sin mencionar cualquier servicio no perteneciente a Asterisk que el sistema esté ejecutando simultáneamente).

### **3.1.3. Selección de Procesador**

El procesamiento de señal que Asterisk realiza puede fácilmente demandar una asombrosa cantidad de cálculos matemáticos complejos por parte del CPU. La eficiencia con que estas tareas son llevadas a cabo será determinada por el poder de la FPU dentro del procesador. Para establecer actualmente el mejor procesador para Asterisk debemos situarnos en un contexto en donde la industria de la computación avanza rápidamente. Periódicamente, quizás cada día, la velocidad de los procesadores es mejorada, sumado al hecho de que Asterisk corre sobre arquitecturas varias. Obviamente, esto es algo bueno, pero también hace que este punto sea más que una simple tarea. Naturalmente, mientras más poderosa sea la FPU, más tareas de DSP Asterisk podrá manejar simultáneamente, por lo que ésta sería la consideración más importante. Cuando se está seleccionando un procesador, la velocidad del clock es solo una parte de la ecuación. Qué tan bien se manejen las operaciones de punto flotante será una clave diferenciadora, ya que las operaciones de DSP en Asterisk establecerán una gran demanda de ellas.

### **3.1.4 Selección de la Placa Madre**

La clave es tener tanto estabilidad como alto rendimiento. Aquí se mencionan algunas directrices a seguir:

Los buses del sistema deben proporcionar la mínima latencia posible. Si se planea una conexión PSTN usando interfaces analógicas o Primary Rate Interface (PRI), las placas Digium Zaptel generarán 1,000 peticiones de interrupción por segundo. Tener dispositivos sobre el bus que interfieran con este proceso causará la degradación de la calidad de llamada. Chipsets de Intel (para CPU de Intel) y nVidia nForce (para CPU AMD) parecen anotar las mejores marcas en esta área.

Dependiendo de que placa se compre, se deberá conocer si se requiere 3.3V o 5V. La mayor parte de placas madre para servidores tendrán ambos tipos, pero los terminales de trabajo típicamente tendrán sólo la versión de 5V.

### **3.1.5. Hardware de Telefonía**

Si se va a conectar Asterisk con algún equipo de telecomunicaciones, se necesitará el hardware correcto. El hardware que se requiera será determinado por lo que se busca proveer con el sistema.

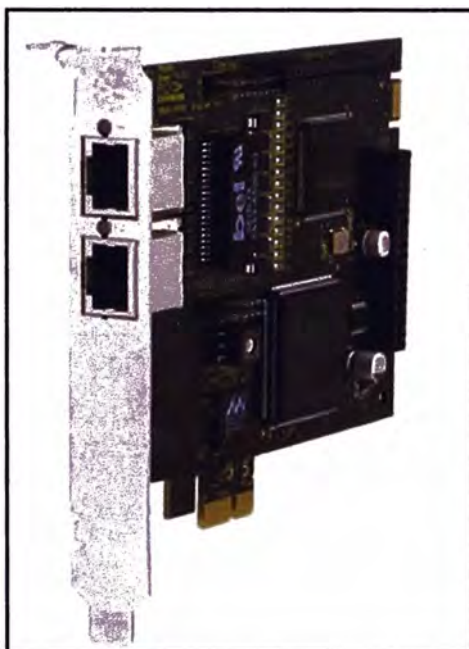
#### **a) Conexión a la PSTN**

Asterisk permite fácilmente tender un puente entre redes de conmutación de circuitos y redes de conmutación de paquete. A causa de la arquitectura abierta de Asterisk (y

código fuente abierto), es posible interconectar interfaces de hardware de cualquier estándar.

### **b) Conexión E1/T1**

Existe una gran variedad de fabricantes de tarjetas para conexión E1 con asterisk, entre ellos podemos mencionar a Digium, Sangoma, entre otros. Para la presente solución se ha escogido trabajar con Digium por ser la de más uso en el mercado Latinoamericano. Con respecto a las tarjetas E1 Digium, existen desde un puerto de conexión E1 hasta 4 puertos E1. En las próximas páginas veremos cual es la tarjeta adecuada para la implementación que deseamos desarrollar, así como su respectiva configuración. La Figura 3.1 muestra una tarjeta digital para conexión E1 en la PBX Asterisk.



**Fig. 3.1** Tarjeta de telefonía E1 para Asterisk ([www.digium.com](http://www.digium.com))

### **c) Conexión FXO/FXS**

Estas tarjetas que permiten la conexión de módulos analógicos también son de gran variedad, existiendo desde tarjetas de de 1 solo puerto, hasta tarjetas de 8 puertos de conexión. Para el presente proyecto, se ha considerado el uso de una de estas tarjetas en el diseño para los anexos analógicos administrativos. La Instalación y configuración de estas tarjetas se cubre más adelante. La figura 3.2 muestra una tarjeta para conexión analógica para Asterisk.

#### d) Teléfonos Físicos

Cualquier dispositivo físico cuyo objetivo primario sea terminar una petición de audio entre dos puntos puede ser clasificado como un teléfono físico. Como mínimo, tal dispositivo tiene un auricular/micrófono (*handset*) y un dial; también puede tener algunas otras características, como una pantalla y varias interfaces de audio.

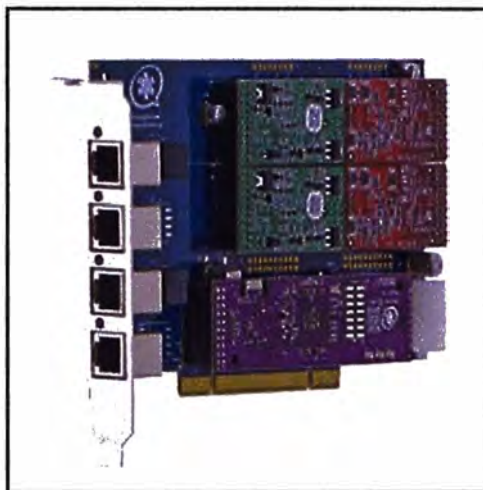


Fig. 3.2 Tarjeta de telefonía analógica para Asterisk ([www.digium.com](http://www.digium.com))

#### e) Teléfonos por Software (Softphones)

Un teléfono por software es un programa que proporciona la funcionalidad telefónica en un dispositivo no telefónico, como una computadora o un PDA (PDA). Cualquier software que corre en una computadora, que presenta la sensación visual de un teléfono y provee como su función primaria la capacidad de hacer y recibir comunicaciones full dúplex de audio es un softphone.

### 3.2. Instalando los componentes

#### 3.2.1. Instalando Asterisk

Asterisk utiliza tres paquetes principales: el programa principal Asterisk (*asterisk*), los drivers de telefonía Zapata (*zaptel*), y las librerías PRI (*libpri*). Si es necesaria una red pura de VoIP, el único requerimiento es el paquete asterisk. Los drivers zaptel son necesarios si se utiliza hardware analógico o digital, o si se utiliza el driver ztdummy (se explicará mas adelante) como interfaz de temporización. La librería libpri es técnicamente opcional a menos que se utilice una interfaz ISDN PRI, pero se recomienda que sea instalada en conjunto con el paquete zaptel.



Para compilar Asterisk, se debe instalar el compilador Compilador de C (GCC) Versión 3.x o posterior) y sus dependencias. Mientras la versión 2.96 del GCC puede funcionar por ahora, versiones futuras no lo harán. Asterisk también requiere bison y ncurses para funcionalidad Command-Line Interface (CLI). La librería de criptografía en Asterisk requiere OpenSSL y sus paquetes de desarrollo. Si se necesita usar para el temporizador ztdummy, o cualquier driver de hardware provisto por Zaptel, será necesario instalar el paquete zaptel. Si está instalando libpri, asegurarse de instalarlo antes que asterisk.

#### **a) El driver ztdummy**

En Asterisk, ciertas aplicaciones requieren dispositivos de temporización para operar. Todos los elementos de hardware PCI Digium proveen una interfaz de tiempo a 1kHz. Si carece del hardware PCI requerido para proveer tiempo, el driver ztdummy se puede utilizar como dispositivo de tiempo. En las distribuciones basadas en el kernel 2.4, ztdummy debe usar el reloj provisto por el controlador UHCI USB.

#### **b) Drivers telefónicos Zapata**

Compilar los drivers telefónicos Zapata para utilizarlos con el hardware Digium es sencillo. Simplemente correr make para el kernel 2.4 o 2.6 (el MakeFile determinará la versión del kernel). Usar estos comandos para compilar Zaptel (reemplace *version* con la versión de zaptel):

```
# cd /usr/src/zaptel-version
```

```
# make clean
```

```
# make
```

```
# make install
```

En servidores corriendo Debian puede suceder que exista un error de compilación (directory not found) al ejecutar make para construir los drivers zaptel. En este caso puede intentarse proceder como se detalla a continuación. Corroborar que estén instalados los encabezados del kernel (kernel-headers) que coincidan exactamente con el kernel que se tiene instalado.

#### **c) Usando ztcfg y zttool**

Dos programas instalados junto con Zaptel son ztcfg y zttool. El programa ztcfg se utiliza para leer la configuración en /etc./zaptel.conf y configurar el hardware. El otro programa se puede usar para verificar el estado del hardware que ha sido instalado.

Por ejemplo, si se usa una tarjeta T1 y no hay comunicación entre los puntos finales deberá ver una alarma roja. Si todo está configurado correctamente y la comunicación

es posible, vera un *OK*. La aplicación *ztool* también sirve para tarjetas analógicas, ya que nos dice el estado actual (configurada, colgada, etc.).

#### **d) Compilando libpri**

Compilar e instalar *libpri* sigue el mismo patrón que lo antes descrito para *zaptel*. *Libpri* es utilizada por varios hardwares TDM, pero aunque no se tenga el hardware instalado es recomendable compilar e instalar esta librería. Se debe compilar e instalar la librería *libpri* antes que *Asterisk*. Aquí los comandos:

```
# cd /usr/src/libpri-version
# make clean
# make
# make install
```

#### **e) Compilando Asterisk**

Con los paquetes *zaptel* y *libpri* compilados e instalados, podemos centrarnos en *Asterisk*. *Asterisk* esta compilado con *gz* a través del programa *make*. Para comenzar a compilar *Asterisk*, hay que ejecutar las siguientes líneas de código:

```
# cd /usr/src/asterisk-version
# make clean
# make
# make install
# make samples
```

Se realiza el comando *make samples* para instalar los archivos de configuración por defecto. Una vez más, si se utiliza un sistema que hace uso de los directorios */etc./rc.d/init.d* o */etc./init.d*, deberá correr también el comando *make config*. Esto instala los scripts de arranque y configura el sistema para ejecutar *Asterisk* automáticamente al iniciar.

#### **f) Plugins de utilidad**

El paquete *asterisk-addons* contiene código que permite almacenar el detalle de las llamadas Call Detail Recorder (CDR) dentro de una base de datos My Structured Query Language (MySQL) y en MP3 nativo, también como un interprete para cargar código Perl en la memoria para la vida de un proceso *Asterisk*. Los programas están en *asterisk-addons* donde puede encontrar allí permisos de licencia para prevenir que éstos sean mal implementados directamente en el código fuente de *Asterisk*.

### **g) Comandos CLI**

El archivo binario de Asterisk se encuentra por defecto en `/usr/sbin/asterisk`. Si se ejecuta `/usr/sbin/asterisk`, éste se cargara como un demonio. Hay algunos pocos comandos que se deben tener en cuenta ya que le permiten conectar/reconectar al CLI de Asterisk, establecer el tipo de salida del CLI, y permitir verificar cosas si Asterisk colapsa (para depurar con gdb).

Para explorar el rango completo de opciones, ejecutar Asterisk con `-h`:

```
# /usr/sbin/asterisk -h
```

Acá hay algunas de las opciones más comunes:

- c** Console. Le permite conectarse al CLI Asterisk
- v** Verbosity. Se utiliza para establecer la cantidad de salida para depurar CLI.
- r** Remote. Esto se utiliza para reconectar remotamente a un proceso de Asterisk corriendo actualmente.

## **CAPÍTULO IV**

### **ANÁLISIS Y DISEÑO PARA LA IMPLEMENTACIÓN**

Para el Presente proyecto, analizaremos la implementación de la plataforma tecnológica de un Call Center y estableceremos los requerimientos técnicos necesarios.

#### **4.1. Análisis descriptivo del caso de estudio**

Los requerimientos técnicos solicitados para el Call Center son: 15 puestos de atención Inbound, 15 puestos de atención outbound. Distribución de llamadas mediante ACD. IVR de acuerdo a las especificaciones de atención (horario de atención de lunes a viernes de 08:00 a 20:00), buzón de voz para fuera de horario y superado el tiempo de encolamiento. Además para el cálculo de los canales de atención, se ha solicitado los datos: Cantidad de llamadas en la hora pico y tiempo medio de duración (TMO). Es necesario grabar las llamadas de la campaña INBOUND. Con estos datos comenzamos a parametrizar el diseño.

##### **4.1.1. Gestión de llamadas Inbound:**

Se requiere soportar un tráfico Inbound en la Hora pico de 210 llamadas en la hora pico con un TMO de 4min. Haciendo el cálculo Erlangs, se tiene:

$$\#Canales\ IN = Erlang((\#llamadasHP * TMO) / 3600)$$

Donde:

**#llamadasHP:** Número de llamadas totales en la hora pico.

**TMO:** Tiempo medio de operación por llamada en segundos

$$\#Canales\ IN = Erlang((210 * 4 * 60) / 3600)$$

$$\#Canales\ IN = Erlang(14)$$

$$\#Canales\ IN = 17$$

Este último resultado hace que necesariamente se tenga que disponer de un primario RDSI (30 canales) para soportar las 17 llamadas en horas pico. Adicionalmente, se requiere que las llamadas ingresantes dentro de horario de atención (08:00 – 20:00)

accedan a un menú de 2 opciones (Consultas por Equipo A y Consultas por equipo B), además cuando no haya agente disponible (se contarán con 15 puestos de atención IN), se debe enrutar la llamada a un Buzón de voz. En fuera de horario, las llamadas también serán transferidas al buzón de voz.

#### **4.1.2. Gestión de llamadas OUT**

Para la gestión de llamadas OUT, hay que tener en cuenta que serán 15 posiciones de atención las que realizarán las llamadas, esto hace que se asigne un primario RDSI adicional asignado solo para salida. Por el momento las marcaciones se asumen de modo manual pero es factible incorporar módulos de marcación automática. Se sobreentiende que será necesario contar con los registros de llamada CDR (call Deatiled Record) para realizar seguimiento a las llamadas realizadas por cada agente.

#### **4.1.3. Grabación de Voz**

Se debe disponer de un método de grabación de voz del tipo Full recorder, es decir, el audio de las llamadas entrantes comenzará a grabarse desde el instante en que el cliente recibe respuesta por parte de la PBX Asterisk. Mas adelante detallaremos como el formato del nombre del archivo que se crea con la grabación.

#### **4.1.4 Aspectos técnicos.**

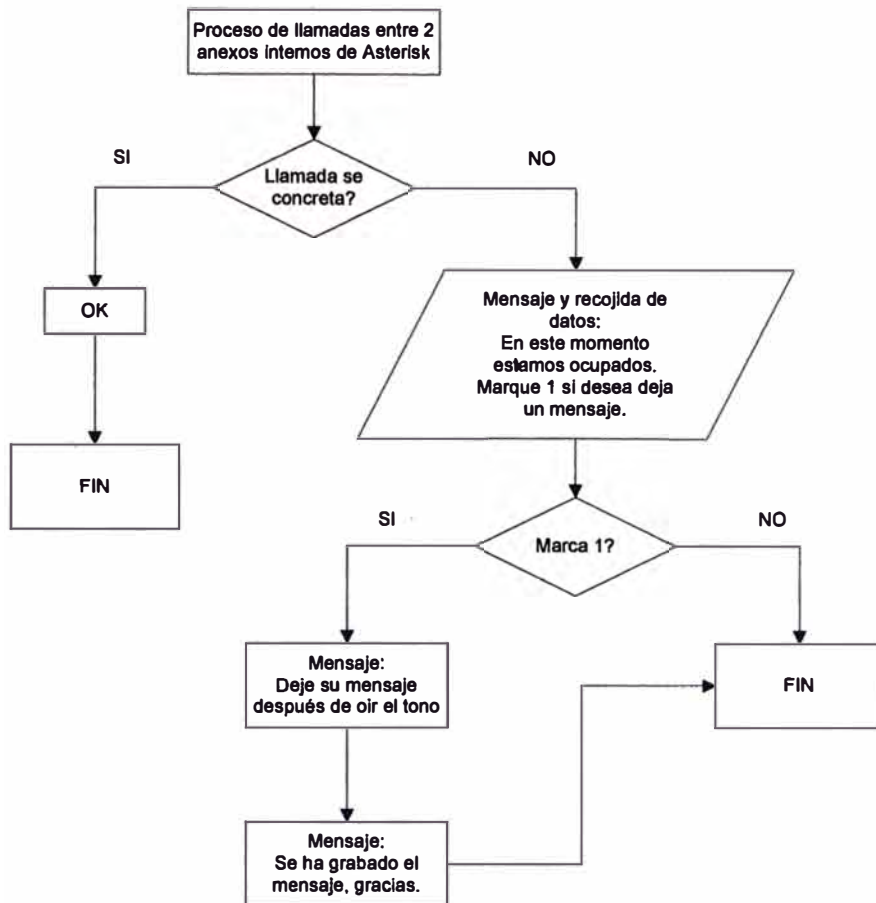
Las siguientes son consideraciones técnicas que se deben tener en cuneta para la implementación de la solución VoIP con Asterisk.

- El cableado de red que se usará para la implementación será CAT6 debido a las exigencias actuales y considerando que, bajo este estándar las distancias de las estaciones finales hacia los equipos de red no significan una preocupación.
- Las PCs que se destinarán para el trabajo de los agentes, deben cumplir un mínimo de requerimientos para poder soportar las aplicaciones Softphone y las aplicaciones CTI que son las que finalmente implican el mayor gasto en recursos. La recomendación es que como mínimo estas PCs tengan las Sgtes Características: Pentium4 2.8GHz 1GB Ram y disco duro de 40GB.
- Debido a que el presente proyecto se centra mas en el análisis de Asterisk en sí, no se entra en detalle de la electrónica de red usada para el despliegue de la implementación, sin embargo es necesario que se tenga en cuenta que se van a separar la red administrativa de la red de puestos de Operación además de considerar una Red DMZ para la mantener la seguridad de la RED. Para tal

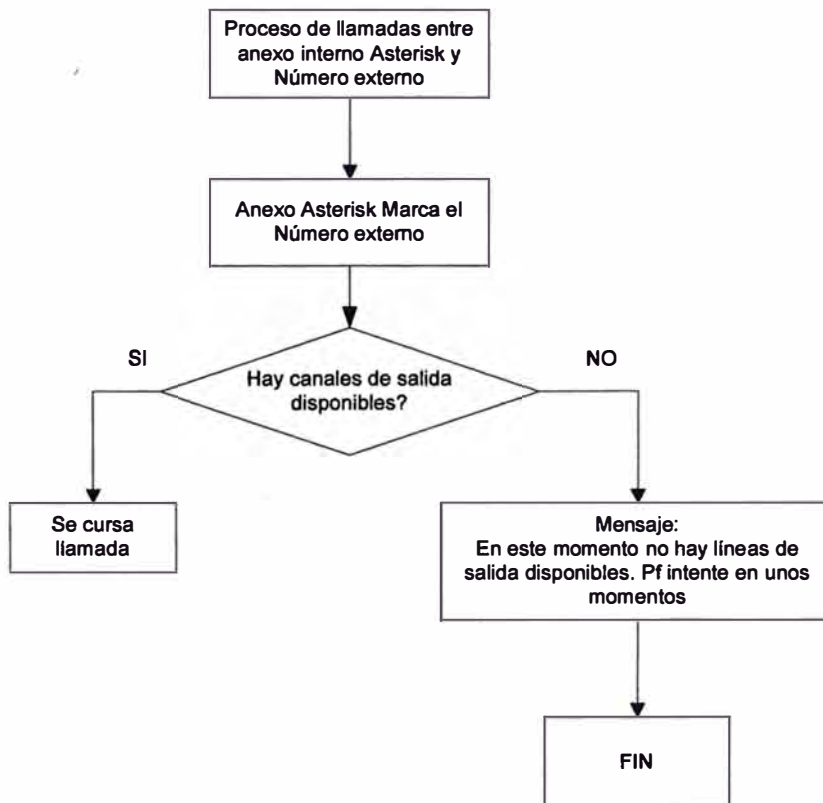
- efecto, se usarán 3 equipos switch, un equipo router y un Firewall. Adicionalmente, se asignará direcciones IP fijas a las estaciones de trabajo.
- Las PCs necesitarán consultar páginas de Internet, por lo tanto se les brindará salida a través de un Servidor Proxy. El Servicio de Internet será un ADSL de 2MB. Para la implementación del servidor proxy se dispondrá de una PC adicional con las mismas características que una PC de atención.
  - Incluiremos un Servidor de BD para las consultas y almacenamiento de datos de los clientes. Las características de este servidor son: Server HP Proliant Dual Core 2.0GHz, 1GB Ram y 80GB de disco duro.
  - Para los servicios de telefonía pública (Primarios y líneas convencionales) se ha escogido un mismo proveedor (Telefónica). Por la escala de la implementación, la contratación de servicios sería por 2 accesos Primarios RDSI y una línea análoga convencional para la conexión de un teléfono que será usado en caso de caída o problemas con la Central Asterisk (Es recomendable que para toda implementación se considere tener un teléfono que no dependa de la PBX).
  - Los puestos del CallCenter trabajarán con Softphones salvo las posiciones de supervisión (2) a quienes se les asignará 2 teléfonos IP Linksys del modelo SPA-841.
  - El hardware de telefonía seleccionado para la conexión de los primarios es, tarjeta DIGIUM E1 digital TE220B (2 puertos E1) para la conexión de los 2 primarios y además una tarjeta TDM410P Análoga con soporte de hasta 4 módulos FXO/FXS. Por el momento solo tomaremos en cuenta 2 módulos FXS para el par de anexos analógicos administrativos del Call Center.
  - Como parte de la electrónica de Red se ha dispuesto contar con lo siguiente, 2 switches de 24 puertos modelo BL2024 marca 3Com para soportar los 30 puestos de operación y los puestos administrativos. Además de 1 switch S3-4200 3Com de 24 puertos para la conexión de los servidores y un Router Cisco 2611.
  - Finalmente, Como servidor para la PBX Asterisk se ha seleccionado un Servidor HP Proliant Xeon Dual Core 2.4GHz con 2GB de Ram y disco duro de 40GB.

#### **4.2. Análisis de la Configuración de la PBX Asterisk para la atención**

Para la Plataforma de atención se debe tener en cuenta los diagramas en las figuras 4.1, 4.2 y 4.3 donde se detalla los diagramas de estado para la atención de llamadas.



**Fig. 4.1 Diagrama de flujo para llamadas internas**



**Fig. 4.2 Diagrama de flujo para llamadas de salida**



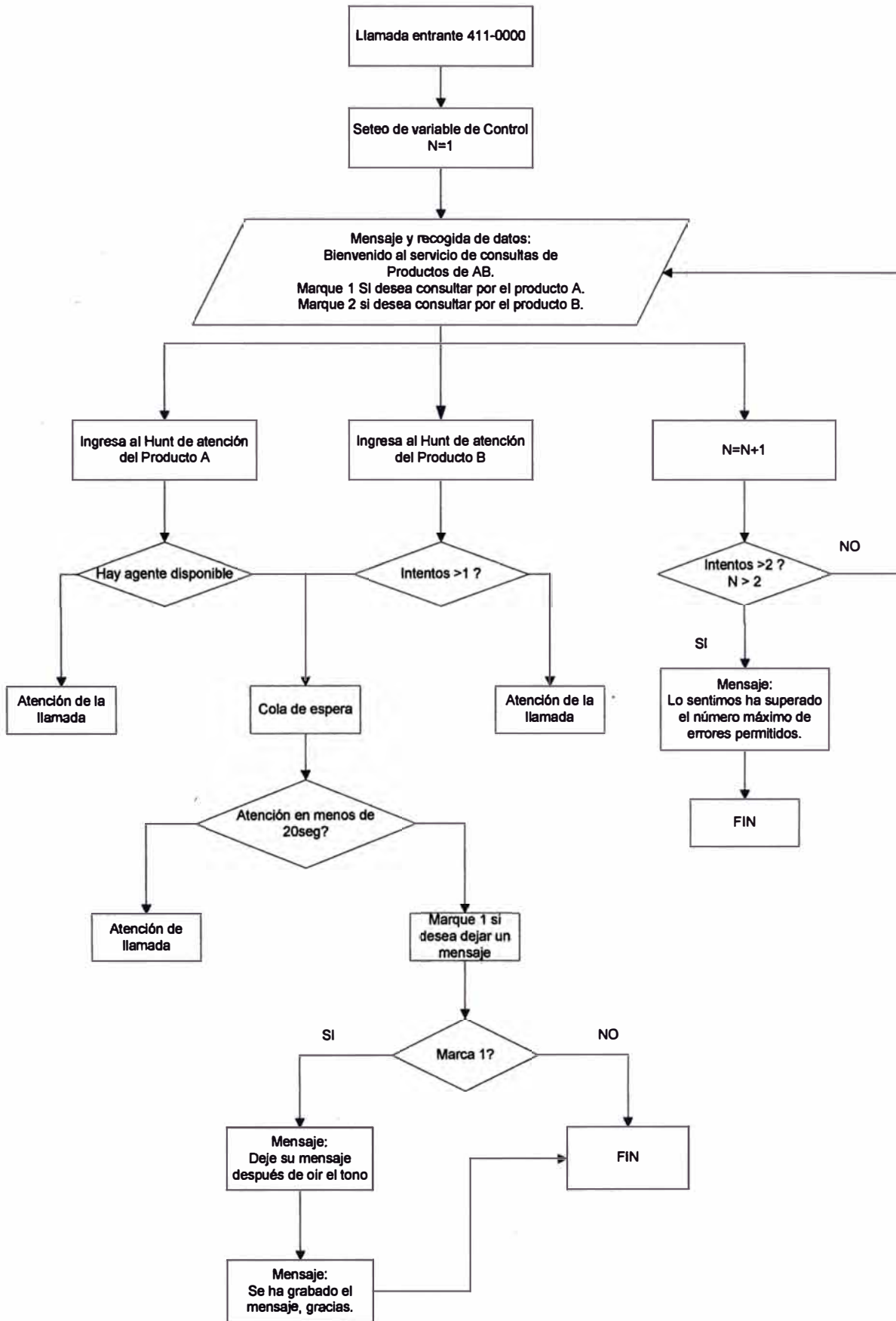


Fig. 4.3 Diagrama de flujo para llamadas de entrada

En Resumen, las necesidades para la implementación del Call Center del caso de estudio se muestran a continuación:

**Tabla 4.1 Resúmenes de los Requerimientos**

Item	Consideración diseño	Diseño Proyectoado	Consideraciones de Hardware	Consideraciones de Software
<b>PBX</b>	PBX Asterisk	Call Center	Servidor 2.4Ghz Dual 2GB RAM	Asterisk versión 1.4.4 Sobre Linux Red Hat Enterprise
<b>Gestión IN</b>	17 canales	1 Primario RDSI	Tarjeta 2 puertos E1 Digital	Drivers fabricante
<b>Gestión OUT</b>	15 canales	1 Primario RDSI		
<b>IVR</b>	SI	IVR en el número de acceso	Ninguno	Programación Asterisk
<b>buzón de Voz</b>	SI	Para gestión IN cuando no hay agente disponible	160 GB (Como mínimo para el almacenamiento de las grabaciones)	
<b>Grabación de llamadas</b>	SI	A demanda		
<b>Anexos administrativos</b>	SI	2 teléfonos	Tarjeta 2 puertos FXS Analógicos	
<b>Teléfonos IP</b>	SI	2 teléfonos	2 Teléfonos IP Linksys	
<b>PCs</b>	33	30 PCs de atención, 2 administrativas y un Proxy	Pentium 4, 2.8GHz, 1GB RAM, 40GB	Licenciamiento de Software y S.O.
<b>Headsets</b>	40			
<b>Switch</b>	3	3 switches 3COM	24 puertos c/u	
<b>Router</b>	1	1 router Cisco		

### 4.3. Presentación de la solución

Con base a los requerimientos y las proyecciones de Hardware y Software realizados, se presenta la siguiente solución de diseño para la implementación:

#### 4.3.1. Esquema de la Red de datos

Del análisis de los requerimientos, se ha establecido los siguientes valores y parámetros para el esquema de Red asociado:

Red de Operadores en el segmento 10.20.1.0/24

- Red de Servidores 10.20.2.0/24
- Direcciones para Gateways 10.20.1.1/24 y 10.20.2.1/24
- Para la Red DMZ se mantendrá el estándar del equipo Modem-router ADSL del proveedor, es decir 192.168.1/24.
- Se considera que para todo requerimiento hacia Internet se debe pasar a través del Proxy.

El diseño de la red de datos contempla el uso de Softphones, ya que el uso de Teléfonos incidiría en la separación de las redes de voz y datos en el diseño-

El esquema de la Red de datos se muestra en la Fig. 4.4

#### ESQUEMA DE RED PARA LA IMPLEMENTACIÓN DEL CALL CENTER

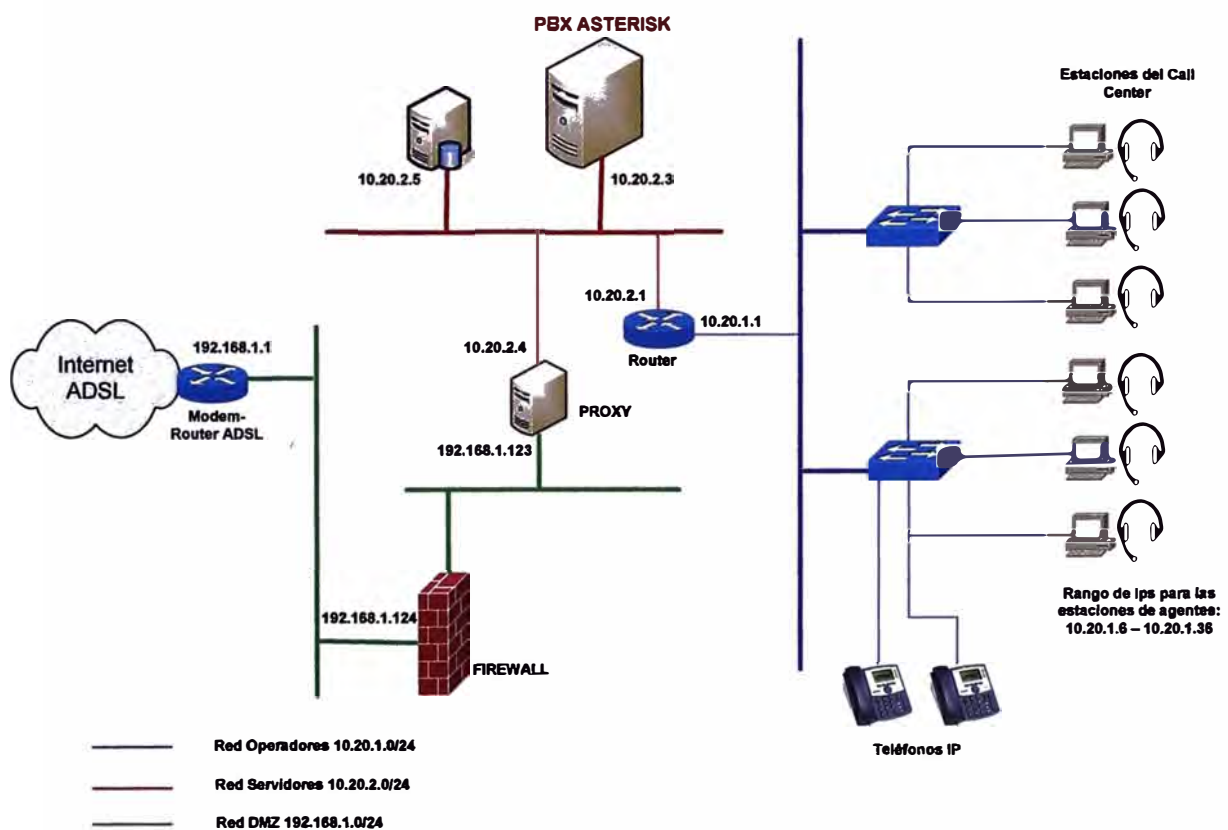


Fig. 4.4 Esquema de Red de datos

#### 4.3.2. Esquema de la Red de voz

El esquema de la red de voz consiste básicamente en las conexiones del servidor Asterisk con la PSTN tal como se muestra en la Fig. 4.5

## ESQUEMA DE VOZ PARA LA IMPLEMENTACIÓN DEL CALL CENTER

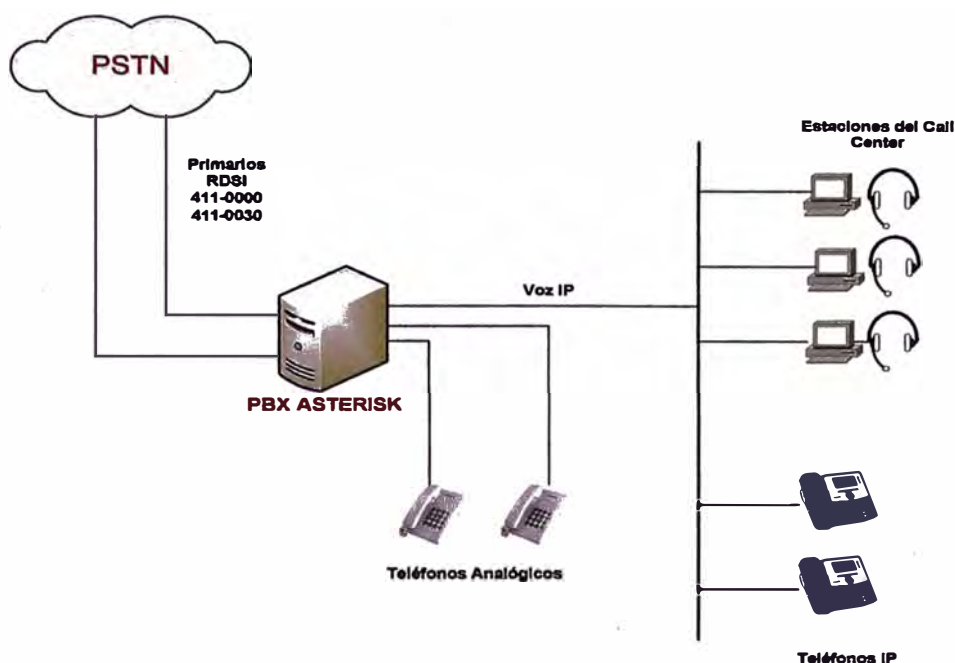


Fig. 4.5 Esquema de Red de Voz

### 4.3.3. Software y Programación

La instalación de Asterisk se efectúa como se indicó en la respectiva sección de este trabajo, utilizando el sistema operativo Linux distribución Debian (kernel 2.6), así como también los archivos de configuración `extensions.conf` y `sip.conf`, como se detalla más adelante, para manejar los internos y el IVR.

La configuración de los clientes (extensiones SIP en los puestos de atención), ya ha sido consignada previamente y se deberá proceder acorde a la misma. En la configuración propuesta se han estructurado los internos y privilegios para los mismos de la siguiente manera:

Quince internos de ventas (gestión Outbound), quince internos para gestión Inbound, dos de administración (0800), dos extensiones analógicas y una línea convencional que no depende de la PBX Asterisk.

Todos los anexos tienen acceso directo a la línea de salida presionando el número 9. Es decir, pueden efectuar cualquier llamada.

Los internos comienzan en 8000 y terminan en 8036.

Todos los internos pueden comunicarse mutuamente simplemente discando el número correspondiente.

Todos los internos tienen disponibles sus respectivas casillas de voicemail para dejar mensajes en caso de no estar disponibles. Para poder acceder a las casillas de

voicemail se ha configurado el interno 8080. Se debe ingresar el número de interno y contraseña (1234 por defecto) para autenticarse.

A continuación exponemos los archivos, en cada línea se comenta la utilidad de cada comando.

Primero Configuramos las extensiones para los puestos de atención en el archivo `/etc/asterisk/sip.conf`

### **sip.conf**

**[general]**

;

context=default

srvlookup=yes

;

;INTERNOS: 30 anexos de agentes, 2 administrativos,

;

**[8000]**

;

;Interno de operadora

;

type=friend

secret=operadora

context=internal ; El contexto internal controla lo que se puede hacer

nat=no ; Este teléfono no está detrás de un NAT

host=dynamic ; Este dispositivo se registra con nosotros

qualify=yes ; par calificado si no está a mas de 2000ms de distancia

canreinvite=no ; Asterisk por defecto trata de redireccionar

;

;----- Definimos todos los anexos

;

**[8001]**

;

type=friend

secret=8001

context=internal

nat=no

host=dynamic

```

qualify=yes
canreinvite=no

```

```
;
```

```
[8002]
```

```
;
```

```

type=friend
secret=8002
context=internal
nat=no
host=dynamic
qualify=yes
canreinvite=no

```

; Se continúa hasta el anexo 8030 para completar los 30 anexos para las estaciones  
;de agentes.

```
;
```

```
[8030]
```

```
;
```

```

type=friend
secret=8030
context=internal
nat=no
host=dynamic
qualify=yes
canreinvite=no

```

```
;
```

;----Definimos los anexos administrativos con propiedades gerenciales

```
;
```

```
[8050]
```

```
;
```

```

type=friend
secret=8050
context=internalgerencia ; el contexto de gerencia tiene mayores privilegios
nat=no
host=dynamic
qualify=yes
canreinvite=no

```

```
;
```

```
[8051]
;
type=friend
secret=8010
context=internalgerencia
nat=no
host=dynamic
qualify=yes
canreinvite=no
;
```

Con respecto a la creación de los agentes para la atención de las llamadas, modificamos el archivo `/etc/asterisk/agents.conf`

### **agents.conf**

```
[agents]
; A continuación definimos todos los agents del Call
agent => 1001,1234,Jose
agent => 1002,1234,Wilson
agent => 1003,1234,Juan
agent => 1030,1234,Jose
```

Procedemos a configurar la forma en el que el ACD repartirá las llamadas a los agentes, esto lo hacemos modificando el archivo `queues.conf`

### **queues.conf**

```
[ColaCliente1] Definimos la cola COLAENTRADA1 que será la de la opción 1
;(Producto A)
music=default
strategy=leastrecent ; Esta estrategia redirige la llamada a la interfaz que fue menos
;recientemente llamada
timeout=15 ; Tiempo de espera
wrapuptime=0
monitor-format=gsm
monitor-join = yes
; A continuación seteamos los agentes que son miembros de la cola
member => Agent/1001,1
member => Agent/1002,1
```



```

member => Agent/1010,1 ; debemos cubrir los 10 agentes que atenderán esta cola,
;desde el 1001 hasta el 1010
;
[ColaCliente2] ; Definimos la cola COLAENTRADA2 que será la de la opción 1
;(Producto A)
music=default
strategy=leastrecent ; Esta estrategia redirige la llamada a la interfaz que fue menos
;recientemente llamada
timeout=15 ; Tiempo de espera
wrapuptime=0
monitor-format=gsm
monitor-join = yes
; A continuación seteamos los agentes que son miembros de la cola
member => Agent/1011,1
member => Agent/1012,1
member => Agent/1015,1 ; Debemos cubrir los 5 agentes que atenderán esta cola,
desde el 1011 hasta el 1015

```

Procedemos a configurar el Dial Plan (IVR y tratamiento para las llamadas), para esto modificamos el archivo extensions.conf

### **extensions.conf**

```

[globals]
;variables globales
NUMEROMARCADO=0
NUMVOICEMAIL=8080
SIPOPERADORES=SIP/8001&SIP/8002&SIP/8003&SIP/8030; Se debe cubrir las 30
;extensiones
SIPADMINISTRACION=SIP/8050&SIP/8051; Se cubren las extensiones
administrativas
CANALESDESALIDA=ZAP/g2; Seteo de variable para indicar salida por Primario #2
;
[default]
;A Continuación comienzo la definición del conexto internal (Los anexos del Call usan
este contexto)
[internal]
;

```

```

include => checkvoicemail ;extension para acceder a los voicemail
include => internoainterno ;extension para manejar llamadas entre internos
;
exten => 9,1,GoTo(internoexternogerencia,s,1) ;si disca 9 va al contexto de llamada
;ext. para gerencia
exten => t,1,Playback(pbx-goodbye); time out general, despedida
exten => t,2,Hangup() ;fin de llamada
exten => i,1,Background(pbx-invalid);interno invalido, permite ingresar uno nuevo o
;cortar
exten => i,2,WaitExten(10);espera 10 segs a ver si ingresa un interno válido
exten => i,3,Playback(vm-gooby); locución de despedida
exten => i,4,Hangup();fin de llamada
;
[internalgerencia]
;
;Los internos administrativos usan este contexto
include => checkvoicemail
include => internoainterno
;
exten => 9,1,GoTo(internoexternogerencia,s,1) ;si disca 9 va al contexto de llamada
;ext. administrativa
exten => t,1,Playback(pbx-goodbye); time out general, locución de despedida
exten => t,2,Hangup() ;fin de llamada
;
exten => i,1,Background(pbx-invalid);interno invalido, permite ingresar uno nuevo o
;cortar
exten => i,2,WaitExten(10); espera 10 segs a ver si ingresa un interno válido
exten => i,3,Playback(vm-gooby); Locución de despedida
exten => i,4,Hangup();fin de llamada
;
[internoainterno]
;
exten => _80[1-5]X,1,SetVar(INTERNO=${EXTEN});inicializa la variable de canal
;INTERNO con el numero marcado
exten => _80[1-5]X,2,SetVar(IRVOICEMAIL=0); incializa la variable de canal
;IRVOICEMAIL a cero
exten => _80[1-5]X,3,Dial(SIP/INTERNO,10,r);permite llamar a internos

```

```

;
exten => _80[1-5]X,4,SetVar(IRVOICEMAIL=1);no atendió el dial después de 20
segundos, marca IRVOICEMAIL=1
exten => _80[1-5]X,5,Background(personal-nopodemosatenderlo); mensaje de
;internos no disponibles
exten => _80[1-5]X,6,WaitExten(3);espera para que ingrese un interno
exten => _80[1-5]X,7,WaitMusicOnHold(15);esperar 15 segundos con música ya que
;desea intentar llamar de nuevo
exten => _80[1-5]X,8,GoTo(3);va a la prioridad 3 donde llama nuevamente al interno
;INTERNO
;
exten => _80[1-5]X,104,SetVar(IRVOICEMAIL=1);no atendió porque esta ocupado o
no conectado
exten => _80[1-5]X,105,Background(personal-nopodemosatenderlo);mensaje de
;internos no disponibles
exten => _80[1-5]X,106,WaitExten(3);espera para que ingrese un interno nuevo
exten => _80[1-5]X,107,WaitMusicOnHold(15);si decide esperar mantiene 15seg con
música antes de llamar de nuevo
exten => _80[1-5]X,108,GoTo(3);va a la prioridad 3 donde llama nuevamente al interno
;INTERNO
;
exten => 1#,1,GoToIf(${IRVOICEMAIL} = 1)?2:10);ingresó 1# después de tratar de
;llamar a interno (IRVOICEMAIL=1).
exten => 1#,2,VoiceMail(${INTERNO}@default)
exten => 1#,3,Hangup();fin de llamada una vez que deja el mensaje
;
exten => 1#,10,Playback(vm-goodbye); ingresó 1# sin haber ingresado un interno
;previamente, despedida
exten => 1#,11,Hangup();fin de llamada
;
;A continuación Definimos el Contexto VoiceMail
[checkvoicemail]
;
exten => ${NUMVOICEMAIL},1,VoiceMailMain();permite verificar el voicemail llamando
;al interno NUMVOICEMAIL
;
; Definimos el comportamiento de las llamadas de Entrada

```

[externointerno]

;

exten => 4110000,1,Set(CALLERID(name)=Cliente1:\${CALLERID(num)});Almaceno  
;los datos de la llamada

exten => 4110000,2,Answer();contesta la llamada al Número de ingreso 411-0000

exten=>2020,2,Set(MONITOR\_FILENAME=/var/spool/asterisk/monitor/\${EXTEN}-  
STRFTIME(\${EPOCH},,%Y%m%d%H%M%S)}-\${UNIQUEID}-\${CALLERID(num)});

;Comienzo a grabar el audios de la llamada. La grabación se almacenará en la ruta  
;/var/spool/asterisk/monitor/ y el nombre del archivo generado tendrá el número de  
;extensión que recibirá la llamada así como también la fecha y hora de atención.

Exten => 4110000, ,Gotoif(Time(08:00-20:00|mon-fri),5,8); Procedemos a configurar el  
;horario de atención de 08:00 a 20:00.

exten => 4110000,3,Background(bienvenida-menu); reproduce el mensaje de  
;bienvenida y el menú de 2 opciones disponibles

exten => 4110000,4,WaitExten(4);espera 4 segundos para que ingrese un interno

;

;Si marcó alguna de las 2 opciones, entonces definimos la atención para cada una

exten => 1,1, Ringing; Comenzamos la definición si es que marcó la opción 1

exten => 1,2,SetMusicOnHold(default)

exten => 1,3,Queue(ColaCliente1|r); La atención la brindarán los agentes miembros de  
;la ColaCliente1 definida en el fichero queues.conf

exten => 1,4,GoTo(internoainterno,1#,2); Si supera el tiempo de encolamiento, se  
;envía al buzón de voz

exten => 1,5, Hangup

;

exten => 2,1, Ringing; Comenzamos la definición si es que marcó la opción 2

exten => 2,2,SetMusicOnHold(default)

exten => 2,3,Queue(ColaCliente1|r); La atención la brindarán los agentes miembros de  
;la ColaCliente2 definida en el fichero queues.conf

exten => 2,4,GoTo(internoainterno,1#,2); Si supera el tiempo de encolamiento, se  
;envía al buzón de voz

exten => 2,5, Hangup

;

;En caso de que no haya marcado una opción valida después de los 4 segundos

exten => 4110000,5,SetVar(NUMEROMARCADO=\${NUMEROMARCADO} + 1})

```

exten => 4110000,6,Gotoif(${NUMEROMARCADO}>1:4110000,10:4110000,4); Se
;envía a escuchar nuevamente el menú si no ingresa nada o marca opción no valida
;de lo contrario continúa con el tratamiento por opción
;
; A continuación definimos las características para las llamadas de salida
[internoexternonormal]
;
exten => s,1,Dial(${CANALESDSALIDA}/${NUMEROMARCADO:1});marca número
;ingresado saca 9 de adelante y usa CANALESDSALIDA que se definió como rango
;del primario de salida
exten => s,2,Hangup();fin de la llamada
exten => s,102,Playback(personal-nohaylineas);si no hay lineas disponibles reproduce
;un mensaje
exten => s,103,Hangup();fin de la llamada
exten => i,1,Hangup();maneja entradas inválidas
;
[internoexternogerencia]
;
exten => s,1,Dial(${CANALESDSALIDA}); le da tono de la PSTN, puede efectuar
;cualquier llamada
exten => s,2,Hangup();fin de la llamada
exten => s,102,Playback(personal-nohaylineas);reproduce mensaje si no hay lineas
;disponibles de salida
exten => s,103,Hangup();fin de la llamada
;
; Modificamos el context general
[general]
;
; Default formats for writing Voicemail
;format=g723sf|wav49|wav
format=wav49|gsm|wav
; Maximum length of a voicemail message in seconds
;maxmessage=180
; Minimum length of a voicemail message in seconds
;minmessage=3
; Maximum length of greetings in seconds
;maxgreet=60

```

; How many milliseconds to skip forward/back when rew/ff in message playback

skipms=3000

; How many seconds of silence before we end the recording

maxsilence=10

; Silence threshold (what we consider silence, the lower, the more sensitive)

silencethreshold=128

;

[default]

8001 => 8001 ,Interno Agente

8002 => 8002 ,Interno Agente

8032 => 8003 ,Interno Ventas; Cubrimos todos os anexos del Call Center

;

;Definimos las extensiones 8033 y 8034 que serán los anexos analógicos y por lo tanto

;deben invocar el puerto asignado para ellos en la tarjeta analógica (Ver archivo

;zapata.conf)

exten => 8033,1,Dial(Zap/64);

exten => 8034,1,Dial(Zap/65)

Para la configuración de los Primarios y canales analógicos, modificamos el archivo zapata.conf.

### **Zapata.conf**

[trunkgroups]

;

;A continuación se define la configuración del Primario #1

[channels]

context = default

group = 1

switchtype = euroisdn

signalling = pri\_cpe

channel => 1-15,17-31

;

;Ahora se define el primario #2

context = default

group = 2

switchtype = euroisdn

signalling = pri\_cpe

channel => 32-46,48-62

; Ahora definimos los canales 63, 64, 65 y 66 que son los 4 puertos de nuestra tarjeta analógica.

```
;context=default
;signalling=fxo_ks
;channel => 63
;
context=default
signalling=fxo_ks
channel => 64
;
context=default
signalling=fxo_ks
channel => 65
;
context=default
signalling=fxo_ks
channel => 66
```

#### **4.4. Presupuesto y tiempo de ejecución**

Es necesario tener en cuenta los siguientes aspectos económicos:

**Costo del Hardware**, esto incluye el costo de las PCs a desplegar así como también el costo del servidor Asterisk con tarjetas de telefonía, teléfonos IP y otros servidores (Proxy, Servidor de BD, etc).

Los precios que se muestran a continuación en las tablas 4.2, 4.3 y 4.4, son precios referenciales.



Tabla 4.2 Gasto para implementación de Hardware proyectado

	Características	Cantidad	Costo Unitario	Total
<b>PCs</b>	Pentium 4, 2.8GHz, 1GB RAM, 40GB	32	\$300.00	\$9,600.00
<b>PBX Asterisk</b>	Server HP Proliant Xeon Dual Core 2.0GHz, 2GB RAM	1	\$1,400.00	\$1,400.00
<b>Servidor BD</b>	Server HP Proliant Dual Core 1.8GHz, 2GB RAM	1	\$800.00	\$800.00
<b>Servidor Proxy</b>	Pentium 4, 2.8GHz, 1GB RAM, 40GB	1	\$300.00	\$300.00
<b>Tarjeta digital E1 (2 puertos) para Asterisk</b>	TE220B (echo Cancelation) Digium	1	\$1,200.00	\$1,200.00
<b>Tarjeta de conversión análoga</b>	TDM410P (2FXO, 2FXS) Digium	1	\$500.00	\$500.00
<b>Teléfonos IP (SIP)</b>	SPA-841 (Linksys)	2	\$80.00	\$160.00
				<b>\$13,960.00</b>

Costo de la implementación de Red, esto involucra los equipos switch, routers, patch panels, cableado.

Tabla 4.3 Gasto para implementación de diseño de Red

	Características	Cantidad	Costo Unitario	Total
<b>Switch de Borde (24 puertos)</b>	BL2024, 3Com	2	\$220.00	\$440.00
<b>Switch Core (24 puertos)</b>	S3-4200, 3Com	1	\$500.00	\$500.00
<b>Router</b>	Cisco 2611	1	\$1,200.00	\$1,200.00
<b>Cableado</b>	Puntos, cableado y Patch Panel	50	\$22.00	\$1,100.00
				<b>\$3,240.00</b>

Costo de servicios, esto involucra el costo de instalación y rentas mensuales de los accesos primarios, línea convencional y servicio de internet.

Tabla 4.4 Gasto por servicios de instalación

	Características	Cantidad	Costo Unitario	Total
Instalación Primario RDSI	Proveedor Telefónica	2	\$60.00	\$120.00
Instalación ADSL	Proveedor Telefónica	1	\$0.00	\$0.00
Instalación Línea Convencional	Proveedor Telefónica	1	\$15.00	\$15.00
				<b>\$135.00</b>

Tabla 4.5 Renta Mensual

	Características	Cantidad	Costo Unitario	Total
Renta Mensual Primario RDSI	Proveedor Telefónica	2	\$600.00	\$1,200.00
Renta Mensual ADSL	Proveedor Telefónica	1	\$300.00	\$300.00
Renta Mensual Línea Convencional	Proveedor Telefónica	1	\$40.00	\$40.00
				<b>\$1,540.00</b>

- Costo de Software que es casi en su totalidad el costo de licenciamiento para el sistema operativo de las PCs, Base de datos y otras aplicaciones que en este caso usarán Windows 2000. El promedio de gasto para este licenciamiento es aproximadamente \$4000.00.

Adicionalmente, el plazo de implementación para este proyecto está ligado en su mayor parte al tiempo proyectado para la realización del cableado. Se proyecta que al tratarse de unos 40 puntos de Red a ser distribuidos en un ambiente diseñado para Call Center, el tiempo estimado puede ser de entre 4 a 6 días. Por la parte de implementación de la misma PBX Asterisk, este no debería tardar mas de 3 días a partir de los inicios de la configuración.

## **CONCLUSIONES**

La solución Asterisk presentada en este trabajo se enfoca en una implementación para Call Center, sin embargo, es valido para cualquier empresa que no necesariamente este en el rubro de los Call Center ya que cualquier otro rubro diferente no tendrá las mismas exigencias que requiere un Call Center.

Es Claro que Asterisk no hubiera sido desarrollado si no fuera por la Telefonía IP. Prácticamente todas las aplicaciones de Asterisk se basan en el uso de la Voz sobre IP e incluso permite la integración con módulos analógicos y digitales sin mucho esfuerzo. Esto debe ser una de las principales ventajas de Asterisk

A diferencia del uso de una PBX de fabricante, Asterisk tiene código abierto, es decir, se puede programar y ajustar a las necesidades que uno tiene. Esto significa una gran ventaja. Por ejemplo, el Dial Plan en Asterisk se puede modificar a gusto con un entendimiento sencillo de su lenguaje de programación, esto no sucede en las Centrales telefónicas propietarias.

Aunque no se ha tratado el caso, son muy comunes las implementaciones donde se tiene que integrar Asterisk con otras PBX, ya sea con otro Asterisk o con cualquier otra Central. La interconexión se presenta como la interconexión de cualquier otras dos Centrales pero cuando se interconecta un Asterisk con otro Asterisk, es posible usar el protocolo IAX que permite un mejor manejo del ancho de banda que usando SIP o H.323.

Durante la instalación y configuración de Asterisk para realizar el presente diseño surgieron algunas apreciaciones en cuanto a este sistema como una solución de comunicaciones; diseñar una PBX Asterisk no resulta una tarea trivial y lleva bastante tiempo internalizar los conceptos referentes al funcionamiento, manejo de los dial plan, archivos de configuración y demás. Aunque una vez que esto se ha logrado, queda confirmada la versatilidad y el dinamismo que imprime a las soluciones de

comunicación posibles, permitiendo que el desarrollador pueda conjugar sus conocimientos y creatividad para atacar los problemas y requerimientos de una empresa.

Desde el punto de vista de un emprendimiento de telecomunicaciones, los costos de una implementación Asterisk respecto de una central tradicional caen en dependencia directa respecto de las aplicaciones que la empresa requiera: Asterisk realmente se convierte en una solución rentable cuando la empresa explota sus diversas posibilidades de base de datos, interconexión de PBX, manejo de CDR, IVR totalmente personalizable, uso de comunicaciones VoIP, entre otros.

Un tema a tener en cuenta es el hecho de haber seleccionado el uso de softphones en los puestos de operación. Esto significa un ahorro considerable a usar teléfonos donde el hecho mismo de no contar con switches PoE (Power over Ethernet) obligaría a la adquisición de teléfonos con la característica de tener fuentes de alimentación eléctrica. Lógicamente esto implicaría que en los puestos de atención se debe contar con una toma de energía adicional para esto. Además está el hecho de que con el uso de Softphones se puede hacer uso de paquetes Asterisk para la integración CTI.

**ANEXO A**  
**HOJA DE ESPECIFICACIONES TARJETAS DIGIUM E1**

#### **4 Port T1/E1 PRI**

4 Digital Interface Ports Support both Voice and Data

Selectable T1, E1 or J1 Mode

Half-Length, Full-height Card

TE405P for use only with a 5.0V PCI 2.2 compliant slot.

- Bundled with VPMOCT128 Echo Cancellation Module as TE407P

TE410P for use only with a 3.3V PCI 2.2 compliant slot.

- Bundled with VPMOCT128 Echo Cancellation Module as TE412P

TE420 for use only with a PCI-Express 1.0 compliant slot.

- Bundled with VPMOCT128 Echo Cancellation Module as TE420B

#### **2 Port T1/E1 PRI**

2 Digital Interface Ports Support both Voice and Data

Selectable T1, E1, or J1 Mode

Half-length Digital Card

TE205P for use only with a 5.0V PCI 2.2 compliant slot.

- Bundled with VPMOCT64 Echo Cancellation Module as TE207P

TE210P for use only with a 3.3V PCI 2.2 compliant slot.

- Bundled with VPMOCT64 Echo Cancellation Module as TE212P

TE220 for use only with a PCI-Express 1.0 compliant slot.

- Bundled with VPMOCT64 Echo Cancellation Module as TE220B

#### **1 Port T1/E1 PRI**

1 Digital Interface Port Supports both Voice and Data

Selectable T1, E1, or J1 Mode

Half-length, Low Profile, Digital Card

TE122 for use only with a 3.3V or 5.0V PCI 2.2 compliant slot.

- Bundled with VPMOCT032 Echo Cancellation Module as TE122B

TE121 for use only with PCI-Express 1.0 compliant slot.

- Bundled with VPMOCT032 Echo Cancellation Module as TE121B

#### **Target Applications**

Legacy PBX/IVR Services

Voice-over Internet Protocol (VoIP) services

Complex IVR Trees

“Meet-Me” Bridge Conferencing

VoIP Gateways (supports SIP, H.323 and IAX)

**Calling Card Platforms**

Voice/Data Router (replace expensive routers)

PRI Switch Compatibility – Network or CPE

**Data Modes**

SyncPPP (both Fixed and Dialup)

Frame Relay

Cisco HDLC

Multi-link PPP

**PRI Switch Compatibility**

EurolSDN (PRI or PRA) — Q.931/Q.921

AT&T 4ESS

DMS 100

Lucent 5E

Network or CPE

National ISDN 2

CAS Voice Modes

**CAS Voice Modes**

Feature Group D

E&M Wink

a-Law,  $\mu$ -Law, and Linear Modes Supported



**ANEXO B**  
**HOJA DE ESPECIFICACIONES TARJETAS DIGIUM 410 ANALÓGICAS**

**Feature List**

Caller ID and Call Waiting Caller ID

ADSI Telephones

PCI Half-length Slot

RJ-11C Connector

**Technical Specifications**

500-Mhz Pentium III or better with 64MB RAM

Must have available PCI Slot

Operation Range: 0° to 50°C, 32° to 122° F

Physical Specifications

Operation Range: 0° to 50°C, 32° to 122° F

Storage Range: -20° to 65°C, 4° to 149° F

Humidity: 10-90% non-condensing

**Product Applications Scenarios**

Small Office Home Office (SOHO) applications

Gateway Termination to Analog Telephones

Add Inexpensive Analog Phones to Existing PBXs

Wireless Point-to-Point Applications between Asterisk Servers

## **BIBLIOGRAFÍA**

- [1] Scott Keagy.2002 – Integración de Redes de Voz y Datos – Cisco systems.
- [2] Flavio E. Gonzalves .2007 – Asterisk PBX Guía de Configuración – Tercera Generación.
- [3] Gagliardi, Alejo.2006 -Asterisk configuración e implementación- NexIT Specialist no22.
- [4] García, Rafael Achaerandio.2005 -Oportunidades de negocio y nuevos servicios y aplicaciones en las redes del futuro- IDC España.
- [5] IETF RFC 1889.2000 -RTP: A transport protocol for real-time applications.
- [6] IETF RFC 1890.2000 -RTP: profile for audio and video conferences with minimal control.
- [7] IETF RFC 2327.2001 -SDP: Session Description Protocol.
- [8] Mata, Iago Soto.2005 -Soluciones VoIP en Linux - Adianta ([www.adianta.net](http://www.adianta.net)).
- [9] Meggelen, Jim Van.2004 -Asterisk documentation project Volumen One: An introduction to Asterisk- The Asterisk Documentation Project.
- [10] Sitio oficial Asterisk.2008 - <http://www.asterisk.org>.
- [11] Spenser, Marck.2005 -Introduction to the Asterisk Open Source PBX- Digium.
- [12] Tamusiunas, Fabricio.2005 -Asterisk, configuracion y gerenciamiento- Nic.br ([www.nic.br](http://www.nic.br)).