



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Una aproximación a la Telefonía 2.0: Asterisk™

Manuel Camargo Lominchar

14 de junio de 2012



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS

Una aproximación a la Telefonía 2.0: Asterisk™

- Departamento: Lenguajes y sistemas informáticos
- Autor del proyecto: Manuel Camargo Lominchar
- Director del proyecto: Manuel Palomo Duarte
- Director del proyecto: Juan Manuel Dodero Beardo

Cádiz, 14 de junio de 2012

Fdo: Manuel Camargo Lominchar

Agradecimientos

Sin ayuda de las siguientes personas este Proyecto no hubiera podido ser realizado, por ello doy los siguientes agradecimientos...

...a mi madre, por haberme enseñado la principal habilidad para que esto se haya podido materializar, la lectura.

...a mi padre, por haberme mostrado el primer principio básico de la telefonía, el habla.

...a Lucía por haber potenciado el segundo y último principio fundamental, la escucha.

...a Sergio Serrano, Rosa Atienza y Elio Rojano, por haberme introducido en la curiosa senda de la que trata este Proyecto.

...a todos los componentes de mi equipo privado de testing, destacando a Alejandro Díaz, por la parte que les corresponde.

...y a Manuel Palomo, por haberme ofrecido el camino más interesante para plasmar mi idea en forma de Proyecto.

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright © 2012 Manuel Camargo Lominchar.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Notación y formato

Aquí incluiremos los aspectos relevantes a la notación y el formato a lo largo del documento. Para simplificar podemos generar comandos nuevos que nos ayuden a ello, ver `comandos.sty` para más información.

Cuando nos refiramos a un programa en concreto, utilizaremos la notación:

asterisk.

Cuando nos refiramos a un comando, o función de un lenguaje, usaremos la notación:

`quicksort`.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Objetivos	1
1.3. Motivación	2
1.4. Estructura de la Memoria	3
1.5. Conceptos Generales	4
1.5.1. ¿Que es una PBX?	4
1.5.2. ¿Que es Asterisk?	4
1.5.3. ¿Que es Telefonía IP?	4
1.5.4. ¿Que es la Telefonía 2.0?	5
2. Planificación y Planteamiento	7
2.1. Planteamiento	7
2.1.1. Método del Caso	7
2.2. Planificación	8
2.2.1. Aprendizaje e Investigación	8
2.2.2. Desarrollo e Innovación	9
2.2.3. Fases de Creación	10
3. Documentación Teórica	13
3.1. Sistema MediaWiki	13
3.1.1. Origen de la WIKI	13
3.2. Estructura General	14
3.2.1. Fundamentos	15
3.2.2. Plan de Marcación	15
3.2.3. Protocolos e Interfaces	16
3.2.4. Sistema de Módulos	18
3.2.5. Conceptos Avanzados	19
3.2.6. Bibliografía y Enlaces de Interés	21
3.3. Estructura de los Artículos	22
3.3.1. Consideraciones de Formato y Estructura	24

4. Desarrollo del proyecto	25
4.1. Elección del Servidor	25
4.1.1. Marcas Generalistas	25
4.1.2. Consideraciones Técnicas	25
4.1.3. Servidor de Pruebas del Project	26
4.2. Elección e instalación del Sistema Operativo en el Servidor	26
4.2.1. Valorando Distribuciones Linux	26
4.2.2. La selección personal: Ubuntu Server. Motivos Generales	27
4.2.3. Primeros Pasos en el Servidor	28
4.3. Virtualización para Asterisk	29
4.3.1. Valorando Opciones de Virtualizacion	29
4.3.2. Ventajas de la virtualización con KVM	30
4.4. Instalación de la Máquina Virtual	30
5. Caso de Estudio: UCA Autos	35
5.1. Introducción	35
5.1.1. Instalación del sistema Asterisk	37
5.1.2. Comprobación: Instalación del sistema Asterisk	38
5.1.3. Configurando equipos y el entorno SIP	39
5.1.4. Comprobación: Configurando equipos y el entorno SIP	41
5.1.5. Configuración del Primer Plan de Marcación	42
5.1.6. Comprobación: Configuración del Primer Plan de Marcación	44
5.1.7. Introduciendo al proveedor de Telefonía	45
5.1.8. Comprobación: Introduciendo al proveedor de Telefonía	46
5.2. Mejorando lo presente	48
5.2.1. Preparando las Llamadas Salientes	50
5.2.2. Resolución:Preparando las Llamadas Salientes	55
5.3. Interconectando máquinas Asterisk	57
5.3.1. Acceso al protocolo IAX	58
5.3.2. Resolución: Acceso al protocolo IAX	60
5.4. Distribuidor de Piezas de Repuesto	61
5.4.1. Colas en Asterisk	62
5.4.2. Resolución: Colas en Asterisk	65
5.5. Nuevas Incorporaciones	66
5.5.1. Generación Automática de Eventos	67

5.5.2.	Resolución: Generación Automática de Eventos	68
5.5.3.	Probando nuestro CTI, Asterisk Manager Interface	69
5.5.4.	Comprobación: Probando nuestro CTI, Asterisk Manager Interface	72
5.5.5.	Devolviendo el Control al Cliente	73
5.5.6.	Resolución: Devolviendo el Control al Cliente	82
5.5.7.	Análisis de Asterisk con Lupa	83
5.5.8.	Resolución: Análisis de Asterisk con Lupa	84
5.5.9.	Conferencias Corporativas	85
5.5.10.	Resolución: Conferencias Corporativas	91
5.6.	Bienvenidos a la Telefonía 2.0	93
5.6.1.	Respuesta de Voz Interactiva	94
5.6.2.	Respuesta de Voz Interactiva	99
6.	Conclusiones y Trabajo Futuro	101
6.1.	Conclusiones	101
6.1.1.	Sobre WIKIAsterisk	101
6.1.2.	Método del Caso y Caso de Estudio	101
6.1.3.	Máquina Virtual	102
6.2.	Dificultades Técnicas Superadas	102
6.3.	Planteamientos Futuros	103
	Software utilizado	105
	Difusión de WIKIAsterisk	109
	Anexo I: Código Referenciado	113
	Bibliografía y referencias	134
	GNU Free Documentation License	139
1.	APPLICABILITY AND DEFINITIONS	139
2.	VERBATIM COPYING	140
3.	COPYING IN QUANTITY	141
4.	MODIFICATIONS	141
5.	COMBINING DOCUMENTS	142
6.	COLLECTIONS OF DOCUMENTS	143
7.	AGGREGATION WITH INDEPENDENT WORKS	143
8.	TRANSLATION	143
9.	TERMINATION	144
10.	FUTURE REVISIONS OF THIS LICENSE	144
11.	RELICENSING	144
	ADDENDUM: How to use this License for your documents	145

Indice de figuras

2.1. Esquema general de una sesión siguiendo el Método del Caso	8
2.2. Planificación general del Proyecto	9
2.3. Planificación del Desarrollo del Proyecto	10
2.4. Fases del Desarrollo del Proyecto	11
3.1. Estructura general de WIKIAsterisk	14
3.2. Esquema de la sección Introductoria de la WIKI	15
3.3. Esquema de la sección del Plan de Marcación de la WIKI	17
3.4. Esquema de la sección de los Protocolos e Interfaces de la WIKI	18
3.5. Esquema de la sección de Módulos Asterisk	20
3.6. Esquema de la sección de Aspectos Avanzados del Sistema	22
3.7. Esquema de la sección de Bibliografía y Enlaces de Interés	23
3.8. Indicador de Artículo pendiente de Completar o Mejorar	24
3.9. Indicador de Artículo pendiente Formatear	24
4.1. Capacidad inicial del disco duro en la Maquina Virtual	33
5.1. Capacidad del disco duro tras la instalación completa de Asterisk	38
5.2. Estructura Inicial de Extensiones de Usuario	39
5.3. Configuración de una Extensión con el Programa X-Lite v4.1	42
5.4. Ejemplo de una llamada en formato CLI	44
5.5. Flujo de Llamada de la Operadora Automática	52
5.6. Ejemplo de salida y entrada a través de un Proveedor IP	55
5.7. Ejemplo de salida sin retorno a través de un Proveedor IP	56
5.8. Ejemplo de salida de un mensaje en el Buzón de Voz	57
5.9. Captura de la Pantalla de Disponibilidad de Agentes de Asternic	64
5.10. Ejemplo de Estadísticas Asternic Stats	65
5.11. Llamada Entrante de alerta	69
5.12. Configuración de la extensión en Vtiger	70

5.13. Configuración de los parámetros AMI en Vtiger	71
5.14. Llamada Entrante en Vtiger	73
5.15. Llamada Saliente en Vtiger	73
5.16. Gráfico Circular de Llamadas Totales	85
5.17. Histograma de Llamadas por días	86
5.18. Lista de Conferencias Dinámicas Web-MeetMe	91
5.19. Diagrama de una llamada en el sistema a IVR	98
6.1. Captura del Software T _E Xnic Center	106
6.2. Captura del Softphone Zoiper	107
6.3. Captura de la Plataforma MinGW ejecutando comandos GIT	108
6.4. Estadísticas de acceso del Blog 10000horas	109
6.5. Estadísticas WIKIAsterisk con Firestats	110
6.6. Logotipo de la Certificación dCAP	111

Capítulo 1

Introducción

1.1. Contexto

Desde los inicios de la telefonía a finales del siglo XIX, el mercado ha dictado el orden en el proceso de creación y desarrollo, en un mundo sin estándares, donde las compañías emergentes impusieron su visión, establecieron el control y la oferta de servicios estaba limitada a lo que las mismas podían ofrecer exclusivamente.

Justamente un siglo después, surgió el primer movimiento para empezar a crear un camino común pero a su vez, más flexible y abierto a la demanda general impuesta por la nueva forma de concebir el mundo de la comunicación a distancia. Su nombre fue Asterisk. En la última década ha ido expandiéndose por todo el mundo, pero solo en los últimos 3 años ha sufrido realmente un importante auge. Tanto los principales proveedores de telefonía, así como las compañías líderes en el sector de las telecomunicaciones han sabido adaptarse al cambio y empiezan a adoptar este sistema con vistas puestas al futuro.

1.2. Objetivos

Aunque está visto que cada día cobra mayor relevancia, la comunidad de Asterisk en Español, ha permanecido circunscrita al contenido y desarrollo de las fuentes que provienen del inglés. Con este proyecto pretendo sentar las bases de Asterisk en nuestro idioma, abarcándolas de la siguiente forma:

1. Para la documentación Teórica, la creación de una WIKI exclusiva de Asterisk, basándome en sus principales fuentes, en mi propio conocimiento y experiencia con el sistema.
2. Desarrollar un Caso Práctico que cubra el máximo de la teoría, basado en un sistema Asterisk, siguiendo el Método del Caso y reflejarlo en la memoria
3. Demostrar el caso realmente, utilizando una Máquina Virtual que se adjuntara al resto del proyecto.
4. Utilizar herramientas de Software Libre como es Asterisk para preservar su filosofía de uso y distribución.

Este proyecto pretende cubrir los siguientes aspectos sobre el sistema Asterisk y lo directamente relacionado al mismo:

- Estado del arte y conceptos fundamentales
- Arquitectura de Asterisk
- Principales Protocolos de comunicación: SIP, IAX,...

- Interfaces con la telefonía clásica.
- Despliegue del Plan de Marcación, Asterisk como una PBX.
- Múltiples Aplicaciones específicas como:
 1. Sistemas de Conferencias
 2. Gestión de Call-Centers
 3. Grabación y Monitorización
 4. Sistemas de FAX
 5. Buzones de Voz, Música en Espera, etc...
- Bases de datos y Asterisk Real Time
- Manejo del Call Detail Recording
- Profundizar en la seguridad del sistema Asterisk
- Métodos de Distribución Automática de Llamadas (p.e. Marketing)
- Sistemas Automatic Speech Recognition y Text-To-Speech
- Programación en Asterisk Gateway Interface con un lenguaje de programación interpretado como PHP
- Introducción al CTI de Asterisk: Asterisk Manager Interface.
- Breve análisis de las interfaces web más populares de terceros

1.3. Motivación

Siguiendo en la línea, es curioso observar, como la mayoría de las empresas proveedoras de servicios de telecomunicaciones y en especial, proveedores de centralitas, PBX y sistemas de conmutación telefónica, preservan sistemas de protección, relativamente complejos, con difícil acceso a la documentación relativa para el manejo de las mismas, y aún peor, para el desarrollo.

Esto queda relegado a empresas de segundo y tercer nivel, que se configuran en un plano de “Distribuidores” y “Partners” y queda todo el sistema cerrado, muy parecido al panorama que había en el software hasta hace poco menos de una década donde empresas Líderes como Microsoft, Oracle, o IBM operaban con la misma estructura (y continúan prevalenciando dichos vestigios)

Considerando que las distancias que existían entre el entorno de las telecomunicaciones, concretamente entre sistemas de Hardware y Software, y partiendo que este último factor ha cobrado mayor relevancia, consideré por mi parte viable, el hecho de realizar un acercamiento a este mundo y comprobar las posibilidades que este ofrecía al entorno de la Ingeniería Informática. Gracias a asignaturas como **Redes y Diseño e Interconexión de Redes**, de Tercer curso, surgió mi especial interés por este sector, y me propuse a desarrollarlo en los meses consecutivos, aprovechando además la figura profesional que tenía en aquellos días dentro de mi empresa.

En el momento de empezar a indagar en el mundo de las PBX de algunas marcas como Alcatel, Avaya y Panasonic, observé como comentaba anteriormente, grandes dificultades para profundizar, y curiosamente en mi búsqueda de información para dichos sistemas por Internet, casualmente, encontré la alternativa de Asterisk como una plataforma de telecomunicaciones que además podía funcionar como una PBX convencional, y todo sustentado bajo una estructura Open Source. Lo curioso quedaba en la idea de pensar, como sería posible conectar un sistema Open Source, con toda la red de Telefonía, y justo al poco tiempo descubrir, como Asterisk era justo lo que esto resolvía, gracias a toda una red de Tarjetas de comunicaciones y dispositivos pasarela conectadas a Ethernet, que permitirían realizar esta conexión de forma sistemática y relativamente más simplista y aún así preservando la idea del Software Libre en el trasfondo.

En contrapartida, mi primera experiencia fue observar, como la mayor parte de la documentación, era poco clara y sistemática, aún existiendo suficiente cantidad, y pensar que uno de los grandes impedimentos para muchos posibles usuarios, era que en gran medida, la mayor parte de la documentación se

encontraba en Inglés. Aunque afortunadamente, esto no me afecta personalmente, fui consciente que era una barrera para muchos otros conocidos, y que con el tiempo esto fue una de las grandes figuras que apoyaron mi motivación, para desarrollar un proyecto basado en información relativa a este sistema, íntegramente en Español, dándole un carácter más sistemático y tras conversaciones con mi tutor, surgió la idea de canalizar este planteamiento en forma de WIKI, de ahí surgió la idea de realizar **WIKIAsterisk** 3.

Por otro lado, gracias a mi experiencia dentro del sector de las PYMEs he podido observar la problemática que surge en estas empresas, dadas las increíbles limitaciones económicas, y de estructura, que imponen los sistemas PBX actuales. Pese a que las fichas técnicas de los productos de telefonía, ofrecen aparentemente unos niveles de calidad espectaculares, con una expectativa de infinita funcionalidad, a la hora de solucionar los problemas concretos que surgen en las empresas, se convierte en una auténtica odisea, que muchas veces resulta inconclusa, y otras veces, forzada a ser solucionada mediante la incorporación de software de terceros, que conecta con las Interfaces de Integración de la Telefonía a los Ordenadores (CTI) [42], pero que a cambio, supone un desembolso en licencias considerable. En términos generales, dada la situación del mercado y con la filosofía que tienen las empresas del “bootstrapping” financiero, luchando cada céntimo, este tipo de inversiones no suelen demostrar un retorno a corto plazo demasiado claro, por tanto suele optarse por la solución manual, o menos eficiente, y suelen quedar como proyectos abandonados.

En este sentido, yo he conocido las aplicaciones del sistema Asterisk, como una PBX orientada a las PYME, aunque es cierto que cada vez día están surgiendo mayores implementaciones de las grandes empresas, e incluso los principales proveedores de telefonía integran equipos especialistas en este sistema, las empresas líderes del sector, como Alcatel-Lucent, Panasonic, Avaya, Aastra, etc. siguen dominando en primera línea gracias a su fiabilidad demostrada, aunque es probable que esta estructura vaya mutando en los próximos años. Por tanto, mi conocimiento acerca del sector, me ha permitido idear un concepto práctico para demostrar en gran medida paso a paso, una buena porción de las posibilidades que puede aportar al tejido empresarial.

1.4. Estructura de la Memoria

Este documento recoge toda la información relativa al desarrollo del proyecto. La estructura podría desglosarse de la siguiente forma:

- Un primer capítulo introductorio hablando de los conceptos básicos, motivaciones por el desarrollo y objetivos planificados
- En segunda instancia, todo el desarrollo a nivel de planificación, y el planteamiento desde el inicio de la idea, hasta el fruto como resultado final
- El tercer capítulo se hace un esquema general, sobre el entorno utilizado para desarrollar el medio teórico en el que se sustenta este Proyecto, el sistema WIKI
- En cuarto nivel, se tratan los aspectos esenciales sobre el soporte práctico en forma de Máquina Virtual y los medios utilizados para llevar a cabo esta idea.
- El quinto capítulo, desarrolla un Caso de Estudio utilizando una técnica de enseñanza y aprendizaje desarrollada en la HBS [7] para demostrar a nivel práctico formalmente el concepto subyacente de Asterisk
- El último tema, trata de dar lugar a mis conclusiones personales sobre todo lo desarrollado y algunas reflexiones y dificultades que destacaría al finalizar el desarrollo del proyecto

Por otro lado, se incluyen unos apéndices, para complementar un poco más la información aportada por todos los componentes aplicados en este Proyecto y como medio de soporte.

- Una breve descripción de todo el Software más destacado utilizado para desarrollar el Proyecto
- Una descripción en términos generales de los medios difusión aplicados, para dar a conocer la WIKI especialmente a la finalización del proyecto, presentes y futuros estimados
- Todo el desarrollo de código referenciado al proyecto, específicamente scripts y ficheros de configuración finales.

1.5. Conceptos Generales

En esta sección trataremos por encima algunos conceptos Generales relativos a la Telefonía IP, y en particular al segmento de la telefonía que hace referencia al sistema Asterisk, como **Telefonía 2.0**.

1.5.1. ¿Que es una PBX?

PBX, hace referencia a Private Branch Exchange [48], en esencia es una Central de Conmutación. En los inicios de la telefonía, en estas centrales, el aspecto de conmutación era realizado por personal físico, que conectaba los cables de forma que dos pares podían hablar entre si. Luego incorporaron sistemas mecánicos que hacían esta conmutación, y finalmente, surgieron los sistemas digitales. Hoy en día con el surgimiento de la telefonía IP, ha quedado obsoleto hasta el concepto de conmutación.¹

1.5.2. ¿Que es Asterisk?

Según la definición formal, Asterisk es un entorno de trabajo de código abierto, creado para diseñar aplicaciones de comunicación. Asterisk convierte un ordenador cualquiera, en un completo servidor de comunicaciones. Además potencia los sistemas PBX IP, las pasarelas VoIP, servidores de conferencias, y mucho más. Es utilizado por pequeños y grandes negocios, Call Centers, proveedores de telefonía, y sedes gubernamentales en todo el mundo. Asterisk es gratuito, libre y de código abierto.

Realmente esta pregunta puede ir mucho más allá de esta definición común que se establece para el sistema. Existen ciertas consideraciones que hacen a Asterisk lo que realmente es en la actualidad, y una de las principales surge de la propia naturaleza del sistema: La filosofía Open Source.

Considerando Asterisk como una plataforma integral de comunicaciones, podría decirse que es la más importante en su ámbito, y ha prevalecido como única por muchos años en un entorno, donde todos los sistemas de comunicación directamente competidores, eran totalmente privativos. Aunque con el tiempo, estas fueron sacando interfaces comúnmente conocidos como CTI (Computer Telephony Integration [42]) para la integración de sistemas de terceros para cumplir funciones muy específicas, la potencia de estas interfaces era y sigue siendo, bastante limitada dado que el verdadero núcleo de los sistemas privados, permanencia cerrado al público.

Con Asterisk se sienta un precedente desde el momento que el código es de libre acceso, modificación y ampliación, dado que múltiples colectivos pudieron desarrollar sus necesidades y adaptarlas a lo que venían buscando en un momento concreto, que un sistema de comunicaciones pudiera ofrecerles, dentro de sus extensivas bondades.

1.5.3. ¿Que es Telefonía IP?

Varios años después del nacimiento del protocolo SIP, este resulto un candidato idóneo para establecer un marco basado en un entorno de telefonía para Internet, dentro un sistema de comunicaciones entre dos

¹Un curioso video explicativo de la primera evolución de la telefonía: <http://www.youtube.com/watch?v=zDEqDtGaHW0>

puntos por mensajes (he de ahí el concepto protocolo), concretamente destinadas al envío y recepción de datos multimedia. Los primeros datos, resultaron ser de tipo audio exactamente igual que los sistemas tradicionales de telefonía [25]. Como se encontraban utilizando un protocolo destinado al entorno de Internet, y se realizaba una comunicación al estilo telefónico, de ahí apareció el concepto, también denominado VoIP [40] [49].

Realmente existen otros protocolos en los que se basa la Telefonía IP, como los desarrollados por Skype (ahora Microsoft) basado en un códec propietario y Google basado en el códec libre iLBC. Cisco entre otros también ofrece protocolos propietarios, pero de menor relevancia a nivel público como el SCCP. Concretamente la principal característica del primero que es el más popular a nivel mundial, es la increíble integración con todo tipo de redes, además de aprovechar la propia red de Pares (el ordenador de cada usuario) para optimizar la calidad de las llamadas, eligiendo a los candidatos mejores para cada llamada en función del punto donde nos conectásemos, al puro estilo peer to peer (P2P, originalmente desarrollado a partir del software Kazaa).

Comparativamente, SIP se basa en la idea cliente servidor, en el que todos los pares conectan a un solo servidor principal. Esto supone las ventajas subyacentes a la gestión directa de los pares, y desventajas en la comunicación dado que muchas veces puede ser que la ruta al mismo en función del lugar pueda ser deficiente, degenerando la calidad de las llamadas comparativamente.

1.5.4. ¿Que es la Telefonía 2.0?

El término oficial de la Telefonía 2.0 es el de las **Comunicaciones Unificadas** [43]. Hay que considerar que hoy en día existen múltiples sistemas de comunicación, los cuales suelen ser utilizados con sistemas individuales que además no tienen ningún tipo de relación entre ellos, y en consecuencia no pueden resurgir posibles sinergias entre los mismos.

Además el mantenimiento individual de cada uno de estos sistemas implica un mayor consumo de tiempo, que repercute en costes productivos, de personal, de seguridad, infraestructura y mantenimiento. Por ello empezaron a surgir determinados sistemas específicos cuya idea se sustentaba en servicio de aplicación general para dotar un concepto en concreto y que abarcara la plenitud de las necesidades actuales. Es muy común en este caso, el ejemplo de las credenciales de acceso a los sistemas.

Siguiendo este ejemplo, surgió un protocolo específico para almacenar información relativa a individuales, llamado Lightweight Directory Access Protocol, que concebía esta idea en forma de directorios. Con ello era posible la capacidad de crear una estructura de usuarios, y todas las aplicaciones que requiriesen un sistema de autenticación basado en usuario y contraseña, pudieran aprovechar la información almacenada en el mismo. Esta idea podría conceptualizarse en forma de Base de Datos, pero realmente trasciende más allá de esta, dado que queda encuadrada en un marco conceptual basado en un protocolo, como fue SIP en sus días, para la Telefonía IP.

Lo mismo ocurre por ejemplo, con los sistemas de Correo Electrónico, Buzones de Voz de Telefonía, Bandejas de FAX, y comunicaciones en tiempo real como las Conferencias y la mensajería instantánea, son múltiples mecanismos que hoy en día casi todas las empresas utilizan de manera regular, pero haciendo reflexión, y exceptuando contados casos si recordamos, existe una distinción para cada uno de estos sistemas claramente demarcada. Eventualmente los nuevos sistemas On-Demand (a demanda), también han surgido para aplacar un poco los efectos colaterales de no recurrir a tiempo a las comunicaciones unificadas.

Un ejemplo de todo esto podría ser, el clásico uso de un entorno web-mail por ejemplo basado en un servidor de correo Microsoft Exchange, funcionando simultáneamente, con un buzón de voz en el teléfono fijo ofrecido por el software de la PBX de la empresa, y añadiendo otro buzón de voz, ofrecido por la

operadora de telecomunicaciones. Utilizamos Google Talk para comunicarnos con nuestros compañeros de trabajo, la empresa tiene contratados servicios On-Demand de Adobe Connect para establecer las reuniones virtuales, y todos los FAX entran en una fotocopidora central, a la que van a parar los FAX de todo el departamento juntos apilados en la misma bandeja desorganizadamente.

El resultado es en la mayor parte de los casos, una infra-utilización de los sistemas de mensajería instantánea, y una sobre-utilización de los sistemas de e-mail, con tendencia a concebirlos como pseudo-sistemas de mensajería instantánea de forma errónea ya que originalmente no fueron diseñados para ello ². Por otro lado, solo atendemos a FAXes muy específicos que solicitamos de forma expresa, dado que la gran desorganización existente en la pila de FAXes recibidos resulta demasiado molesta, y tenemos que pagar una cantidad de dinero considerable para poder tener el servicio de Adobe Connect funcionando, para servicios muy puntuales y eventualmente infrautilizados también para las reuniones online, dado que todavía no reemplaza las reuniones reales (En este sentido Skype dado que es gratuito su uso, esta cubriendo estas necesidades aunque no fuera esa su idea original, volviendo al ejemplo del correo vs la mensajería instantánea). Y para terminar, es clásico ver como la mayor parte de los usuarios de telefonía tiene su buzón de voz activado, pero casi nunca hace escucha del mismo, lo que resulta en desconfianza del medio muy generalizada, aunque viéndolo prácticamente sea extremadamente útil, dado que no siempre hay un teclado delante para dar un mensaje.

¿Que ocurriría, si en la misma bandeja de correo electrónico, entrarán los mensajes de audio del buzón de voz en un formato ligero, tanto para los mensajes dejados en nuestro móvil como en nuestro teléfono de sobremesa, y adicionalmente, todos los FAXes entrarán como ficheros adjuntos dentro de nuestra buzón de e-mail simultáneamente?

¿Sería viable que desde un teléfono, ubicado dentro de nuestro ordenador, pudiéramos realizar conferencias de audio o video, recibir mensajes instantáneos, y que en caso de ausencia estos se convirtieran en correos electrónicos, y de forma añadida pudiéramos cursar llamadas a todos los destinos posibles, y utilizando por ejemplo un teléfono convencional para ejercer como dispositivo de entrada y salida de audio?

Estas son las dos preguntas que tratan de resolver la Telefonía 2.0 en la actualidad.

²Clásico ejemplo de dos personas al teléfono, una le pide a la otra su e-mail para enviarle un correo sobre la marcha, y el correo no llega, al final acaban colgando si pode haber realizado la gestión in-situ que pretendían dado que el correo no resulto en ese momento lo eficiente que creían debería haber sido

Capítulo 2

Planificación y Planteamiento

2.1. Planteamiento

El desarrollo del proyecto se ha sustentado en tres pilares fundamentales:

1. Desarrollo de una Wiki, para la base teórica que veremos más desarrollado en el capítulo 3
2. Desarrollo de un Caso de Estudio aplicado en el capítulo 5, siguiendo el **Método del Caso**
3. Creación de una Máquina Virtual, para crear un entorno de desarrollo y demostración basado en el Caso de Estudio, y visto en profundidad en el capítulo 4

2.1.1. Método del Caso

El método del Caso es un sistema desarrollado para la enseñanza específicamente, y esta probado ser uno de los sistemas más eficientes de la actualidad para el aprendizaje teórico-práctico con un especial hincapie en el segundo elemento, y está más orientado al desarrollo que a la investigación. Es por esto, por lo que cada día en más programas de Postgrado se esta incorporando esta metodología satisfactoriamente.

Curiosamente, esta metodología es relativamente reciente, comparado quizá a otras metodologías centenarias o milenarias, desarrollada por la Universidad de Harvard[7] a principios del siglo XX [36], orientada inicialmente a la carrera universitaria de Derecho, en la que se ponían casos reales de temas legales, y los alumnos tenían que enfrentarse a los mismos, siguiendo conceptos aprendidos en clase, y documentándose por sus propios medios para conseguir resoluciones positivas. Esto de alguna forma, incentiva el ingenio, dado que esta demostrado, que el sistema de enseñanza magistral aunque pueda intentar resultar más interactivo y participativo según la calidad del docente, la información realmente solo se produce de forma unidireccional del docente a los alumnos y el aprendizaje nunca suele derivarse, dado que suelen ser sistemas con orientación dirigida.

Debido a que una de las motivaciones de este proyecto, era desarrollarlo siguiéndolo una metodología eminentemente práctica, y mi pretensión fuese, que trascendiera lo máximo posible, para personas interesadas en el sistema Asterisk en el futuro. Por ello la intención de encuadrarlo todo dentro de un marco didáctico y formativo, en vez de un marco de carácter divulgativo o investigativo.

Concretamente para el desarrollo de este proyecto, el uso del Método del Caso, se dará aplicando un único Caso de Estudio, basado en una empresa del sector de la Automoción recién entrada al negocio, la cual desea introducirse ampliamente, en las tecnologías de la comunicación de forma eficiente, y para ello toma la colaboración de nosotros, representando una pequeña empresa que implanta sistemas de comunicaciones basados en Asterisk. Todo esto queda desarrollado al detalle a lo largo del **Capítulo 5**.

Dentro del capítulo en cuestión, cada sección se va subdividiendo en varios subapartados. La idea conceptual es la siguiente:

1. Primero se enuncia el caso de estudio, explicando los problemas, las necesidades que surgen para cubrir ese problema, y algunas ideas puntuales, sobre como orientar la solución, pero de manera solo lo suficientemente específica para poder ajustarla de manera efectiva.
2. En segundo nivel, se trabaja el caso desde una de las múltiples perspectivas posibles de resolución que existe, aportando al lector, un grupo de alumnos que comparten el mismo interes, o en grupo con un docente que dirija la sesión, una guía para orientar el camino que da lugar a una posible solución concreta
3. En tercer lugar, da lugar al sistema de comprobación de que la solución llevada a cabo original, o alternativa, produce los efectos deseados según la especificación del Caso de Estudio en cuestión.

En resumen, puede verse en la Figura 2.1 de ejemplo que muestra las diferentes fases de un Caso siguiendo la Metodología del Caso en cuestión.

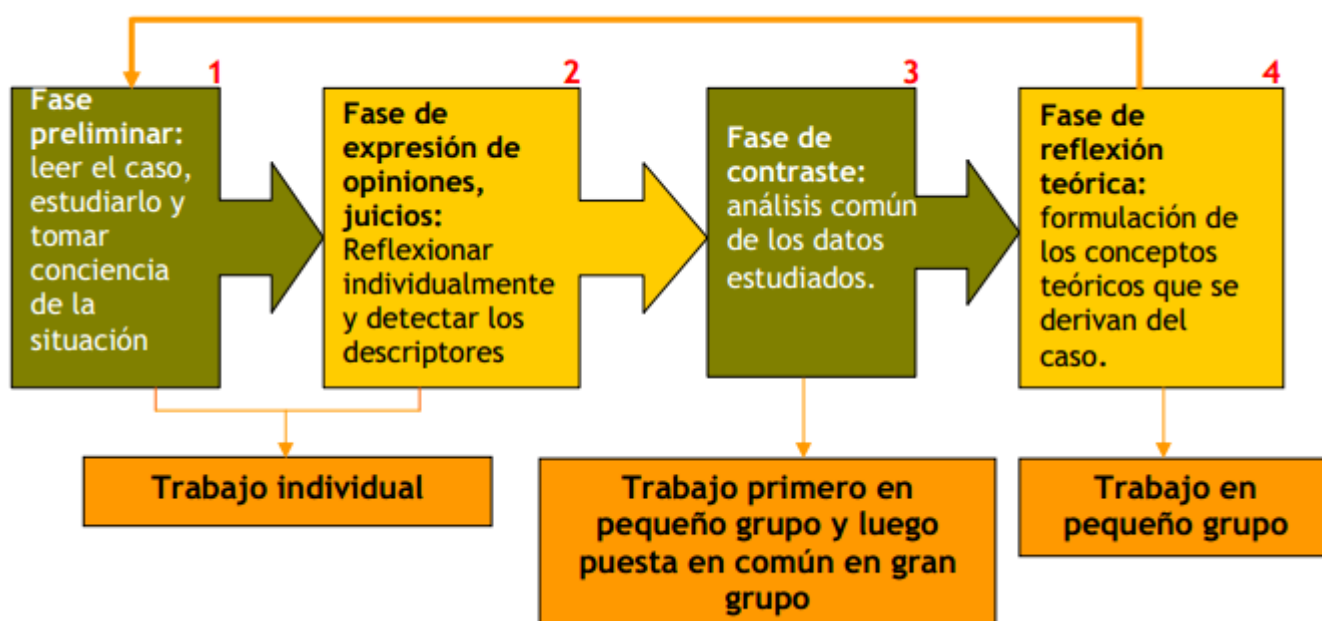


Figura 2.1: Esquema general de una sesión siguiendo el Método del Caso

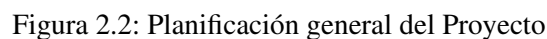
2.2. Planificación

Para considerar la planificación del Proyecto se han podido observar dos periodos comprendidos antes del inicio, y al comienzo de la escritura de este proyecto.

2.2.1. Aprendizaje e Investigación

En primer lugar se dio una fase de aprendizaje del sistema Asterisk. Esta fase comenzó a mediados del 2010, durante 6 meses, acompañado de un curso de la empresa colaboradora y que actualmente lidera el proyecto Asterisk, Digium [17]. Este curso era de carácter inicial (llamado **Fast Start**), y cubría aspectos básicos de inicio para empezar a realizar algunos desarrollos de baja escala. Podría considerarse un curso de iniciación al mundo de las PBX orientado específicamente a Asterisk.

La programación general en estos meses quedaría definida según la Figura 2.2, todas las figuras de planificación han sido diseñadas con la herramienta Planner [26]



A principios de Marzo 2012, se diseña la programación para la creación de este proyecto que durará hasta finales de Junio según previsto, lo que incluye solo la fase de desarrollo. Esta programación se divide en los tres aspectos principales en los que se basa este proyecto: El Caso de Estudio, la Máquina Virtual de Desarrollo y la WIKI teórica.

9

De forma concurrente, durante el desarrollo de varios Artículos más significativos, se iría desarrollando el Caso de Estudio haciendo referencia a los mismos. Esto empezaría aproximadamente dos semanas después del inicio del desarrollo de la WIKI, y acabaría unas dos semanas después justo coincidiendo con la fecha límite del proyecto.

Y también de forma simultánea, el Caso de Estudio se sustentaría en en desarrollo de la máquina virtual, que comenzaría prácticamente también a la vez, y acabaría una semana antes aproximadamente, dado que la última semana serviría adicionalmente para configurar y retocar en términos generales los detalles de este documento.

Todo esto quedaría reflejado en la Figura 2.3.

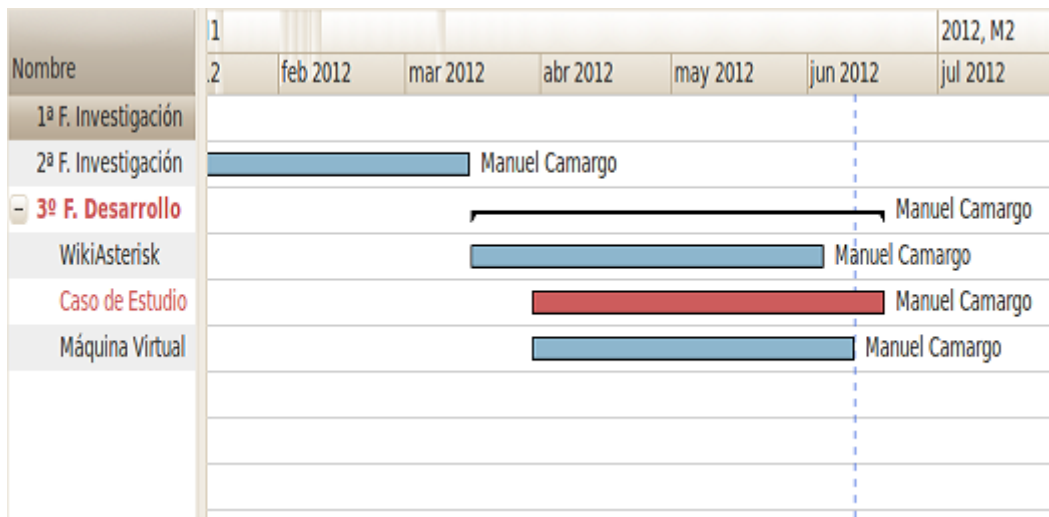


Figura 2.3: Planificación del Desarrollo del Proyecto

2.2.3. Fases de Creación

Durante el proceso de desarrollo, las distintas secciones se programaron de forma equitativa, agrupándolas por módulos conceptuales. El procedimiento era el siguiente:

1. Desarrollo de la teoría y las bases que sustentaban la sección
2. Pruebas prácticas en Máquina Virtual, desarrollo, puesta a punto y documentación
3. Diseño y creación del Caso de Estudio que encuadre el problema planteado

El desarrollo de las secciones puede contemplarse en la Figura 2.4 y la descripción de las mismas fue la siguiente:

1. Introducción

En este apartado, se pretende contemplar una visión panorámica del sistema Asterisk, incluyendo todos los datos generales sobre el sistema, historia y su funcionamiento interno, Arquitectura en esencia, los procedimientos de Instalación y Actualización

2. Protocolos basados en la Red

La idea general, es dar pinceladas muy generales sobre los protocolos más distendidos, pero no concretar demasiado, dado que la intención de este proyecto no es profundizar tanto en la Telefonía

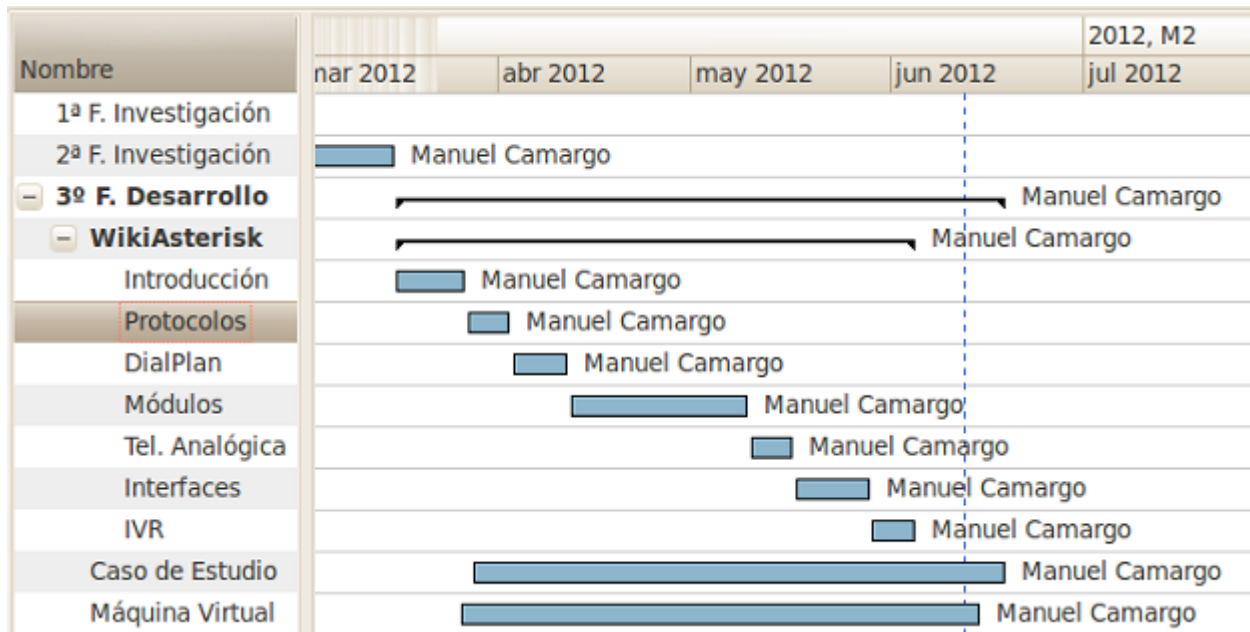


Figura 2.4: Fases del Desarrollo del Proyecto

IP, sino en el sistema Asterisk en concreto y como maneja las situaciones a nivel productivo. Es por ello que el planteamiento a seguir es el de desarrollar información acerca de estos protocolos enfocándolos exclusivamente a su papel que desempeñan dentro del sistema Asterisk

3. Primeros Pasos en el Plan de Marcación

Se pretende agrupar, todo el procedimiento general para empezar a desarrollar los primeros planes de marcación, todos las aplicaciones más comunes y una primera aproximación para el lector. En esto se incluyen no solo todas las ideas generales, basadas en este sistema de programación propio de Asterisk, con carácter de script, sino sus referencias más comunes. Además era importante tener una aproximación inicial al los protocolos dado que la aplicación de un plan de marcación siempre va directamente relacionado a los mismos.

4. Grupos de Módulos

La intención es ofrecer una visión general sobre los Módulos que fundamentan los pilares básicos de Asterisk, agrupados por categorías o en función del papel que desempeñan a nivel empresarial, o funcional. Evidentemente es la parte más densa del proyecto, aunque mas o menos se dedica el mismo tiempo por módulo dado que aproximadamente es posible desarrollar la misma cantidad de contenido para los respectivos.

- Sistema de Buzones de Voz
- Call Centers y Colas de Llamadas
- Grabación y Monitorización de Llamadas
- Sistemas de Gestión de FAX
- Bases de Datos y Configuración en RealTime

- Generación Automática de Llamadas y Marcadores
- Registro de Llamadas y Eventos
- Sistemas de Conferencia

Además de todo esto también cubrir ciertas Interfaces Web Gráficas que hacen alguno de estos módulos más versátiles y dinámicos por definición

5. **Introducción a las Interfaces de Telefonía Analógica**

Dado que la pretensión de este proyecto es focalizar principalmente en los sistemas de voz sobre IP, también existe un nicho en Asterisk relacionado al mundo de la telefonía analógica y es necesario cubrirlo para aportar más soluciones y más considerando que en gran medida el tejido empresarial todavía se nutre de los mismos. En esta parte la idea es cubrir tanto las tarjetas que incorporan interfaces de telefonía analógica como los Gateway que interfazan a través de un protocolo SIP con nuestro Asterisk.

6. **Interfaces de Comunicaciones Unificadas**

La idea detrás de esto es poder demostrar el verdadero “poder” del sistema Asterisk, dado que el manejo de las interfaces que ofrece Asterisk es lo que permite las posibilidades ilimitadas de interconexión con otros sistemas y ofrecer las sinergias relativas. En este apartado la idea es detenerse un poco más y dar lugar a las pruebas, mayor investigación a fondo y desarrollos más precisos. Principalmente hay dos áreas de cobertura en función de cada una de las dos interfaces de Asterisk:

- Asterisk Manager Interface
- Asterisk Gateway Interface

7. **Creación de Mecanismos de Respuesta Interactiva de Voz**

Finalmente, aunando todo lo visto anteriormente, poder desarrollar y profundizar en el contexto, dar a conocer las técnicas de Interacción por Voz entre las máquinas y las personas, y ofrecer el concepto diseñar un sistema de Telefonía 2.0 de pequeña escala, para demostrar las bondades de esta tecnología en desarrollo

Capítulo 3

Documentación Teórica

En este capítulo desarrollaremos un poco más la idea detrás de desarrollar una plataforma teórica basada en un sistema WIKI, concretamente el software MediaWiki, todas sus repercusiones y ventajas que aportan en general al desarrollo de este proyecto.

También daremos un recorrido por la estructura de la WIKI [44] y así poder entender el concepto subyacente de su estructura.

3.1. Sistema MediaWiki

Para la realización del sistema WIKI se ha usado de la plataforma MediaWiki [8] la cual esta diseñada específicamente para este propósito.

Es conveniente considerar que la edición en la misma es semejante a la de otras WIKIs y concretamente a la más popular, la de Wikipedia dado que en general todos estas aplicaciones pertenecen a la misma fuente.

Realmente más allá de destacar como se ha realizado el despliegue del sistema, y las buenas prácticas, en esta sección vamos a tratar la estructura de la WIKI en sí.

3.1.1. Origen de la WIKI

Desde el momento que tome la determinación de desarrollar la WIKI, mi primer planteamiento fue lanzarla públicamente a un servidor en Internet de mi propiedad y enmarcarla bajo un dominio que en un futuro pudiera facilitar su adecuación. La primera idea que rondaba en mi mente era el diseño gráfico de toda una referencia basada en el sistema Asterisk con alguna referencia a España, para dar a entender fielmente la intención asociada al idioma que se pretendía instaurar.

A partir de aquí realice el registro del dominio [22] muy significativo para el tema que pretendía cubrir, y simultáneamente, lo asocié al servidor web que comentaba anteriormente. Con esto resuelto ya solo quedaba instalar MediaWiki y configurarla con los parámetros estándar de uso. De aquí recibiría su nombre final el proyecto WIKI: **WIKIAsterisk**

El planteamiento original era dejar la WIKI sin acceso público hasta la finalización integral del proyecto y conclusión favorable, y tras esto, dar acceso únicamente a editores de confianza. El método para definir editores de confianza todavía no ha quedado demasiado definido, pero el concepto de dejarla libremente

pública todavía no me resulta atractivo, dado que tengo constancia que existen bastantes sistemas destructivos desatendidos en Internet, y las exigencias de Moderación pueden resultar excesivas, al menos para preservar íntegramente los contenidos de calidad.

Otro planteamiento sería el bloqueo de las páginas principales, y dejar libre acceso para crear otras páginas desarrollando nuevos temas. Solo los moderadores, y editores de confianza tendrían acceso a este tipo de edición y también esta alternativa es candidata a ser resolutive tras la “liberación del sistema”

3.2. Estructura General

La WIKI esta dividida en 6 secciones claramente diferenciadas, como puede verse en la Figura 3.1:

1. Introducción
2. Plan de Marcación
3. Protocolos e Interfaces
4. Módulos Principales
5. Conceptos Avanzados
6. Bibliografía y Referencias



Figura 3.1: Estructura general de WIKIAsterisk

3.2.1. Fundamentos

El apartado de Fundamentos de Asterisk, está enfocado a dar una perspectiva general del sistema, y establecer los primeros precedentes para su funcionamiento, partiendo de la instalación del mismo, se subdivide en tres artículos principales, según figura 3.2:

1. **Introducción** En este artículo se definen conceptos básicos sobre Asterisk, se trata un poco sobre su historia, creadores, y nuevos proyectos como Asterisk Scalable Communications Framework (Asterisk SCF), que está destinado a convertirse en el futuro real de Asterisk aunque todavía no ha sufrido el despegue inicial dado que se encuentra en fase de desarrollo.
2. **Instalación**
Como su nombre hace referencia en este artículo se definen todos los aspectos relevantes a la instalación de Asterisk en nuestro sistema operativo de elección.
3. **Arquitectura**
Aquí se desarrolla en detalle toda la estructura interna de Asterisk, en concepto de Arquitectura, como se ha conceptualizado el proyecto, buscando siempre el máximo nivel de escalabilidad e independencia, basada en un sistema de módulos integrables individualmente, y una interconexión entre los mismos con metodología estándar.

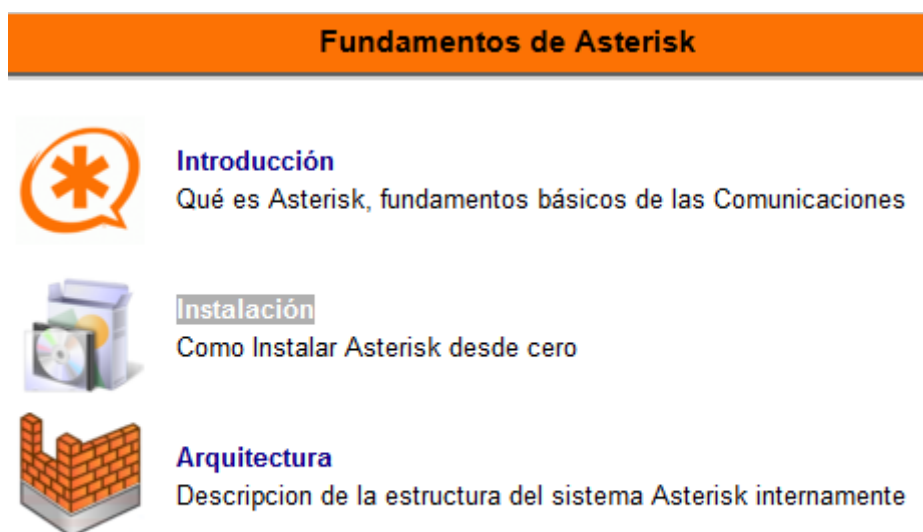


Figura 3.2: Esquema de la sección Introductoria de la WIKI

3.2.2. Plan de Marcación

La base del sistema Asterisk es el plan de Marcación. Realmente es la columna vertebral y es el método para poner en funcionamiento a voluntad del desarrollador todas las posibilidades que ofrece esta plataforma de telecomunicaciones. Todo esto puede verse en la figura 3.3

1. **Introducción Dialplan** Podría considerarse el primer capítulo introductorio a todo el concepto y distribución del Plan de Marcación. Como se fundamenta cada concepto y los primeros pasos para construir una aplicación funcional basada en lo que Asterisk puede ofrecer.

2. Aplicaciones Básicas

Para poder poner en funcionamiento lo aprendido a través del apartado de introducción al Plan de Marcación es necesario conocer una serie de aplicaciones que podrían considerarse de índole imprescindible. En este apartado se van desarrollando cada una de estas aplicaciones individualmente con una detallada explicación del funcionamiento de las mismas.

3. Funciones

Quizá uno de los conceptos menos evidentes y reconocidos del sistema Asterisk, dado que aunque realmente sean verdaderas funciones propiamente dichas, son consideradas como elementos internos y estáticos del sistema, que generalmente solo hacen referencia práctica a modificaciones de comportamiento o características de la estructura de módulos, aunque posiblemente para entender este concepto es necesario realizar una recapitulación general de la Arquitectura del sistema.

4. AstDB

Asterisk incorpora una base de datos interna, para almacenar información poco sensible y quizá bastante dinámica de carácter no relacional llamado Berkeley DB. Quizá podría considerarse como una de las alternativas más flexibles pero a su vez menos potentes. Puede servir para pequeñas ideas, pero en el largo plazo es algo relativamente poco práctico. En este artículo se desarrolla muy por encima esta idea en términos muy generales.

5. Dialplan Avanzado

Este artículo quizá pueda parecer el más desubicado de toda la Wiki porque se adentra en temas más sofisticados, pero a su vez se encuentra en el sitio apropiado dado que podría considerarse el último paso en el dominio del Plan de Marcación. Hay que considerar que una vez que hemos aunado todos los conceptos dispares relacionados al sistema Asterisk, todas las técnicas adicionales para construir Planes de Marcación mas sofisticados hacen de Asterisk una herramienta aún más potente y práctica. También se hace una introducción teórica a los sistemas de Respuesta de Voz Interactiva, porque realmente estos últimos son más comprensibles en la práctica que sobre el papel.

3.2.3. Protocolos e Interfaces

En esta sección se tratan de cubrir todos los elementos relacionados al intercambio de mensajes o información, por resumirlo en líneas muy generales. Por un lado contamos con los Protocolos más comunes de Asterisk, aun habiéndose descartado algunos secundarios como el SCCP, Skinny o el MGCP de Cisco, los cuales también ofrecen servicio Asterisk, pero no son los más utilizados a nivel general. Todo esto puede observarse en la Figura 3.4

1. **SIP** En este capítulo se explica en gran detalle el protocolo más importante asociado a Asterisk y la Telefonía IP. No es una referencia para conocer el protocolo SIP en particular, sino para conocer como funciona el protocolo en relación a Asterisk y que provecho podemos sacar del mismo

2. **IAX**

Por otro lado tenemos un protocolo específico, creado por la misma persona que fabricó Asterisk para cubrir una serie de deficiencias propias de SIP e intentar aunar los medios y la señalización, además de estandarizar un protocolo que para aquel momento se encontraba demasiado disperso. Aunque hoy en día ha seguido evolucionando SIP, e IAX ha quedado relegado como protocolo específico de comunicación entre máquinas Asterisk. Todo esto con mucho más en detalle queda explicado de este apartado

3. **CLI**



Figura 3.3: Esquema de la sección del Plan de Marcación de la WIKI

Asterisk aporta una interfaz muy singular basada en línea de comandos, para los administradores en la cual se pueden lanzar los mismos. Pero realmente el interés de esta interfaz suele ser de carácter informativo, dado que va ofreciendo un importante nivel de detalle de eventos configurable, del sistema interno de Asterisk en tiempo real. En esta sección hacemos un muy breve resumen al respecto dado que su función en el proyecto es meramente transitiva, aunque para el día a día de un administrador de Asterisk pueda llegar a ser importante.

4. DAHDI

Estas siglas hacen referencia a Digium/Asterisk Hardware Device Interface. Es decir, una interfaz para poder dialogar con el hardware específico de Digium y compatible, en relación a Asterisk. Esto hace realmente referencia a tarjetas para conectar con la telefonía clásica. Es un mundo donde es posible profundizar bastante, pero dado que la intención del proyecto es dar una breve visión sobre estas capacidades pero no profundizar demasiado en el detalle de las mismas, se realiza un buen repaso sobre las opciones más comunes dentro de esta rama.

5. Pasarelas ATA

Las Pasarelas o Gateways, juegan el papel de las tarjetas de comunicación como las vistas en el apartado de DAHDI, pero enlazando con nuestra máquina Asterisk a través del protocolo SIP. Existen cientos de Pasarelas de este tipo, con muchos submodelos a su vez. Esta sección esta destinada a poder ofrecer una guía y un manual claro y conciso de la funcionalidad que aportan las pasarelas más reconocidas en el mercado actual.

Protocolos e Interfaces



SIP

Toda la información necesaria acerca del Protocolo de Inicio de Sesión asociada a Asterisk



IAX

Información concreta del Protocolo específico de Intercambio entre sistemas Asterisk



CLI

Información acerca de la Interfaz de Línea de Comandos del sistema



DAHDI

Interfaz de Dispositivos Hardware de Digium para Asterisk



Pasarelas ATA

Métodos de conexión con la red de telefonía estándar alternativos

Figura 3.4: Esquema de la sección de los Protocolos e Interfaces de la WIKI

3.2.4. Sistema de Módulos

Siguiendo el esquema establecido en la Arquitectura de Asterisk, considerando que Asterisk se fundamenta principalmente en el despliegue de módulos que van cubriendo las necesidades específicas de forma autónoma, en este apartado se explican con gran detalle los módulos que conforman la base de las Comunicaciones Unificadas que ofrecen Asterisk, en un término nivel medio, considerándolos todos como fundamentales pero a su vez muy específicos en función de las distintas combinaciones de instalaciones posibles. En detalle puede verse un extracto en la Figura 3.5

1. **Buzones de Voz** Tratamos con mucho detalle todos los aspectos relevantes al sistema de Buzones de Voz, tanto a nivel de configuración como de gestión. Este es uno de los apartados que suele ofrecer más funcionalidad dentro del sistema Asterisk pero que suele quedar relegado si no aplicamos un nivel de configuración razonable para lo que ofrece, por eso aquí detallamos como alcanzarlo.
2. **MeetMe**
Aquí hacemos referencia al sistema de Conferencias ofrecido por Asterisk llamado MeetMe, que realmente sigue destacando por su uso. Esta sección es muy probable que sufra modificaciones en un futuro próximo dado que ya también hacemos alguna mención a las nuevas tendencias en este entorno dentro del Artículo

3. Colas de Llamadas

Este capítulo está orientado al mundo de los Call Centers como primeros y fundamentales usuarios de la estructura de colas de llamadas como tal, y agentes atendiendo principalmente. Hacemos especial mención a varios aspectos derivados de este tema tales como las Estrategias en la entrada de llamadas, sistema de prioridades, y aspectos generales que hacen referencia a este mundo particular.

4. Monitorización de Llamadas

En Asterisk el término monitorización hace referencia al hecho de estar conectado activamente a un canal de audio. En esta sección se tratan de definir los mecanismos de grabación y escucha activa de llamada que ofrece nuestro sistema.

5. Música en Espera

El concepto de Música en Espera (MOH), también está externalizado como un módulo en Asterisk y es relativamente configurable. Solamente damos unas pinceladas sobre el tema, aunque es algo que puede resultar relativamente extenso en función de las posibilidades que podamos encontrar en esta funcionalidad.

6. FAX

Los sistemas de FAX pueden haber quedado ya prácticamente obsoletos, pero aun así se siguen utilizando de forma bastante generalizada. Asterisk ofrece la posibilidad de configurarlos e integrarlos en mayor medida, y aquí se explican algunas formas para realizar este cometido.

7. Codecs y Formatos

Una breve descripción sobre la gran mayoría de codecs y formatos disponibles en el sistema Asterisk.

8. Asterisk Realtime

Aquí se detalla con gran precisión, como Asterisk es capaz de operar con bases de datos, y concretamente una capacidad que tiene específica de llevar el control de la configuración interna del sistema directamente desde tablas dinámicas basadas en bases de datos de tipo relacional.

3.2.5. Conceptos Avanzados

Una vez que se ha cubierto la gran parte de los conceptos cruciales para el buen entendimiento del sistema Asterisk, es necesario dotar la WIKI de un apartado con conceptos más avanzados en el aprendizaje que suelen surgir la necesidad en etapas más desarrolladas. Entre estos conceptos se incluyen ideas sobre el control, sobre la capacidad de desarrollar nuevos aspectos propios, y capacidades para interconectar sistemas externos a Asterisk, con este de forma eficaz. [3.6](#)

1. **Seguridad** Es importante tratar todos los aspectos relevantes a la seguridad ejemplar de un sistema integral, dentro de un apartado específico. Quizá podían haber quedado dispersos en cada una de las secciones correspondientes, pero con este apartado se pretende dar una visión más orientada a la auditoría que a las buenas prácticas.

2. Generación Automática de Llamadas

Cuando pensamos en la capacidad de generar llamadas de forma automática, nos suele venir a la mente intenciones orientadas al Marketing, y a la posibilidad de ofrecer algo de forma automatizada. Pero yendo un poco más lejos aquí explicamos muchas de las funciones que podemos utilizar con este sistema que provee Asterisk para automatizar una serie de procesos relativos entre la telefonía y la computación.

Sistema de Módulos



Buzones de Voz

Gestión y Configuración del sistema de Buzones



MeetMe

Sistema de Conferencias específico de Asterisk



Colas

Sistema de gestión y distribución Automático de Llamadas



Monitorización de Llamadas

Sistemas de grabación, monitorización y escucha de Llamadas



Música en Espera

Configuración General del mecanismo de Música en Espera



FAX

Sistemas de Fax en Asterisk Oficiales y No Oficiales



Codecs y Formatos

Información acerca de los Codecs y Formatos en el sistema



Asterisk Realtime

Como crear una configuración de Asterisk Dinámica en Tiempo Real

Figura 3.5: Esquema de la sección de Módulos Asterisk

3. Registro Llamadas y Eventos

Asterisk establece un riguroso control de varios aspectos, entre ellos incluso, del flujo de llamadas y eventos que van ocurriendo, tal y como podría ocurrir con otros tipos de servidores como los dedicados a ofrecer servicios Web, pueden almacenar todas las peticiones de entrada y salida al servidor. Con esto se pueden establecer mecanismos de control, facturación entre otros, especialmente interesantes desde una perspectiva eminentemente práctica.

4. TTS y ASR

TTS hace referencia a Text-To-Speech, es decir, Texto a Voz, o Síntesis de Voz, y ASR, significa Automático Speech Recognition, o reconocimiento de Voz Automática. Podrían considerarse las

interfaces de Voz entre la máquina Asterisk, y las personas que pudieren utilizarlas. Estos dos sistemas en esencia vienen explicados en profundidad haciendo referencia a las múltiples alternativas que nos podemos encontrar en este entorno

5. **AGI**

El sistema AGI, Asterisk Gateway Interface, está orientado a poder ofrecer un soporte externo de programación en múltiples tipos de lenguajes de programación, para desarrollar aplicaciones específicas que interactúen con los flujos de llamadas dentro de Asterisk, con mecanismos relativamente sencillos de implementar y muy orientados a un entorno de desarrollo para mejorar la funcionalidad intrínseca del sistema

6. **AMI**

Asterisk Manager Interface, es una interfaz modular, totalmente independiente de Asterisk, que establece un tipo de Pasarela a través de la cual puede recibir comandos para invocar aplicaciones dentro del sistema Asterisk, y simultáneamente vuelca información en la misma, accesible desde el exterior y proveniente específicamente de todos los módulos que están interconectados dentro de la estructura de nuestra máquina de telecomunicaciones.

7. **Interfaces Web**

Por cada uno de los módulos explicados, podría decirse que pueden existir una o varias interfaces cuya función principal es cubrir una serie de aspectos relativos a los mismos, como la configuración, el control y la gestión o la monitorización de los eventos que suceden en ellos. Dado que en el Caso de Estudio, se plantea el uso de determinados Interfaces Web como alternativas interesantes, aprovechamos para desarrollarlos de manera práctica principalmente con algunos aspectos teóricos específicos de los mismos.

3.2.6. **Bibliografía y Enlaces de Interés**

El último apartado simplemente esta destinado a realizar referencias a determinadas páginas de interés general dentro del mundo Asterisk, otro tipo de páginas que hayan podido tener alguna relación con WIKIAsterisk, y finalmente, medios de comunicación para establecer contacto con el autor de este proyecto. [3.7](#)

1. **Asterisk: The Definitive Guide** Este libro es por muchos considerado, la “Biblia” de Asterisk y podría decirse que gran parte de esta WIKI se fundamenta en este libro. Además es la lectura cabecera por la mayoría de los expertos. Una de las pretensiones de esta Wiki es la de superar algún día los contenidos de este libro tanto en calidad como en cantidad con el resto de las aportaciones de valor.
2. **Github Asterisk-PFC**
Dado que el proyecto ha sido gestionado desde diversas máquinas, entre las cuales se encuentran, el ordenador de trabajo y el servidor donde se aloja la máquina virtual del proyecto, y además con la idea de poder realizar un seguimiento de las distintas ediciones que ha ido sufriendo el proyecto desde las primeras instancias, se originó el planteamiento de alojar la mayor parte de los ficheros más relevantes al mismo, dentro de un sistema de control de versiones, y el elegido fue GIT [\[32\]](#), dentro ampliamente conocido repositorio **Github** [\[18\]](#).
3. **Guía Wikimedia**
Una de las guías que más útil me resultó para introducirme en el mundo de la edición de textos en formato Wiki, y en el que se sustenta todo el desarrollo de WikiAsterisk, pasando desde el diseño estructural, hasta las plantillas para simplificar la labor y modelación.

Conceptos Avanzados



Seguridad

Conceptos Generales acerca de la seguridad en Asterisk



Generación Automática de Llamadas

Métodos para enviar llamadas automáticamente



Registro Llamadas y Eventos

Información acerca del CDR y CEL de Asterisk



TTS y ASR

Manipulando la Voz: Text-to-Speech y Automatic Speech Recognition



AGI

Asterisk Gateway Interface, la interfaz de programación



AMI

Asterisk Manager Interface, el CTI de Asterisk



Interfaces Web

Sistemas Web que facilitan la visualización y la gestión de ciertos Módulos de Asterisk

Figura 3.6: Esquema de la sección de Aspectos Avanzados del Sistema

4. Contacto

Varios medios de comunicación para ponerse en contacto conmigo como Autor principal y único de este proyecto.

3.3. Estructura de los Artículos

El planteamiento establecido para desarrollar los artículos estuvo basado en una Estructura general definida en la mayoría de los artículos fundamentada en la siguiente base:

1. Descripción corta

Una breve descripción introductoria para el artículo explicando brevemente la posible relación con Asterisk, o definiendo el acrónimo en caso que existiese

Bibliografía y Enlaces de Interés



Asterisk: The Definitive Guide

Libro principal en el que se fundamenta este Wiki



Github Asterisk-PFC

Repositorio Github con todo el contenido del PFC que basa esta Wiki



Guía Wikimedia

Guía de edición para Wikimedia



Contacto

Para más información o cualquier necesidad que surja estos son mis medios de contacto

Figura 3.7: Esquema de la sección de Bibliografía y Enlaces de Interés

2. Tabla de Contenidos

En todos los artículos existe un índice para marcar una guía especialmente útil en caso de querer referenciar determinadas posiciones específicas del mismo sin tener que mirar el artículo completo

3. Introducción

Aún llamado de diferentes formas, todos los artículos poseen una primera sección introductoria que sirve para empezar a desarrollar el tema en profundidad empezando a sacar posibles subdivisiones del mismo, o enumerado funcionalidades

4. Contenido Principal

Podría considerarse la masa crítica del Artículo donde se describe todo en detalle.

5. Referencias

Sería semejante al uso de los “pie de página” de cada artículo, las cuales se encuentran dispersas por todo el Contenido Principal con un pequeño enlace que las referencia. Aquí no se ha tomado forzosamente un idioma de referencia dado que se considerarían enlaces externos

6. Véase También

Enlaces a otros artículos de la WIKI, teniendo siempre en mente, que absolutamente todo, tiene alguna relación con otro artículo. La idea de esto, es que fuera viable poder establecer como mínimo un camino de lectura, y poder recorrer el 100 % de los artículos sin tener que visitar la página principal.

7. Enlaces Externos

Todas las empresas, documentos digitales y direcciones web interesantes que pudiera hacer referencia directamente al artículo y cobraran sentido con respecto al mismo. Podría considerarse un tipo de bibliografía añadida sin referencia directa, al artículo conjunta al grupo de **Referencias**.

3.3.1. Consideraciones de Formato y Estructura

En el momento de crear un artículo se diseñaba principalmente aplicando un Artículo de ejemplo:

<http://wikiasterisk.com/index.php?title=Estructura>

El cual contenía la estructura anteriormente comentada en líneas generales, y utilizaba algunas plantillas prácticas para el desarrollo del mismo.

También se creó una categoría específica para englobar todas las Plantillas posibles en caso que pudiera requerirse de su uso:

[http://wikiasterisk.com/index.php?title=Categor %C3 %ADa:Principal:Plantillas](http://wikiasterisk.com/index.php?title=Categor%C3%ADa:Principal:Plantillas)

Dentro de la Estructura existen dos indicadores específicos del estado del Artículo. Para mi criterio les he dado el siguiente uso:

■ To Do

En caso que el Artículo no estuviese terminado, o tuviera intención de ampliarlo en un futuro, se marcaría con esta Plantilla, que además asigna una Categoría especial, para poder sacar en un futuro un listado de todos los artículos marcados y así poder seguir trabajando donde lo dejamos. Se representa con una pequeña caja como podemos ver en la figura 3.8

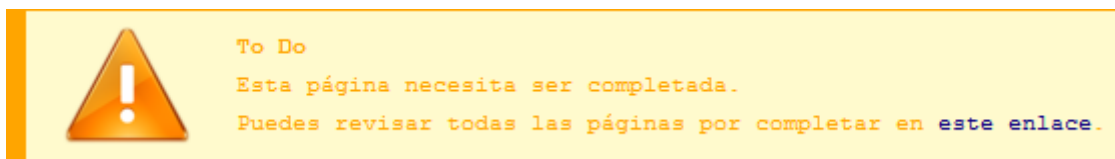


Figura 3.8: Indicador de Artículo pendiente de Completar o Mejorar

■ Formato

Por otro lado, en caso que un Artículo no haya cumplido toda la estructura anteriormente mencionada al completo, incluyendo al menos un enlace de cada tipo, y como nota adicional, al menos una imagen relativa al contexto (para resultar más gráficamente descriptivos de manera general), quedaba marcado con una caja representativa indicando que le falta cumplir las reglas de Formato preestablecidas. La representación de la misma puede verse en la Figura 3.9

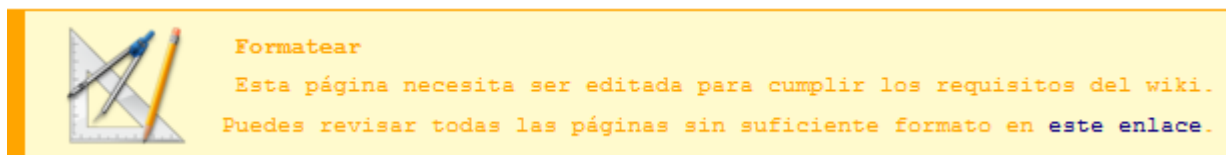


Figura 3.9: Indicador de Artículo pendiente Formatear

Capítulo 4

Desarrollo del proyecto

4.1. Elección del Servidor

A la hora de plantearse que Servidor va a ser más adecuado para nuestro sistema aunque tenemos múltiples opciones, hay que tener presentes varios aspectos. Hay que considerar que aunque Asterisk, sea ampliamente reconocido por funcionar en cualquier sistema, dado que realmente se trata de un software que funciona bajo un sistema de tipo *NIX, el rendimiento no depende tanto en exclusiva de las prestaciones de la máquina sino de las posible incidencias colaterales, y a su vez de ciertos aspectos que deben preservar un sistema de funcionamiento crítico como es el sistema de telefonía en términos generales (Considerando que hasta hoy en día, ni siquiera una persona física puede prescindir un solo día de su teléfono en activo).

4.1.1. Marcas Generalistas

Entre los equipos más populares destacan las marcas Dell y HP por su extensiva y conocida fiabilidad entre la comunidad Asterisk. Es bien conocido uno de los problemas populares surgidos en general con los servidores, dificultades con el sistema de interrupciones IRQ, dado que algunos servidores de alta producción pueden llegar a contener múltiples tarjetas PCI de comunicaciones y hacer un uso intensivo de las mismas. Pero también hay que considerar que hoy en día, gracias a las nuevas tarjetas PCI-x y también la posibilidad de realizar las comunicaciones a través de dispositivos externos al servidor (Gateways) ya no es esto tan crucial. Por eso a nivel de marcas, siguen prevaleciendo estas dos, pero solo por el soporte de calidad que ofrecen.

Considerando la opción Dell por la relación calidad-precio principalmente, lo correcto sería plantearse aun el más bajo de gama, algo con prestaciones exclusivas de servidor. Por un lado la posibilidad de tener fuentes de alimentación redundantes, además de tener dos o cuatro sockets para múltiples procesadores, y una placa base con opción a poner memorias registradas. En otro caso hasta un equipo DIY [\[46\]](#), podría hacerle la competencia a cualquier servidor de Marca como veremos mas adelante.

4.1.2. Consideraciones Técnicas

Otro aspecto interesante, pero no imprescindible, es la capacidad de acceder a una tarjeta específica de las marcas para realizar un sistema RAID por hardware, y opcionalmente, la disponibilidad concreta, de cajas tipo enrackables para adaptarse óptimamente a nuestros respectivos CPD.

Me gustaría ofrecer una opinión personal en términos generales. No merece la pena gastarse el dinero que cuesta un servidor si no vamos a tener las prestaciones especiales que ofrece una placa de servidor y todo lo que le rodea (posibilidad de fuentes redundantes, memorias registradas, dos o cuatro sockets, hotswaps, controladoras RAID de alta gama, etc.). Hoy en día por un bajo coste (entre 1000 y 2000 euros), podemos tener un equipo del nivel de un servidor en prestaciones y verdadera fiabilidad. Con el ahorro tenemos opciones a aplicar redundancia en los sistemas. Más curioso aun, los servidores en formato de caja de torre de las marcas, no aportan un precio mejorado a mayores prestaciones, si bien, no requieren de rack, en caso de poseerlo, es hasta un mayor inconveniente de espacio y desorden. De hecho hoy en día están surgiendo empresas que se dedican a ensamblar ordenadores eligiendo cuidadosamente los componentes más robustos del mercado y con ello implementar sistemas con una tasa muy baja de retorno ¹.

Para entornos de producción pero con necesidades reducidas, quizá la opción más interesante, podría ser tener un servidor de bajo nivel tipo DIY y con opción de registrar todos los terminales en otro servidor paralelo redundante, a través de comunicaciones de datos, en caso de avería.

4.1.3. Servidor de Pruebas del Project

Para este PFC, me planteo un servidor cualquiera de pruebas, y hacer experimentos. Pero me gustaría ir un poco más allá. La primera “práctica” va a ser montar un pequeño servidor de Producción que cumpla las funciones generales que aportan los servidores Asterisk: Servidor de FAX, Sistema IVR, un pequeño Call-Center para dar soporte a la empresa a nivel IT a través de la VoIP y las líneas analógicas que poseeremos, etc. . .

Para ello cuento con un equipo dedicado a efectos de servidor, con los siguientes componentes internos, procesador con 4 núcleos, 8Gb de RAM y 2 discos duros SATA2 (para funciones de RAID), y demás componentes clásicos.

Además, conectada internamente posee una tarjeta de bajo calibre, una Digium Wildcard B410P con 1 puerto FXO y 1 puerto FXS (para acceso telefonía analógica PSTN).

4.2. Elección e instalación del Sistema Operativo en el Servidor

Como he comentado anteriormente, una de las “bondades” de Asterisk es que aun no siendo realmente multiplataforma, debido a que esta diseñado exclusivamente para sistemas *NIX, dado el basto abanico que disponemos para este tipo de sistemas, existen unos más interesantes que otros en cuestiones específicas de optimización de recursos. Hoy en día el 99 % de las instalaciones de Asterisk (y de otros tipos de sistemas servidor, como servidores ftp, web, etc.), se realizan sobre sistemas GNU/Linux. Pero para este apartado, el tema a tocar, sería concretamente la decisión de hacerlo, ¿sobre que distribución?

4.2.1. Valorando Distribuciones Linux

Siguiendo la línea, realmente hoy en día todas las distribuciones son válidas, incluso distribuciones basadas en UNIX, desde Open Solaris, incluso distribuciones para dispositivos con un sistema operativo embebido como OpenWRT [34], y evidentemente, cualquier otra distribución GNU/Linux específica, tipo RedHat, OpenSuse, Fedora y demás.

¹algunas PYME ya reportan con este sistema una reducción significativa de un 0.5 % de RMA sobre el total de equipos ensamblados a medida, basado en comentarios personales con mis contactos principales dentro del sector.

Pero en el mundo Asterisk las distribuciones mas destacadas por la comunidad son CentOS y Debian por igual. De todas formas esto no resulta nada nuevo ya que hoy en día, CentOS y Debian se reparten el pastel de los servidores, incluso por encima de Fedora y RedHat Enterprise.

Debian es para mí personalmente, la distribución que mas he tenido la oportunidad de manejar. Y además entre la comunidad de desarrollo, Debian esta cogiendo cada vez más popularidad, aunque existen ciertas Distribuciones, especificas de Asterisk y cada vez más reconocidas como Elastix [30], o Asterisk NOW! [15] basadas en CentOS.

Pero realmente, existe una distribución dentro del mundo Debian que simplifica masivamente la vida de configuración a nivel Hardware, Ubuntu en su versión especifica para servidores, Ubuntu Server. Es cierto que a día de hoy no es una personificación del máximo rendimiento y la optimización de los recursos del sistema ya que de por si añade un consumo “cabecera” (overhead) a la CPU que no es nada positiva para un sistema con una función especifica, como cumplirá en este caso lo que aquí estamos tratando, un sistema PBX.

4.2.2. La selección personal: Ubuntu Server. Motivos Generales

Pero ahora volviendo atrás al momento de la decisión de configuración de Hardware me encontraba con una simple cuestión: nuestra maquina de pruebas, tampoco iba a servir como un sistema de propósito específico y de alto rendimiento. A efectos prácticos de prueba, no es la intención servir como Call Center para cubrir 1000 llamadas simultáneas. Además Asterisk infraescalado en prestaciones, ni aun con un sistema como Debian en su versión mas ligera, seria capaz de soportar esto. Es por esto quizá, por lo que la facilidad de autoadaptación del Hardware en el sistema fue por ultimo mi decisión final de utilizar Ubuntu Server como sistema.

El instalador, en términos generales, es relativamente sencillo de seguir, y tras instalar Ubuntu Server pude comprobar como prácticamente todo el Hardware fue reconocido, y autoconfigurado, desde la Controladora RAID, hasta la VGA en alta resolución (no es gran cosa, pero personalmente me resulta muy útil trabajar en 1920x1080 incluso en consola si el sistema ofrece esta posibilidad como es el caso. Evidentemente, solo faltaba por reconocer la tarjeta Digium B410P descrita anteriormente.

La segunda parte quizá mas significativa de la configuración del sistema fue la del planteamiento de estabilidad del sistema a nivel de Disco Duro. En este ejemplo disponemos de dos discos duros de semejantes características, por lo que se veía motivada la opción de montar un RAID. Es determinante, que la mayoría de las placas base, a pesar de traer un controlador RAID, realmente no es un controlador puro y dedicado como podrían ser los controladores PERC de Dell. Considero un controlador dedicado, a ese controlador capaz de liberar a la CPU de la carga de realizar todas las tareas para que el RAID se viera satisfecho. En este caso mi intención era montar un RAID 1 (tipo Espejo), para aportar más estabilidad al sistema.

En general la mayoría de las placas base suelen traer controladores Silicon Image, ATI Nvidia, VIA, etc., pero realmente no dejan de ser controladores de tipo software, gestionados desde la BIOS con solo las características básicas para poder proveer de información suficiente de como debería trabajar (pero no capaces de trabajar autónomamente). Esto podría resultar interesante, excepto por una cuestión muy importante: Si esta controladora se averiase (o mas común, si la misma placa se averiase), necesitaríamos para recomponer ese RAID una controladora exactamente igual, o compatible de semejantes características, capaz de recomponer eso. Hoy en día, esto puede resultar excesivamente complejo, ya que la mayoría de las piezas quedan descatalogadas en poco menos de 2 años, y pese a que continua su producción, a partir de los 4 años es prácticamente imposible conseguir una a no ser que la busquemos a través de un Bróker, o que la pidamos a la marca bajo demanda (con un coste 10 veces mas caro de lo que

costó originalmente repercutiendo en definitiva sobre el TCO del sistema severamente). Esta dependencia al Hardware, para mí me resulta inviable. Si tuviéramos la opción de tener controladoras específicas como las PERC de Dell que comentaba antes, que siguen ciertos estándares al menos dentro de la misma marca, y que su producción se extiende más allá de los 5 años, además de ofrecer un hardware dedicado y liberando a la CPU de esta carga, entonces definitivamente sí podría decidir, que merece totalmente la pena.

Entonces, en este caso, teniendo dos discos duros iguales, y la intención es montar un RAID 1, la solución bajo mi criterio más fiable nos la ofrece el propio sistema operativo. GNU/Linux provee la solución, Md Raid, con **mdadm**. El RAID software por excelencia.

4.2.3. Primeros Pasos en el Servidor

Para configurar esto en un sistema de pequeña escala la idea es muy sencilla:

Ubuntu ofrece en el propio instalador la opción de configurar esto de manera gráfica y eventualmente es muy práctico ya que se establece justo desde el momento de la inserción del primer dato del sistema, pero no necesario, ya que mdraid provee de un sistema de sincronización desde el momento de la puesta en marcha. En cualquier caso, por si a alguien le interesara configurarlo en manualmente directamente desde la shell de Linux, el concepto lo describo a continuación:

1. Para cada disco tenemos que crear primero una partición de tipo RAID arrancable (exactamente iguales, ocupando el disco completo, y con el flag B de bootable).

Podríamos utilizar una herramienta de particionado tipo cfdisk. Simplemente, toca, crear una nueva partición con **New**, en Tipo, ponerle **fd** (Particion Linux Raid), en flag, ponerle **bootable**, y finalmente **“Write”** para escribir las modificaciones sobre la partición. Repetir este proceso para el otro disco.

2. En segundo lugar, toca “instalar” el Raid software con mdadm. En el instalador simplemente hay que especificar algunos parámetros, como el tipo de raid (1, mirror, espejado, misma información en los dos discos duros), el número de discos involucrados, el número de discos sobrantes (Spares, como en el sistema JBOD por si acaso) y seleccionar los discos involucrados. Y el sistema raid estará listo en unos segundos.

En el caso shell es quizá aun más sencillo:

Considerando que los discos son SDA y SDB (sata disk A y sata disk B siendo ambos discos tipo sata, si fuera IDE/PATA, sería hda, y hdb)

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1 (el 1 es la partición 1 que creamos anteriormente, una sola partición).
```

En este caso habremos llamado a nuestro recién creado RAID md0 a nivel de sistema.

3. Acto seguido, toca particionar nuestro recién creado RAID. En este caso, sería como si tuviéramos un solo disco duro. Podríamos directamente particionarlo con ReiserFS, Ext3 o Ext4, o podríamos elegir un formato más avanzado como LVM2. Yo personalmente soy más partidario de utilizar LVM por la opción de hacer una copia de seguridad del sistema completo utilizando Snapshots. Un RAID 1 aunque ayuda al uptime del servidor, ya que aunque haya un error de disco, el sistema seguirá en pie, pero no asegura que la información se destruya en un momento determinado (o peor aun, se corrompa). La única forma de preservar la fiabilidad de la información es utilizar las copias de seguridad. Y los snapshots de LVM son la forma idónea de realizarlas sin tener que parar el sistema, y mantener el 100 % de tiempo de actividad en nuestro sistema.

Sobre la estructura de particionado, existen muchas fórmulas, pero una de las posibles sería la siguiente. En primer lugar, una partición para el sistema de arranque. Esto no es estrictamente necesario pero, realmente es una buena práctica poner el sistema de arranque en un tipo de partición

más sencilla como ext2 que la partición del sistema principal, que en este caso sería ext4. Yo suelo darle bastante cantidad de espacio, del orden de 10 veces más de lo que necesita. Con 50 Mb es suficiente, pero si en un momento determinado vamos cargando múltiples kernels para cualquier asunto, o actualizaciones sucesivas, y no tener la necesidad de ir borrando lo antiguo, suelo poner unos 500 Mb. Con los discos duros de hoy en día no supone un gasto indebido y nos curamos en salud

En segundo lugar el espacio Swap. Hay que considerar una cosa: Si tenemos que utilizar el sistema Swap en nuestra maquina Asterisk, vamos a tener serios problemas. Yo personalmente me plantearía elegir dedicar 0 Mb. Pero con mi política de reservar espacio para todo y que en un momento dado, aunque el sistema vaya mal, al menos no vaya a pique con facilidad, me gusta reservar tanta memoria Swap como memoria RAM disponible haya.

El resto, volumen principal, en ext4 que es el más reciente en estos momentos, y cumple su función. Los sistemas Ubuntu lo permiten y Asterisk funciona perfectamente en este tipo de sistemas.

4.3. Virtualización para Asterisk

Generalmente, siempre que sea posible evitarlo, no es una gran idea Virtualizar Asterisk, por la principal deficiencia que presentan todos los sistemas de Virtualización: la escasa optimización de los recursos por parte de los sistemas basados en esta idea. Se suele optar por sistemas “sobre el metal” (bare-metal) a la hora de diseñar plataformas Asterisk puesto que es la forma de obtener el máximo exponencial dedicado para su uso.

En contrapartida, la virtualización, ofrece grandes bondades que en ciertos entornos pueden superar ampliamente, a la principal deficiencia. La característica más positiva que nos ofrecen, es la capacidad de restablecer un sistema en tiempo record y que simultáneamente, es un símbolo de protección por su efecto de back-up.

4.3.1. Valorando Opciones de Virtualizacion

Existen múltiples sistemas de virtualización, pero los mas reconocidos, en el mundo del software libre son tres: OpenVZ, Xen, y KVM.

OpenVZ surge como evolución del sistema Virtuozzo, exténsamente conocido en el mundo de los Alojamiento web. La principal ventaja de OpenVZ es que se integra con el sistema host perfectamente, y una de las limitaciones es que tanto el host como el cliente han de ser sistemas GNU/Linux (esa es la principal razón de esta magnifica integración).

Por otro lado, Xen, nacido en 2003, es un sistema de virtualización basado en máquinas cuyos sistemas operativos están diseñados específicamente para trabajar sobre él. Ha habido múltiples evoluciones en lo que este requerimiento se ha flexibilizado bastante (específicamente, para sistemas operativos cuyo código no es abierto como Microsoft Windows, para poder “saltar” esta barrera original). Se convirtió en la solución de código libre mas distendida en términos de virtualización.

En última instancia surgió KVM como nuevo competidor, a raíz de que el Kernel de Linux (versión 2.6.20 año 2007) permitía por primera vez utilizar las extensiones Intel-VT y AMD-V de virtualización. Pretendía que entrar en el mercado dominado por VMWare en el mundo propietario, y Xen en el mundo Open Source. Por otro lado también emergían nuevas soluciones como Virtual Box de Sun Microsystems, que a posteriori sería absorbida por Oracle.

4.3.2. Ventajas de la virtualización con KVM

En el origen surgían varios inconvenientes, era necesario tener un procesador con soporte para virtualización por hardware, como son los procesadores con tecnología AMD-V e Intel-VT, el número de seguidores evidentemente era limitado, así como el apoyo económico y tecnológico que ofrecían las grandes marcas (RH, Novell, Citrix...) y además surgió exclusivamente como un sistema basado en línea de comandos, bastante complejo, sin posibilidades de adaptarlo a interfaces gráficas de forma relativamente aceptable.

Pero hoy en día todo esto quedó atrás. Por un lado, casi todos los procesadores tienen la capacidad de la virtualización basada en hardware. A lo mejor máquinas antiguas sufren esto, pero ya es realmente raro mantener un servidor de tal antigüedad con semejantes características, y ni siquiera plantearse el hecho de incorporar las máquinas virtuales en él.

Es curioso como la curva de seguidores, en una balanza, lleva tiempo inclinándose a favor de KVM. Desde la lectura de estas líneas KVM ha superado ampliamente a la comunidad de XEN, de hecho XEN cada vez pierde mas “seguidores” a costa de KVM como puede verse en la mayoría de las estadísticas².

Sobre los apoyos tecnológicos, económicos, y técnicos, la verdad es que KVM empezó a gozar del apoyo de uno de los principales actores en el ámbito GNU/Linux, RedHat, y desde entonces empieza a hacerle frente incluso a VMWare (aunque todavía dista años luz de hacerle frente en la cuota de mercado).

Y conjunto a esto, también han surgido las interfaces gráficas para dar un apoyo definitivo e impulso al sistema KVM. También surgieron distribuciones específicas de Linux, como Proxmox, que implementan eficazmente, máquinas virtuales KVM y las dotan de complementos muy interesantes para facilitar la vida al gestor.

Hay que decir para terminar que KVM es de los pocos sistemas de virtualización que soporta el mecanismo PCI Passthrough (paso directo), esto ofrece, la opción de poder soportar tarjetas PCI “exóticas”, como tarjetas de sonido, o lo que aquí nos atañe, tarjetas del sistema Asterisk, de forma nativa, sin tener que manipular la información que circula entre ellas y la máquina. Con esto, a Asterisk se le ofrece la capacidad de obtener una fuente de sincronización pura, y podrían verse las funciones que hasta la fecha eran imposibles o demasiado complejas para verse implementadas directamente en máquinas virtuales.

Concluyendo, para este PFC, se ofrecerá una máquina virtual de ejemplo, conteniendo un sistema Asterisk y toda su funcionalidad, aprovechando todo lo descrito hasta aquí, y esta será creada a partir del sistema KVM.

4.4. Instalación de la Máquina Virtual

Como he comentado, vamos a utilizar el sistema KVM sobre nuestro Ubuntu instalado (podría instalarse en cualquier sistema de hecho, y aplicarse la MV para cualquier entorno dado). A continuación describo el proceso de configuración del mismo:

En el caso de haber elegido Ubuntu, como es el caso aquí descrito, existe la posibilidad de realizar la instalación completa de KVM a través de tasksel, con la opción Virtual Machine Host, que engloba todos los paquetes apt asociados a este sistema de virtualización.

Los paquetes principales son:

1. **QEMU-KVM:** Los sistemas de virtualización KVM basado en QEMU(forma parte del proceso de virtualización de KVM)

²Ejemplo, Google Trends, <http://www.google.es/trends/?q=xen,+kvm,+openvz&ctab=0&geo=all&date=ytd&sort=1>

2. **Libvirt-bin:** Librerías encargadas de administrar KVM y QEMU
3. **Ubuntu-VM-Builder:** Herramienta de línea de comandos para administrar maquinas virtuales basadas en KVM
4. **Bridge-Utills:** Proporciona la conexión “puente” entre la red que proviene del sistema host y las maquinas virtuales. Este es el clásico método de compartir la conexión de forma autónoma y aislada para las maquinas virtuales, en el mundo de la virtualización.

El sistema de conexión puente, aun así no funciona por defecto en la instalación de KVM, y requiere de QEMU para operar, y para la instalación de nuestra maquina Asterisk será importante que tenga un acceso independiente. Para ello tenemos que instalar más paquetes:

1. **QEMU:** Sistema de virtualización
2. **Libcap2-bin:** Librerías de las herramientas de capacidades de Linux, setcap, sucap, etc...

Y con el siguiente comando adquirimos la propiedad CAP_NET_ADMIN necesaria para esta funcionalidad de acceso puente:

setcap cap_net_admin=eip /usr/bin/qemu-system-x86_64

Por otro lado, necesitamos configurar la interfaz de red para que funcione como posible puente para nuestras maquinas virtuales. Dentro del fichero de configuración, en caso de Debian /etc/network/interfaces hay que hacer algunas modificaciones para adaptar la interfaz (eth0 clásicamente) y crear el puente br0. A continuación un posible ejemplo:

```
auto eth0
iface eth0 inet manual
auto br0
iface br0 inet static
    address 192.168.1.200
    netmask 255.255.255.00
    gateway 192.168.1.1
    network 192.168.1.0
    broadcast 192.168.1.255
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
```

Realmente ya no es necesario mucho más que esto a nivel de configuración, a partir de aquí ya solo queda la creación de las máquinas. Existen múltiples métodos, y con el tiempo como comentaba antes, sobre las virtudes de KVM, se han desarrollado no solo interfaces gráficas, sino sistemas para la línea de comandos que facilitan la creación enormemente. A nivel command-line puedo destacar dos principales: Virt-Manager para la creación de maquinas virtuales en general, y Ubuntu-VM-Builder, que nos interesa concretamente para lo que nos atañe con este proyecto.

Con Ubuntu-VM-Builder (vmbuilder) tenemos la posibilidad de generar una nueva maquina virtual, basada en una versión de Ubuntu a nuestra voluntad, y con una instalación totalmente desatendida. Es lo mas parecido a una maquina virtual con OpenVZ que nos podemos encontrar para KVM. Este “asistente” fue diseñado por Canonical específicamente para Ubuntu, he ahí su potencial. Para la creación con este sistema tenemos dos opciones:

1. Mediante la creación de un fichero de texto, que utilizaremos en conjunto al comando para que lo tome por defecto. Esta opción es muy interesante, si tenemos intención de diseñar maquinas nuevas en un futuro, y no queremos pasar por el tedioso estado de tener que releer toda la información específica acerca de esta aplicación. Es de alguna forma, un sistema recordatorio para que en el futuro veamos que parámetros elegimos para nuestras máquinas virtuales

2. Hay otra opción, más rápida, sencilla y directa, y es aplicando el comando con todas sus opciones. Y es la que voy a aplicar para este caso.

Las características que voy buscando en la maquina a crear para este PFC son las siguientes:

1. Maquina basada en Ubuntu, versión 11.10 (Oneiric Ocelot)
2. Arquitectura AMD64
3. Dirección IP: 192.168.1.100
4. Puerta de Enlace: 192.168.1.1
5. Usuario: asterisk
6. Contraseña: asterisk
7. Memoria RAM: 2 Gb
8. Disco Duro: 4,5 Gb (un DVD completo en el que alojare la maquina virtual final)
9. La red ira a través de la interfaz puente que creamos anteriormente

Vamos al directorio donde queremos “instalar” la maquina virtual. Un ejemplo clásico seria el directorio especifico de libvirt: /var/lib/libvirt/images/, supongamos que creamos aquí un directorio llamado **pfc-asterisk** para contener nuestra maquina virtual.

Para definir el particionado del disco virtual a crear, dentro de este directorio vamos a crear un fichero llamado vmbuilder.partition con la siguiente información:

```
root 4608
```

Y por otro lado, para asegurarnos una conexión sencilla mediante acceso SSH, vamos a crear un fichero que se ejecutara en el primer arranque de nuestro sistema, en el mismo directorio también, un fichero llamado boot.sh

```
apt-get update
apt-get install -qqy --force-yes openssh-server
```

Entonces el comando quedaría así:

```
vmbuilder kvm ubuntu --suite=oneiric --flavour=virtual --arch=amd64
--mirror=http://de.archive.ubuntu.com/ubuntu -o --libvirt=qemu:///system
--ip=192.168.1.100 --gw=192.168.1.1 --part=vmbuilder.partition
--user=asterisk --name=Asterisk --pass=asterisk --addpkg=unattended-upgrades
--addpkg=acpid --addpkg=aptitude
--firstboot=/var/lib/libvirt/images/pfc-asterisk/boot.sh --mem=2048
--hostname=pfc-asterisk --bridge=br0
```

Con el comando **vmbuilder kvm ubuntu --help** tenemos acceso al significado de cada uno de los parámetros aquí utilizados aunque

Una vez terminado el proceso, que no debe tardar demasiado, ya solo queda arrancar el sistema, a través de la consola de maquinas virtuales, Virsh.

```
# virsh --connect qemu:///system
virsh # list --all
virsh # start pfc-asterisk
```

Desde este momento, y pasado el rato que termina el proceso de arranque de Ubuntu, podríamos acceder con cualquier cliente SSH a nuestra maquina virtual apuntando a la dirección que habíamos definido en las opciones del ejemplo: 192.168.1.100, y debería conectar sin ningún problema.

El fichero de configuración con toda la información con la que hemos diseñado esta maquina virtual (y que podríamos editar a voluntad), se encuentra ubicado en la ruta, /etc/libvirt/qemu/pfc-asterisk.xml

Este fichero es muy práctico, dado que es el fichero que nos facilita la portabilidad para mover nuestra máquina (junto evidentemente, al disco duro virtual ubicado en la anterior ruta) a otra maquina física, e instalando todo el sistema KVM, tenerla activa en solo unos minutos, con total abstracción de los drivers específicos de la nueva máquina. He aquí donde realmente radica el verdadero potencial de las maquinas virtuales.

Tenemos múltiples opciones para trabajar con las distintas aplicaciones de manejo y gestión de maquinas virtuales, y con la creación de las mismas concretamente basadas en KVM-QEMU, pero este resumen es en líneas generales, el procedimiento seguido por mi parte enfocando en el propósito específico de este proyecto.

```
asterisk@pfc-asterisk:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        4,3G  637M  3,4G  16% /
udev            1001M   4,0K 1001M   1% /dev
tmpfs            402M   188K  402M   1% /run
none              5,0M     0   5,0M   0% /run/lock
none            1005M     0 1005M   0% /run/shm
```

Figura 4.1: Capacidad inicial del disco duro en la Maquina Virtual

Capítulo 5

Caso de Estudio: UCA Autos

Con todo el sistema, preparado para empezar a trabajar con Asterisk, a continuación presento un ejemplo como caso de uso para una empresa ficticia, la cual tiene un enorme interés en implementar un sistema Asterisk, dadas las características que este le aportará para su negocio.

5.1. Introducción

Don Zutano Doe, gerente, y propietario de un nuevo concesionario de automóviles llamado UCA Autos, basado en una prestigiosa marca a nivel nacional, es un gran aficionado a todos los adelantos tecnológicos y piensa que estos influirán en gran medida en el amplio desarrollo de su negocio. A través de un compañero de su club de tenis, que trabaja en una empresa de telefonía, le ha dado a conocer un nuevo sistema de comunicaciones que le podría aportar una serie de funcionalidades que le resultaron bastante interesantes para implementar en su nuevo negocio. Gracias a este contacto, tuvo la oportunidad de establecer relaciones con una modesta empresa local, con cierta experiencia en este ámbito.

En la primera reunión con el responsable de proyectos, el Sr. Doe pudo transmitir sus pretensiones iniciales. Dada que la estructura de la empresa aun estaba por ser determinada, de momento solo necesitaba funcionalidades básicas para cubrir los aspectos fundamentales de servicios primarios a nivel de comunicaciones para el negocio.

A priori la estructura departamental estaba definida, y había un listado de personal para poder definir una primera instantánea de cómo debería funcionar todo.

El responsable de proyecto pudo describirlo de la siguiente forma.

Existían 6 departamentos:

1. Departamento Administración
2. Departamento Comercial
3. Departamento de Almacén y Logística
4. Departamento de Postventa y Servicio
5. Departamento de Marketing y Calidad

Cada departamento tenía un responsable, y todos ellos dependían directamente, del Sr. Doe. Dentro de cada departamento existía un número variable de operarios, administrativos, comercial y personal en general que dependían de cada responsable.

Cada departamento debía tener un número de teléfono de contacto, y además, existía un teléfono general cara al público. En total 6 números de teléfono directos.

La empresa todavía no era muy grande, y disponía de un número limitado de personal de los cuales los que realmente necesitaban acceso telefónico, se distribuían de la siguiente forma:

Departamento de Administración: Responsable + 3 administrativos Departamento Comercial: Responsable + 4 comerciales Departamento de Almacén: Responsable + 3 operarios Departamento de Posventa: Responsable + 2 recepcionistas Departamento de Marketing: Responsable + 2 operadores. Más el Gerente

En total 20 usuarios inicialmente

De momento, el Sr. Doe no tenía demasiado clara una posible estructura jerárquica con restricciones de llamadas, y relaciones entre departamentos, así que la primera idea, era tener un sistema de telefonía simple, en el que todos pudieran llamar y recibir llamadas abiertamente, pero no descartaba en un futuro, implementar esas restricciones, e incluso tener algún sistema de control.

5.1.1. Instalación del sistema Asterisk

Dado que el nivel Hardware es algo excesivamente cambiante, voy a obviar las características técnicas del entorno y del servidor, haciendo referencia al apartado de elección de servidor para los aspectos determinantes a la hora de elegir una u otra preferencia.

La instalación del sistema Asterisk, se realizará sobre la versión actual mas estable, la versión 1.8, momento en que escribo esta información. He de determinar, que en la actualidad aun existen múltiples ramas dentro de la elección de versionado para el sistema Asterisk. Principalmente existen 5 fundamentales:

1. Rama extremadamente conservadora, y un Fork de Asterisk del sistema base llamado Asterisk-RSP (Real Solid Patchset), que se basa en la teoría de conservarse en un sistema Asterisk totalmente obsoleto a nivel de funcionalidades emergentes (siempre considerando que Asterisk es una solución de comunicaciones, y no de telefonía exclusivamente). Está fundamentada en la versión 1.4, versión que decidieron era el momento de parar la vorágine de avances, y centrarse en la resolución de deficiencias de seguridad del sistema
2. Rama por necesidad o moderadamente conservadora, basada en la anterior versión de Asterisk, 1.6.2. Realmente en esta situación se encuentran instalaciones de Asterisk con desarrollos a medida que no pueden actualizarse porque el sistema se volvería inestable. Realmente todo sistema Asterisk puro, es recomendable actualizarlo dado que la versión 1.6.2 tiene un soporte limitado.
3. Basada el progreso de Asterisk, en este caso todas las instalaciones basadas en la versión 1.8 la cual trae bastantes mejoras, y simplificaciones a nivel de configuración. Se considera por Digium, la versión estable del sistema
4. Basada en el progreso extremado, instalaciones basadas en Asterisk 1.10 (o como le llaman ahora, Asterisk 10). Realmente son versiones de prueba, y jamás recomendadas para instalaciones en entornos de producción como viene siendo habitual en la mayoría de los entornos de administración de sistemas.
5. Liberada recientemente a la comunidad, una rama similar a Asterisk-RSP pero mantenida por Digium, y basada en las versiones mas recientes del sistema que se encuentran en fase estable. Podría considerarse una versión de Soporte a Largo Plazo, y es denominada Asterisk-Certified. En estos momentos esta empezando a imponerse entre la mayoría de los usuarios profesionales que venían siguiendo por sistema la opción número 3.

Para nuestro cliente, en este caso, vamos a seguir la opción número 5, ya que se trata de un servidor de producción y queremos ofrecerle un nivel 2 de servicio según el acuerdo en la capa de servicio estandarizado ¹ [39]

Para el momento, la versión Certificada mas reciente es la versión 1.8.11-cert1 así que descargaremos las fuentes de la siguiente URL:

Descarga Asterisk Certified 1.8.11-cert

Y seguiremos el método de instalación descrito en el artículo Wiki asociado correspondiente.

Una vez con todo el sistema Asterisk desplegado, nos detenemos a comprobar el consumo de recursos en ese momento determinado para comprobar que podremos realizar el resto del despliegue sin mayor inconveniencia.

Todavía tenemos suficiente espacio para añadir componentes de menor magnitud. Existen dos componentes adicionales que cumplirán funciones locales específicas y se relacionarán con nuestra máquina Asterisk prácticamente desde los inicios de su configuración.

¹Wikipedia, Soporte Técnico. http://es.wikipedia.org/wiki/Soporte_t%C3%A9cnico

```
asterisk@pfc-asterisk:~$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	4,3G	2,7G	1,4G	66%	/
udev	1001M	4,0K	1001M	1%	/dev
tmpfs	402M	192K	402M	1%	/run
none	5,0M	0	5,0M	0%	/run/lock
none	1005M	0	1005M	0%	/run/shm

Figura 5.1: Capacidad del disco duro tras la instalación completa de Asterisk

Se trata del paquete LAMP (Linux + Apache como servidor Web + MySQL como SGBD de nuestra BD + PHP como lenguaje de programación en entorno web que se conjugará con nuestro servidor web para poder ofrecer aplicaciones combinables con Asterisk como veremos mas adelante) y por otro lado algún servidor de Correo como Postfix. Para ello nos valemos de la herramienta **tasksel** de Ubuntu que realiza la instalación completa de la forma más ágil para nuestro propósito. Con estas instalaciones apenas consumimos poco más de 100MB así que seguimos teniendo suficiente espacio para continuar el proceso tranquilamente.

5.1.2. Comprobación: Instalación del sistema Asterisk

Para comprobar que el sistema ha sido instalado y configurado correctamente, simplemente podríamos probar a acceder a la Interfaz en Línea de Comandos de Asterisk:

```
# asterisk -r
```

Y a continuación comprobar que todo funciona bien lanzando un comando CLI para recargar toda la configuración:

```
CLI> core reload
```

En caso que no aparezcan errores significa que todo esta bien. Si no hemos emplazado todos los ficheros de configuración tales como **asterisk.conf** o **indications.conf** es probable que salga algún error, por falta de los mismos. Para ello deberemos volver a repasar la guía de Instalación y Configuración inicial provista en la WIKI de Asterisk.

5.1.3. Configurando equipos y el entorno SIP

Ahora daríamos paso a la configuración en primera instancia de los requerimientos solicitados por el Sr. Doe en función de la estructura departamental.

Vamos a considerar, que todavía el presupuesto es demasiado ajustado para la empresa del Sr. Doe como para andar comprando teléfonos que operen con el protocolo SIP [19] (conocidos popularmente como Teléfonos VoIP). Pero hemos comprado una remesa de Auriculares con micrófono integrado para cada una de las extensiones a configurar, que serán adaptados al cada PC de cada usuario y funcionales, utilizando un software capaz de transmitir audio e inicializar sesión en nuestra máquina Asterisk utilizando el protocolo SIP.

Creemos en primer lugar el fichero de configuración dentro de /etc/asterisk/ llamado sip.conf donde añadiremos la configuración específica de cada una de las 20 extensiones solicitadas por el gerente:

/etc/asterisk/sip.conf

```
[general]
; Nuestro idioma
language = es
; Cambiamos el puerto para ofrecer seguridad por ocultacion
bindport = 35060
allowguest = no
```

En primer lugar para la creación de este fichero, hemos seguido las reglas básicas de configuración para el protocolo SIP como podemos ver en el correspondiente artículo de la Wiki.² Significativamente podemos observar que la estructura que hemos seguido para denominar las extensiones, aun no siendo demasiado recomendada, puede verse en la figura 5.2

	Administracion	Comercial	Almacen	Postventa	Marketing
Mánager	11	21	31	41	51
Resto	21	22	32	42	52
	31	23	33	43	53
	41	24	34		
Gerente	10	25			

Figura 5.2: Estructura Inicial de Extensiones de Usuario

Las extensiones constarán de dos dígitos. El primero se encarga de definir el departamento, y el segundo, el puesto. La extensión 10, como extensión especial para definir concretamente la del Sr. Doe, el resto, si el número para designar el puesto es el número 1, correspondería al Responsable de Sección, y los sucesivos dígitos, corresponden al resto de los empleados. Además hemos creado cuatro plantillas para un cometido concreto:

/etc/asterisk/sip.conf

```
[telefonos] (!)
```

²SIP. <http://wikiasterisk.com/index.php?title=SIP>

```

; Vamos a impedir el Transcoding utilizando solo el codec Alaw
disallow = all
allow = alaw
dtmfmode = rfc2833
host = dynamic
qualify = yes
type = friend

[gerencia](!)
context = gerencia

[manager](!)
context = manager

[resto](!)
context = extensiones

```

La primera, teléfonos, es una plantilla que engloba todos los parámetros comunes a las extensiones de la centralita. En este caso, nos interesa trabajar en exclusiva con el códec Alaw (G.711 EU), dado que tenemos una conexión bastante buena, pero nuestra máquina no es demasiado potente, además no queremos invertir en códecs licenciados.

Las tres siguientes plantillas, hacen referencia al “nivel” de los usuarios, y en función de este, los mandaremos a un contexto u otro del Dialplan que construiremos a continuación. A priori, el señor Zutano no nos hizo referencia concreta, sobre la posibilidad de que unos usuarios pudiera hacer cosas que otros no, pero anticipándonos por nuestra experiencia, empezamos a “habilitar” esta funcionalidad.

El sistema de contraseñas, teniendo a lo más básico, aun siendo una mala práctica, hemos utilizado la contraseña por defecto 1234 para todas las extensiones, pero al menos encubierto bajo contraseñas de algo mas de seguridad con un patrón MD5, para mostrar algo más de compromiso.

Finalmente y volviéndonos a adelantar a una probable futura necesidad, hemos asociado a cada extensión su correspondiente buzón de voz, que configuraremos a voluntad si surge el caso más adelante.

Pongo un ejemplo de varias extensiones “que cumplan estas condiciones”:

/etc/asterisk/sip.conf

```

; Gerente
[10] (telefonos,gerencia)
md5secret=2b45660a7b1155943f0132f05bd0e34d
mailbox = 0@admin

; Responsable Administracion
[11] (telefonos,manager)
md5secret=6acb3eb4a4939b22ea6709e87ec79311
mailbox = 1@admin

[12] (telefonos,resto)
md5secret=bb24a2de8a36fdd14e358493cd5abac5
mailbox = 2@admin

```

Ahora pasamos a comprobar que todo ha ido bien. Accedemos a la CLI de Asterisk y con el comando:

```
CLI> sip reload
```


Vemos como nuestra configuración carga adecuadamente. Ahora vamos a comprobar que los pares han sido configurados correctamente. Observando el despliegue de sistemas de la empresa UCA Autos, comprobamos que todos los puestos trabajan en un sistema operativo de Microsoft. En este caso decidimos que vamos a utilizar un teléfono por software, dado que el Sr. Doe de momento no se ha decidido en el modelo de teléfono físico que quiere comprar. El Softphone que hemos elegido por su reconocida estabilidad y calidad es uno gratuito, pero no libre, llamado X-Lite. Tenemos constancia de múltiples Softphone Open Source, pero bajo este sistema operativo, no tienen demasiada estabilidad y pueden complicarnos la vida.

5.1.4. Comprobación: Configurando equipos y el entorno SIP

Configurando la extensión del Sr.Doe de ejemplo como podemos ver en la siguiente Figura 5.3, captura de pantalla de X-Lite.

Comprobamos que se registra correctamente:

```
-- Registered SIP '10' at 192.168.1.200:16992
CLI> sip show peers
Name/username Host          Dyn Forcerport ACL Port  Status
10/10         192.168.1.200 D           N   16992 OK (3 ms)
```

Así que ya podemos proceder a registrar las 20 extensiones, en los 20 equipos de nuestros usuarios y ya estarían listos para empezar a operar con nuestra primera instancia de sistema Asterisk.

SIP Account

Account | Voicemail | Topology | Presence | Transport | Advanced

Account name:

Protocol:

Allow this account for

☒ Call

☒ IM / Presence

User Details

* User ID:

* Domain:

Password:

Display name:

Authorization name:

Domain Proxy

☒ Register with domain and receive calls

Send outbound via:

☒ Domain

☐ Proxy Address:

Dial plan:

Figura 5.3: Configuración de una Extensión con el Programa X-Lite v4.1

5.1.5. Configuración del Primer Plan de Marcación

Ahora los teléfonos no pueden hacer casi nada, excepto llamar a una extensión si saben la ruta exacta, y sin pasar por nuestro sistema Asterisk. Nos interesa que por lo menos puedan realizar llamadas entre ellos marcando el número de extensión según la estructura que definimos anteriormente. Para ello vamos a crear el fichero de configuración del Dial Plan según veremos a continuación:

```
/etc/asterisk/extensions.conf
; --- Plan General de Extensiones
```

```
[extensiones]

exten => 10,1,NoOp()
same => n,Dial(SIP/10)

exten => 11,1,NoOp()
same => n,Dial(SIP/11)

exten => 12,1,NoOp()
same => n,Dial(SIP/12)
```

En primer lugar, creamos números de marcación específicos para todas las extensiones, he puesto las extensiones del ejemplo para continuar con la misma línea

/etc/asterisk/extensions.conf

```
; --- Contexto especifico de los Managers

[manager]

include => extensiones
```

Dado que de momento no existe nada especial para los Mánager, definimos un contexto que apunte al resto de las extensiones, como si fueran extensiones normales

/etc/asterisk/extensions.conf

```
; --- Contexto especifico del Gerente

[gerencia]

include => manager

; --- Otros Sistemas Adicionales de Prueba, solo puede acceder el Gerente.

exten => 1111,1,NoOp()
same => n,Progress()
same => n,MusicOnHold()

exten => 2222,1,NoOp()
same => n,Answer()
same => n,Set(CHANNEL(tonezone)=starwars)
same => n,Dial(Local/10,tT,30)
```

Aquí especificamos un contexto para el gerente, que incluye de forma jerarquica todas las posibilidades que pudiesen tener los manager, y a su vez algunas extensiones de ejemplo, como la capacidad de escuchar la música en espera por defecto, o de llamarse a si mismo para comprobar que todo va bien.

Es importante considerar, que no hemos creado extensiones de marcación dentro del plan para cada una de los dispositivos SIP ya que como veremos adelante, este proceso se simplifica bastante utilizando Macros, como un sistema algo más avanzado del Dialplan de Asterisk.

Para recargar el Plan de Marcación y poder hacer uso de este desde la CLI de Asterisk:

```
CLI> dialplan reload
```

En este fichero vemos algunas cosas peculiares:

La aplicación NoOp sirve para hacer depuración en consola. Si pusiéramos mensajes dentro del paréntesis, se verían en el log del CLI. Se pueden poner incluso variables del sistema, la cuestión es que esto viene más explicando en el apartado correspondiente de la documentación.

Lo verdaderamente significativo es la estructura. Como podemos ver, existen tres contextos (se encuentran dentro de corchetes). Como recordaremos, en el fichero sip.conf anterior, estos hacen referencia a los contextos del plan de marcación donde las marcaciones que realizan los dispositivos irían. En este caso, podemos ver como el contexto [gerente] tiene un parámetro **include** que sirve para ‘incluir’ todas las extensiones del contexto al que apunta el mismo. En este caso, incluiría concretamente por completo el contexto [manager], y este también a su vez incluye el contexto [extensiones]. Esto significa que el gerente tiene “visibilidad” total sobre todo el plan de marcación del sistema, en cambio un dispositivo SIP de un empleado normal, solo podría acceder al plan de marcación específico dentro de su contexto “extensiones”, pero no del contexto del gerente.

Cuando se lo expliquemos al gerente, seguro que se le ocurren aplicaciones prácticas como el hecho de evitar que sus empleados llamen a números internacionales, o de tarificación especial.

5.1.6. Comprobación: Configuración del Primer Plan de Marcación

Esta parte es muy sencilla de comprobar su correcto funcionamiento.

Ahora utilizando nuestro Softphone X-Lite podríamos marcar la extensión 11 y el teléfono empezaría a sonar. Al descolgar se establecería la conexión.

Es posible que esta no se establezca. Para ello debemos utilizar el interfaz CLI para comprobar si el transcurso de la llamada esta produciendose de manera correcta. La salida que debemos ver en este flujo de información en detalle deber ser similar a la que mostramos en la siguiente Figura 5.4

```
== Using SIP RTP CoS mark 5
-- Executing [10@manager:1] Dial("SIP/41-00000082", "SIP/10") in new stack
== Everyone is busy/congested at this time (1:0/0/1)
-- Auto fallthrough, channel 'SIP/41-00000082' status is 'CHANUNAVAIL'
-- Unregistered SIP '21'
-- Registered SIP '10' at 192.168.101.100:19778
> Saved useragent "X-Lite 4 release 4.1 stamp 63214" for peer 10
== Using SIP RTP CoS mark 5
-- Executing [10@manager:1] Dial("SIP/41-00000083", "SIP/10") in new stack
== Using SIP RTP CoS mark 5
-- Called SIP/10
-- SIP/10-00000084 is ringing
== Spawn extension (manager, 10, 1) exited non-zero on 'SIP/41-00000083'
pfc-asterisk*CLI> █
```

Figura 5.4: Ejemplo de una llamada en formato CLI

5.1.7. Introduciendo al proveedor de Telefonía

El último requisito que puso el gerente, es la posibilidad de tener 6 números de marcación directa, uno para cada departamento, y otro para él. Para poder conseguir esto, necesitamos contactar con un proveedor que nos ofrezca esto.

Como siempre, tenemos múltiples alternativas, pero si queremos evitar tener que comprar dispositivos, la más obvia es contactar con proveedores de Telefonía IP. Así que nos pusimos en contacto con una empresa muy conocida de la provincia, llamada **UCA Telecom**. Comparando los planes de precios que nos ofrecía "FonicaTele", el proveedor más conocido a nivel nacional, teníamos una oferta mucho más competitiva y encima, no teníamos la necesidad de andar contratando planes mensuales, ni comprando esos dispositivos que comentaba antes, para interconectar con la Red de Telefonía Conmutada clásica.

/etc/asterisk/sip.conf

```
[general]
register => ucaautos:12345678@home-asterisk.local:35060

[proveedorip]
defaultuser = ucaautos
fromuser = ucaautos
secret = 12345678
host = home-asterisk.local
port = 35060
type = friend
insecure = invite
qualify = yes
context = entrantes
```

Un dato muy relevante en la configuración de nuestro proveedor, es el hecho que el tipo es **friend** y no peer. Si nuestra intención fuera emitir llamadas a través de mismo, lo correcto sería esto. Pero como además, vamos a recibir llamadas entrantes dado que nos va a ofrecer un servicio de números Direct Dial-In (DDI), lo correcto es considerarlo, bidireccional con este tipo en concreto.

Pero de hecho, hay un parámetro que puede resultar aun más curioso de observar, que es el **insecure=invite**. Tenemos un problema si trabajamos con un dispositivo tipo friend: la autenticación bidireccional. Observamos que en la primera línea hay un mensaje register, para registrarnos en el proveedor ip. El tema es que el proveedor IP supuestamente, ha de autenticarse luego con nosotros cuando intenta enviarnos la llamada directa al DDI, pero nosotros al solo tener un dispositivo configurado, no podemos ofrecerle unas credenciales específicas (que probablemente sean las mismas que las de llamadas salientes aunque no tiene porque ser así). Al habernos autenticado primero con el mensaje register, al poner el parámetro insecure de esta forma, como la conexión ya esta hecha (el primer mensaje REGISTER ya se envió), los sucesivos INVITE, no tendrían la necesidad de volver a autenticarse. Si quitamos el insecure, y activamos el debug de SIP, veremos como hay un error en la autenticación entrante, cuando tratamos de llamar a un número DDI.

Lo correcto en este caso, sería convertir este par friend, en un par tipo peer. Y por otro lado, crear un par tipo user, con el usuario de autenticación de entrada del proveedor, en caso que sea otro diferente (como es este caso). Con esto, ya no haría falta tener el parámetro insecure=invite activado, dado que todos los mensajes INVITE se autenticarían con el nuevo dispositivo tipo user, correctamente. Como este caso es relativamente raro (que los parámetros de autenticación de entrada y de salida sean diferentes para nuestro proveedor IP), entonces voy a dejarlo así, dando un voto de confianza a nuestro operador que a final de cuentas, es quien nos va a facturar a final de mes.

Introduciendo los siguientes cambios, obtenemos el resultado correcto de autenticación sin tener que exponer la máquina con el inseguridad de permitir mensajes INVITE sin autenticación:

/etc/asterisk/sip.conf

```
[proveedorip]
defaultuser = ucaautos
fromuser = ucaautos
secret = 12345678
host = home-asterisk.local
port = 35060
++type = peer
--type = friend
--insecure = invite
qualify = yes
--context = entrantes

++[ucatelecom]
++secret = 12345678
++host = home-asterisk.local
++port = 35060
++type = user
++context = entrantes
```

Considerar que **ucatelecom**, es el usuario al que conecta, la empresa UCAtelecom, y la contraseña casualmente es la misma que usamos para conectarnos a ellos casualmente, pero tampoco tendría porque ser así.

Por otro lado, hemos creado un contexto nuevo en el Dialplan, llamado [entrantes] haciendo referencia al lugar donde apunta nuestro nuevo proveedor SIP. Ahí vamos a definir todos los números DDI que nos ha solicitado el gerente, y reenviarlos con la función GoTo al contexto, extensión y prioridad correspondientes de cada departamento.

/etc/asterisk/extensions.conf

```
[entrantes]

exten => 956001101,1,Goto(extensiones,10,1)
exten => 956001111,1,Goto(extensiones,11,1)
exten => 956001121,1,Goto(extensiones,21,1)
exten => 956001131,1,Goto(extensiones,31,1)
exten => 956001141,1,Goto(extensiones,41,1)
exten => 956001151,1,Goto(extensiones,51,1)
```

Y con esto, ya estarían satisfechas las necesidades del gerente, por lo que concertamos una nueva cita con el Sr. Zutano para mostrarle que todo esta correctamente funcionando.

5.1.8. Comprobación: Introduciendo al proveedor de Telefonía

En este apartado tal y como estan definidos los ficheros de configuración todavía es imposible comprobar que todo funciona bien, dado que no tenemos definido en el plan de marcación un metodo para llamar a través del proveedor de telefonía IP.

De momento, para probarlo a modo de ejemplo, podriamos establecer una ruta específica en la configuración específica del contexto del Gerente la siguiente linea:

/etc/asterisk/extensions.conf

```
[gerencia]
exten => 956001121,1,Dial(SIP/proveedorip/956001121)
```

Muy similar a la comprobación del apartado anterior, en este caso considerando por ejemplo, que tenemos configurada la extensión número 21, cuyo número Direct Dial-In (de marcación directa), es el 956001121, si llamamos a este número desde la extensión del gerente (10), debería de empezar a sonar en la extensión 21 el teléfono, dado que gracias a esta línea, estamos mandando llamada a través del par que definimos en el fichero de configuración SIP llamado **proveedorip**, y luego el Proveedor, nos devuelve la llamada a través del mismo par (recordemos que es un tipo **friend**, es decir “permite” tanto la entrada como la salida), y entraría según definido al contexto [entrantes]. En este momento, se redirigiría a la extensión 956001121, que esta a su vez, redirige a la extensión 21, dentro del contexto extensiones.

5.2. Mejorando lo presente

Tras haber realizado los trabajos solicitados, en la siguiente reunión con Zutano Doe, tuvimos la oportunidad de conocer su experiencia. Nosotros sabíamos que cuando viera los resultados querría más, ya que quedaba latente el hecho, que nuestra máquina Asterisk aun tenía mucho potencial por desarrollar.

Uno de los aspectos que notó el Sr. Doe, es que evidentemente, aún no podían cursar llamadas. Era lógico que nosotros no íbamos a ser tan atrevidos como para desarrollar un plan de marcación específico sin antes consultar su opinión dado que aquí estaríamos entrando en el tema económico de facturación, con la compañía elegida UCA Telecom.

Averiguamos que Zutano, mantenía conversaciones con Alemania regularmente, ya que era el su principal proveedor de Vehículos, pero estos temas los trataba el personalmente, y no quería que nadie más pudiera cursar llamadas fuera de nuestro país. Además como es común no se fiaba que sus usuarios pudieran llamar libremente a números de tarificación extraordinaria (80X), o de pago por uso, pero sabía que los números de tarificación adicional, estaban ya demasiado extendidos en las grandes corporaciones, así que no podía de momento evitar su uso (90X). También era consciente, que la mayor parte de los clientes hoy en día, utilizan el móvil como su principal medio de comunicación, así que quería las llamadas a móviles totalmente libres. Un último aspecto que hizo hincapié, es en la posibilidad de que los Mánager pudieran llamar a números de información telefónica (118XX), en caso que cualquiera de sus trabajadores requiriese la información. Le expresamos, que no nos pareció demasiada buena idea, ya que son bastante caros, y hoy en día la mayor parte de esa información esta e Internet, pero aún así, insistió disponer de esto, exclusivo a los Jefe de departamento. Finalmente le hicimos el apunte, que dado la falta de regulación en la actualidad de la telefonía IP, los números de emergencia, podían fallar, por eso aunque habilitaríamos el 112 no se aseguraba su correcto funcionamiento. Tampoco le afectó demasiado, ya que todos sus empleados disponían como mínimo, de su teléfono móvil particular y en el caso de los jefes un móvil de empresa.

Ya que su negocio empezaba a marchar, quería empezar a publicitar un número común genérico, para ofrecérselo a los clientes y poder utilizarlo con fines de Marketing, por ejemplo en sus recientes anuncios en el periódico local. Este teléfono principal, tendría que tener un sistema de interacción con el cliente más personalizado, es decir, la implementación de un IVR (Interactive Voice Response). El Sr. Doe todavía no tenía muy claro como debía operar ya que todavía no estaban sentados los cimientos de la empresa en profundidad. Nosotros le propusimos de momento la siguiente idea:

1. Al entrar la llamada, lanzar un mensaje de Bienvenida
2. A continuación, ofrecer numéricamente las posibilidades de contactar con cada departamento
3. Si no pulsa nada, volver a ofrecerle todas las posibilidades
4. Si sigue sin pulsar nada, entonces dirigirle al departamento de Marketing

Nosotros tenemos un horario de atención al cliente restringido, por tanto sería conveniente, que si nos llaman, fuera de nuestro horario, en vez de aparecer este IVR, apareciera un mensaje que indique que no podemos atenderle, ofreciendo nuestro horario y la posibilidad de dejar un mensaje en un buzón de voz, por ejemplo del Responsable de Marketing. El horario de apertura que nos ofreció el gerente sería de Lunes a Viernes de 9h a 20h ininterrumpidas

Por último, ya que sacamos el tema de los buzones, también estaba interesado en poder disponer de Buzones de Voz para todos su Jefes de Departamento y para él. Todas los mensajes dejados en los contestadores irían a parar a sus correos en cualquier formato de Audio, pero además tendría que existir la posibilidad de ser almacenados en el sistema y escuchados a voluntad por cada uno de los usuarios dentro del sistema. El Sr. Zutano, adelantándose, tenía constancia que Asterisk era de origen americano, así que sugirió, que que si el menú del sistema de buzón de voz estuviese en Español sería un punto muy favorable, ya que no todos sus empleados estaban duchos en el manejo del Inglés.

En principio todo este planteamiento le resulto aceptable y necesario de poner en marcha cuanto antes, aunque seguramente sujeta a modificaciones en un futuro. Siendo realistas, hasta aquí es lo que suelen ofrecer el 99 % de las centralitas de telefonía ofrecidas para cualquier PYME.

5.2.1. Preparando las Llamadas Salientes

Vamos a mejorar el anterior Plan de Marcación existente. A continuación muestro el nuevo código modificado sobre el anterior.

Disecionándolo en primer lugar podemos ver el nuevo contexto [exterior]. Con este contexto especificamos siguiendo los patrones específicos del Plan de Marcación definimos las posibilidades de marcación que podemos hacer para realizar llamadas a números Nacionales (tanto aquellos que empiezan por 8 como por 9), Móviles y números de información (restringiendo exclusivamente a 900, 901 y 902).

/etc/asterisk/extensions.conf

```
[exterior]
; --- Contexto Especifico para Llamadas al exterior

exten => _[89]ZXXXXXXX,1,NoOp()
same => n,Dial(SIP/proveedorrip/${EXTEN})
same => n, Hangup()

exten => _6XXXXXXX,1,NoOp()
same => n,Dial(SIP/proveedorrip/${EXTEN})
same => n, Hangup()

exten => _90[0-2]XXXXXX,1,NoOp()
same => n,Dial(SIP/proveedorrip/${EXTEN})
same => n, Hangup()
```

Por otro lado, dentro de los contextos de Mánager y gerencia, podemos ver también los requisitos específicos a los que solo podrán acceder los mismo como vimos en el primer DialPlan, por el sistema jerárquico creado gracias a la utilización de los “include”. Con estos específicamente, definimos marcaciones para números de información (Aquellos que empiezan por 118), y para el gerente llamadas a números internacionales de destino Alemania, que tienen como peculiaridad, que aunque tienen una marcación común de inicio, a partir del 5 dígito, la numeración es variable considerando a continuación el prefijo regional + el número local (de 3 a 9 dígitos).

/etc/asterisk/extensions.conf

```
; --- Contexto especifico de los Managers
[manager]

include => extensiones

; --- Llamadas a numeros de informacion
exten => _118XX,1,NoOp()
same => n,Dial(SIP/proveedorrip/${EXTEN})
same => n, Hangup()

; --- Contexto especifico del Gerente
[gerencia]

include => manager
include => grabaciones
```

```

; --- Los numeros en Alemania son de longitud variable
; --- Para evitar plantear todas las posibilidades
; --- Hacemos uso del comodin "."

exten => _0049.,1,NoOp()
same => n,Dial(SIP/proveedorrip/${EXTEN})
same => n,Hangup()

```

En siguiente lugar, dentro del contexto [entrantes], hemos añadido un nuevo número Direct Dial-In que redirige a la extensión **start** considerando que sea la genérica para este contexto. En esta extensión vamos a crear nuestra Operadora Automática, tal y como esta definida en el caso de uso. Para ello vamos utilizando una combinación de Aplicaciones según su conveniencia. La idea subyacente es la siguiente y puede verse en la figura 5.5:

1. Ponemos un contador a 0, la idea es que cuando hayamos pasado dos veces por la operadora sin resultado, desviemos la llamada según especificado en el paso número 4
2. Lanzamos el mensaje de bienvenida.
3. Comprobamos si el horario actual coincide con el horario de apertura o no.
4. En caso que no coincida, lanzamos los mensajes de “Cerrado” y a continuación el Buzón de Voz del responsable de Marketing
5. En caso que si coincida, nos vamos a la etiqueta “bienvenida” y a partir de ahí lanzamos el menú contextual para dar la opción a marcar una u otra opción.
6. Si no pulsa nada pasado el mensaje + 1 segundo, desviemos a la extensión “relative timeout”
7. Desde aquí comprobamos cuantas veces hemos pasado por aquí, si solo fue una vez, entonces volvemos al principio del algoritmo
8. En cambio si fueron ya dos veces, mandamos la extensión a Marketing.

/etc/asterisk/extensions.conf

```

; Operadora Automatica
exten => s,1,NoOp()
same => n,Set(ivrcont=0)
same => n,Answer()
same => n(bienvenida),Playback(bienvenida)
same => n,GotoIfTime(09:00-20:00,mon-fri,*,*?abierto)
same => n,Playback(cerrado&horario)
same => n,VoiceMail(1@market)
same => n,Hangup()
same => n(abierto),Background(menu-ivr)
same => n,WaitExten(1)

exten => t,1,NoOp()
same => n,Set(ivrcont=${${ivrcont}+1})
same => n,GoToIf(${${ivrcont}<2}?s,bienvenida:0,1)

; 1 a Ventas
exten => 1,1,Goto(extensiones,21,1)
; 2 a Almacen
exten => 2,1,Goto(extensiones,31,1)
; 3 a Posventa
exten => 3,1,Goto(extensiones,41,1)
; 0 a Marketing
exten => 0,1,Goto(extensiones,51,1)

```

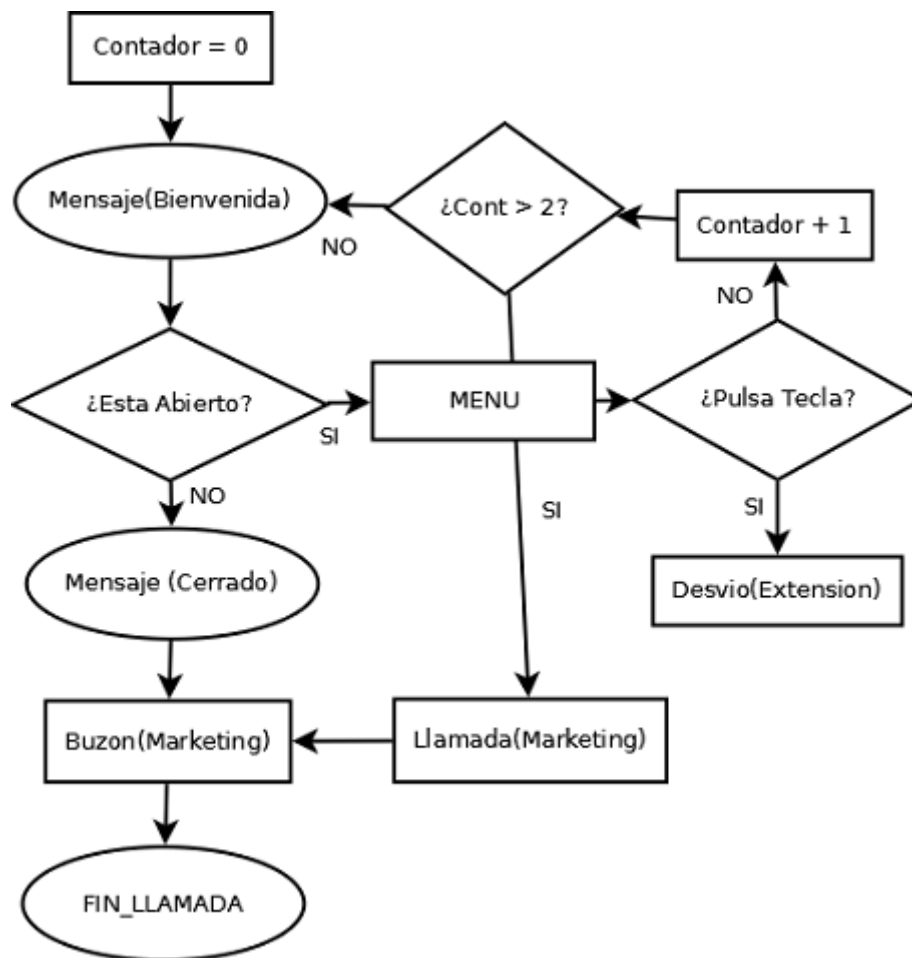


Figura 5.5: Flujo de Llamada de la Operadora Automática

Finalmente, como podía verse anteriormente, hemos modificado todas las extensiones específicas de los Mánager, para adaptarlas al uso de Buzón de Voz. Nótese, que hemos añadido una restricción temporal en la aplicación de llamada **Dial** a 30 segundos, pasados los cuales, se ejecutaría la aplicación de grabación de mensajes de Buzones de Voz. . Por último hemos añadido extensiones específicas para el acceso a los buzones de sus respectivos usuarios, marcando primero el 9 y a continuación el número de la extensión.

/etc/asterisk/extensions.conf

```

; --- Plan General de Extensiones
[extensiones]

include => exterior

exten => 10,1,NoOp()
same => n,Dial(SIP/10,30)
same => n,VoiceMail(0@admin)
; Acceso al menu de Buzon de Voz
exten => 910,1,VoicemailMain(0@admin)
  
```

```

exten => 11,1,NoOp()
same => n,Answer()
same => n,Set(CHANNEL(tonezone)=starwars)
same => n,Dial(SIP/11,30)
same => n,VoiceMail(1@admin)
same => n,Hangup()
; Acceso al menu de Buzon de Voz
exten => 911,1,VoicemailMain(1@admin)

exten => 12,1,NoOp()
same => n,Dial(SIP/12)

```

Dado que la mayor parte de las pistas de audio del sistema Asterisk se encuentran en inglés, y por tanto dificultaría bastante la gestión de los Buzones de Voz por parte de los usuarios, así como determinados mensajes (como los mensajes por defecto para dejar una grabación) para los posibles llamantes en el territorio nacional, hemos tomado la iniciativa de instalar un paquete de voces profesional, publicado por <http://www.voipnovatos.es>, en formato ALAW aún pesados, muy ágil para nuestro sistema. ^{3 4}

La configuración específica del fichero voicemail.conf para que el sistema de Buzones de voz funcione dentro de nuestras pretensiones se detalla a continuación. Para más información podemos consultar el apartado específico dentro del Wiki. ⁵:

/etc/asterisk/voicemail.conf

```

[general]
format = wav49|gsm|wav
serveremail = mcamargo@wikiasterisk.com
attach = yes
maxmsg = 50
maxsecs = 120
minsecs = 5
skipms = 3000
maxsilence = 10
silencethreshold = 128
maxlogins = 2
pbxskip = no
fromstring = VoiceMail Asterisk-PFC
emaildateformat = %A, %B %d, %Y at %r
minpassword = 4
backupdeleted = 20

```

Un dato curioso, la cabecera del fichero voicemail.conf, hace referencia a una generación automática. Esto es debido a que considerando que hemos puesto la contraseña por defecto en muchos buzones, idéntica al número de buzón, por defecto al ejecutar la aplicación VoiceMailMain, solicita el cambio de contraseña. Al realizar este cambio, la aplicación AppVoicemail, hace un cambio en este fichero de forma automática para adaptar las nuevas contraseñas prefijadas. En este caso sería conveniente volver a restablecer las contraseñas a las suyas por defecto como el resto, pero lo dejare en este estado, para que sirva a efectos informativos:

³Sonidos Principales, <http://www.voipnovatos.es/voces/voipnovatos-core-sounds-es-alaw-1.4.tar.gz>

⁴Sonidos Adicionales, <http://www.voipnovatos.es/voces/voipnovatos-extra-sounds-es-alaw-1.4.tar.gz>

⁵Buzones de Voz, Wikiasterisk, http://wikiasterisk.com/index.php?title=Buzones_de_Voz

/etc/asterisk/voicemail.conf

```
;!
;! Automatically generated configuration file
;! Filename: voicemail.conf (/etc/asterisk/voicemail.conf)
;! Generator: AppVoicemail
;! Creation Date: Tue May 22 15:02:20 2012
;!
```

Los datos relativos al mensaje de la zona, suelen ser datos genéricos utilizados en la mayor parte de las instalaciones, basados en las preferencias reconocidas dependiendo la costumbre de la sociedad en cada región del mundo:

/etc/asterisk/voicemail.conf

```
; Parametros por defecto del sistema de mensajes de Asterisk
[zonemessages]
eastern = America/New_York|'vm-received' Q 'digits/at' IMp
central = America/Chicago|'vm-received' Q 'digits/at' IMp
central24 = America/Chicago|'vm-received' q 'digits/at' H N 'hours'
military = Zulu|'vm-received' q 'digits/at' H N 'hours' 'phonetic/z_p'
european = Europe/Madrid|'vm-received' a d b 'digits/at' HM
```

Finalmente para la definición de cada uno de los buzones, hemos elegido algunos de ejemplo para mostrar el nivel de configuración al que pueden llegar a someterse:

/etc/asterisk/voicemail.conf

```
[admin]
0 => 0,John Doe,jdoe@wikiasterisk.com,,tz=european|
dialout=extensiones|sendvoicemail=yes|callback=extensiones|
review=yes|forcename=yes|forcegreetings=yes

1 => 1,Robert Mackencie,rmackencie@wikiasterisk.com,,tz=european|
dialout=extensiones|sendvoicemail=yes|callback=extensiones|
review=yes|forcename=yes|forcegreetings=yes

[ventas]
1 => 1,Hugh Douglas,hdouglas@wikiasterisk.com,,tz=european|
dialout=extensiones|sendvoicemail=yes|callback=extensiones|
review=yes|forcename=yes|forcegreetings=yes
```

Para el envío de e-mail es necesario tener instalado algún servidor de envío de correo. En este caso, durante la instalación gracias a la herramienta tasksel habíamos instalado Postfix, y LAMP que nos serían de gran ayuda en el transcurso de la programación de esta central. En este caso Postfix será el encargado de enviar los mensajes de e-mail a nuestros destinatarios de correo según configuración en el fichero voicemail.conf. Considerar que para el envío de correo es importante que el servidor de correo este bien configurado dado que la mayor parte de los servidores de Internet rechazan correos por doquier, especialmente los mas populares (GMAIL, HOTMAIL, etc). Dentro del fichero /var/log/mail.log podemos ver el resultado del envío en caso que nuestro destinatario no haya recibido el mensaje, y considerar

factores por los cuales nuestro servidor de correo Postfix deba ser mejorado. En este sentido es conveniente que nuestra conexión a Internet este asociada a una dirección IP estática, ya que la mayoría de las direcciones dinámicas han podido ser introducidas en las listas negras como <http://www.spamhaus.org>.

En este caso, dado que no disponemos de IP estática, hemos utilizado un servidor SMTP remoto (smart-host) para configurar nuestro servidor Postfix. Toda la configuración relativa que no atañe directamente a este proyecto podrá verse dentro de la máquina virtual en el fichero principal de configuración del servidor `/etc/postfix/main.cf`. [6]

5.2.2. Resolución: Preparando las Llamadas Salientes

Aunque tuvimos que hacer una modificación en el Plan de Marcación en el apartado anterior para comprobar que las extensiones específicas para llamadas a través del proveedor IP UCATelecom funcionaban, ahora si es posible hacer la prueba directamente y comprobar a través de la interfaz CLI que se efectúa la llamada saliente, y esta regresa entrando por uno de nuestros DDI. En este caso, con cualquier extensión de las creadas en apartados anteriores, probamos a llamar de nuevo al número 956001121. En este caso, si tenemos configurado también la extensión 21, debería empezar a sonar de inmediato.

Podemos comprobar que esto se está efectuando correctamente si obtenemos el resultado de la interfaz CLI como puede verse en la figura 5.6, y en caso que no aparezca correctamente esta salida, podremos observar en qué punto está fallando concretamente.

```
== Using SIP RTP CoS mark 5
-- Executing [956001141@gerencia:1] NoOp("SIP/10-00000085", "") in new stack
-- Executing [956001141@gerencia:2] Dial("SIP/10-00000085", "IAX2/proveedorip/956001141") in new stack
-- Called IAX2/proveedorip/956001141
-- Call accepted by 192.168.101.151 (format ulaw)
-- Format for call is ulaw
-- Accepting AUTHENTICATED call from 192.168.101.151:
> requested format = ulaw,
> requested prefs = (),
> actual format = ulaw,
> host prefs = (),
> priority = mine
-- Executing [956001141@entrantes:1] Goto("IAX2/ucatelecom-7560", "extensiones,41,1") in new stack
-- Goto (extensiones,41,1)
-- Executing [41@extensiones:1] Macro("IAX2/ucatelecom-7560", "telefonos") in new stack
-- Executing [s@macro-telefonos:1] NoOp("IAX2/ucatelecom-7560", "") in new stack
-- Executing [s@macro-telefonos:2] Set("IAX2/ucatelecom-7560", "buzon=41") in new stack
-- Executing [s@macro-telefonos:3] Set("IAX2/ucatelecom-7560", "CHANNEL(tonezone)=starwars") in new stack
-- Executing [s@macro-telefonos:4] Dial("IAX2/ucatelecom-7560", "SIP/41,30") in new stack
```

Figura 5.6: Ejemplo de salida y entrada a través de un Proveedor IP

En el caso de las otras opciones planteadas de llamadas salientes al exterior, simplemente comprobar según el guión que los patrones de las extensiones para el resto de los casos posibles (llamadas a móviles, llamadas a números de información y números extranjeros) coinciden correctamente. Si el proveedor IP ofreciese salida al exterior real, podrían comprobarse en tiempo real, cosa que realmente no es del todo necesario, ya que podemos ver en nuestra interfaz CLI la salida que reporta el hecho de marcar a las posibles combinaciones y poder comprobar también que se están efectuando de manera adecuada al menos en nuestro lado (independientemente de la configuración del Proveedor IP). Un ejemplo de

esto, considerando que el proveedor IP no tiene configuradas las llamadas a números de información (por ejemplo 11811), en la Figura 5.7 podemos observar como en nuestro servidor la salida si se cursa correctamente a través del proveedor, dado que entra en la extensión 118XX y va pasando prioridades, pero el proveedor acaba rechazando la llamada porque no localiza ese patrón en su Plan de Marcación (o podrían ser otros factores, pero eventualmente ya sabemos que al menos la asociación de Patrón desde nuestro número saliente 11811 con el patron _118XX.

```
== Using SIP RTP CoS mark 5
-- Executing [11811@gerencia:1] NoOp("SIP/10-0000008b", "") in new stack
-- Executing [11811@gerencia:2] Dial("SIP/10-0000008b", "IAX2/proveedorip/11811")
-- Called IAX2/proveedorip/11811
-- Hungup 'IAX2/proveedorip-379'
== Everyone is busy/congested at this time (1:0/0/1)
-- Executing [11811@gerencia:3] Hangup("SIP/10-0000008b", "") in new stack
== Spawn extension (gerencia, 11811, 3) exited non-zero on 'SIP/10-0000008b'
```

Figura 5.7: Ejemplo de salida sin retorno a través de un Proveedor IP

Para comprobar que la operadora automática realiza su función el medio más eficaz sería llamando desde cualquier extensión al número Directo, 956001191, y comprobar manualmente que las distintas operaciones segun el esquema seguido en la Figura 5.5 surte efecto.

Finalmente, acerca de los buzones de voz, en el momento que llamamos a cualquier extensión que tenga asociado un buzón de voz, podemos esperar a grabar un mensaje y ver si en la consola se almacena el resultado del mismo, segun vemos en la figura 5.8

Adicionalmente tenemos la posibilidad de ver si existen ficheros de audio almacenados en los buzones concretos, accediendo por ejemplo, al buzón de la extensión 10 desde la ruta **/var/spool/asterisk/voice-mail/default/10/INBOX/**, o de forma genérica:

/var/spool/asterisk/voicemail/<contexto_buzon>/<numero_buzon>/INBOX/ podemos ver según la imagen que aparece un fichero de audio con el nombre msg00001.wav (o el nombre que haya reportado la Interfaz CLI de como quedó registrado).


```

== Using SIP RTP CoS mark 5
-- Called SIP/10
-- SIP/10-00000092 is ringing
-- Got SIP response 486 "Busy Here" back from 192.168.101.100:19778
-- SIP/10-00000092 is busy
== Everyone is busy/congested at this time (1:1/0/0)
-- Executing [s@macro-telefonos:5] GotoIf("SIP/41-00000091", "1?ocupado") in new stack
-- Goto (macro-telefonos,s,8)
-- Executing [s@macro-telefonos:8] VoiceMail("SIP/41-00000091", "10,b") in new stack
-- <SIP/41-00000091> Playing 'vm-theperson.alaw' (language 'es')
-- <SIP/41-00000091> Playing 'digits/1.alaw' (language 'es')
-- <SIP/41-00000091> Playing 'digits/0.alaw' (language 'es')
-- <SIP/41-00000091> Playing 'vm-isonphone.alaw' (language 'es')
-- <SIP/41-00000091> Playing 'vm-intro.alaw' (language 'es')
-- <SIP/41-00000091> Playing 'beep.alaw' (language 'es')
-- Recording the message
-- x=0, open writing: /var/spool/asterisk/voicemail/default/10/tmp/n4bjg6 format: wav49, 0x192d598
-- x=1, open writing: /var/spool/asterisk/voicemail/default/10/tmp/n4bjg6 format: gsm, 0x19d92f8
-- x=2, open writing: /var/spool/asterisk/voicemail/default/10/tmp/n4bjg6 format: wav, 0x1812f18
-- User hung up
== Parsing '/var/spool/asterisk/voicemail/default/10/INBOX/msg0001.txt': == Found

```

Figura 5.8: Ejemplo de salida de un mensaje en el Buzón de Voz

5.3. Interconectando máquinas Asterisk

Una vez terminados los trabajos, volvemos una semana después a UCA Autos para comprobar que todo va bien. Casualmente nos encontramos con el Sr. Zutano el cual nos comenta que esta muy satisfecho con los trabajos realizados, y la evolución que ha supuesto para su empresa haber montado un sistema como este. Nos hace saber que tiene algunos proyectos en mente, pero todavía no se ha dispuesto a llevarlos adelante porque necesita plantear con los Jefes de Departamento sus opciones a ver la opinión de los mismos, y la mejor forma de implementarlos.

Durante la pequeña conversación, nos dejo caer, que en breve seguramente, ampliaría el personal por departamentos, dado que las cosas le estaban yendo muy bien, y cada día sus necesidades aumentaban. Esto sería relevante para nosotros por el hecho de intentar pasar de una central específica a lo más genérica posible para evitar complicarnos en un futuro.

Justamente pocos minutos después, recibimos una llamada de nuestro contacto técnico en UCA Telecom, informándonos que desde hace apenas unas horas han implementado en sus servidores la nueva funcionalidad de interconexión a través de un protocolo específico para sistemas Asterisk llamado IAX. Una alegría para nosotros dado que todas nuestras instalaciones son Asterisk, y eramos conscientes que no estábamos cumpliendo en la mayoría de las mismas a nivel de seguridad, por una cuestión de tiempo y en consecuencia coste de implementación, excepto en algunos clientes que nos lo pidieron expresamente. De hecho el gerente de UCA Autos ya nos hizo un comentario para su mejoría en un futuro, sobre algo que había escuchado de encriptación de las llamadas, para evitar que le pudieran “pinchar” las líneas telefónicas. Ahora con el sistema IAX teníamos la oportunidad de actualizar nuestras instalaciones en general, incluyendo el concesionario.

5.3.1. Acceso al protocolo IAX

Cara a la generalización del sistema, con vistas a una estandarización de las extensiones, hemos decidido dar un vuelco en la estructura de contextos del sistema de Buzones de Voz. Pasaremos todas los buzones tratados al contexto por defecto [default], y además pasaremos a dotar a todas las extensiones del sistema, de su respectivo buzón de voz pese a no ser una petición original, entendimos que se pensó por motivos de recorte presupuestario. Además si queremos implementar en un futuro un sistema de Directorio de extensiones, este paso será bastante conveniente. El procedimiento para el cambio fue el siguiente:

- En el fichero /etc/asterisk/sip.conf actualizamos todos parámetros de los buzones, poniendole nombres equivalentes al nombre del dispositivo (Y número de extension), seguidos del contexto default (@default)
- En el fichero /etc/asterisk/voicemail.conf pasamos todos los buzones al contexto [default] y cambiamos también los nombres siguiendo el sistema anterior

Pero lo realmente interesante, fue el método de estandarización para las extensiones seguido en el fichero referente al Plan de Marcación:

/etc/asterisk/extensions.conf

```
; --- Macro para Extensiones
; --- Con Buzones de Voz Mejorados

[macro-telefonos]
exten => s,1,NoOp()
same => n,Set(buzon=${MACRO_EXTEN})
same => n,Set(CHANNEL(tonezone)=starwars)
same => n,Dial(SIP/${buzon},30)
same => n,Gotoif("${DIALSTATUS}" = "BUSY"?ocupado)
same => n,VoiceMail(${buzon},u)
same => n,Hangup()
same => n(ocupado),VoiceMail(${buzon},b)
same => n,Hangup()

[macro-buzon]
exten => s,1,NoOp()
same => n,Set(buzon=${MACRO_EXTEN:1})
same => n,VoiceMailMain(${buzon})
same => n,Hangup()

; --- Plan General de Extensiones
[extensiones]

include => exterior

; Minimizando las combinaciones para
; ofrecer justo las extensiones disponibles

; Extensiones de Administracion
exten => _1[0-4],1,Macro(telefonos)
exten => _91[0-4],1,Macro(buzon)

; Extensiones de Comercial
exten => _2[1-5],1,Macro(telefonos)
exten => _92[1-5],1,Macro(buzon)
```

```

; Extensiones de Almacen
exten => _3[1-4],1,Macro(telefonos)
exten => _93[1-4],1,Macro(buzon)

; Extensiones de Posventa y Marketing
exten => _[45][1-3],1,Macro(telefonos)
exten => _9[45][1-3],1,Macro(buzon)

```

Aquí introducimos el uso de Macros. Para ello creamos dos Macros básicas observando un poco los patrones que se repetían en el anterior DialPlan y dado que ahorrábamos en trabajo de redundancia en el tecleo de código, aprovechamos para introducir algunas mejoras al sistema.

Por un lado una macro para la llamada a las extensiones, y en caso de no respuesta a los 30 segundos (o comunicando) pasaría a la siguiente aplicación condicional, que mandaría a la aplicación para la grabación de mensajes del buzón de voz, con un parámetro específico u otro (para lanzar el mensaje de ocupado o no disponible en función del resultado de la llamada).

Por otro lado, una macro para acceder al sistema de gestión de cada buzón de voz, pulsando el 9 seguido de la extensión que queramos consultar.

He utilizado patrones para restringir con exactitud, las extensiones solicitadas por el gerente de UCA Autos, y además cuanto más específico sea nuestro plan, mejor nivel de seguridad alcanzamos.

Ahora atendiendo a la llamada de UCA Telecom, vamos a adaptar nuestro sistema para la comunicación a través del protocolo IAX. Nuestro contacto técnico nos planteo la posibilidad de establecer la comunicación cifrada y luego utilizando dos posibilidades más convenientes de autenticación, aplicando un reto MD5 sobre la clave elegida o utilizando claves RSA privadas y publicas (Asterisk solo permite la encriptación DES3, Estándar de Encriptación de Datos Triple). Siguiendo el mecanismo explicado en el apartado de Seguridad de la Wiki ⁶, generamos nuestra clave pública y privada, con el script según hemos visto explicado, y cargamos las claves en nuestro Asterisk.

Adicionalmente, gracias a una interfaz web que nos facilita el proveedor UCA Telecom, somos capaces de introducir en su sistema, nuestra clave pública, y además nos descargamos la clave pública que nos facilitan para introducirla en nuestro directorio de Asterisk por defecto /var/lib/asterisk/keys y así poder cargar las claves en nuestro sistema.

Y con todo ya generado y preparado, creamos nuestro fichero iax.conf en el directorio por defecto de ficheros de configuración de Asterisk /etc/asterisk según podemos ver a continuación:

/etc/asterisk/iax.conf

```

[general]
bindport=4569
srvlookup=yes
language=es
encryption=yes
; Si usamos claves RSA es conveniente desactivar
; esta opcion.
forceencryption=no
trunk=yes
requirecalltoken=yes

```

⁶Encriptando Comunicación IAX, WikiAsterisk,
http://wikiasterisk.com/index.php?title=Seguridad#Encriptando_la_Comunicaci.C3.B3n_con_RSA

En primer lugar, definimos algunos parámetros generales, que destacadamente, hacen referencia al hecho específico de realizar algún tipo de encriptación de la comunicación, y forzar opciones de seguridad más restrictivas.

A continuación planteamos los dos comportamientos por separado, para definir las claves privadas y públicas RSA, de forma independiente, para conectar con el proveedor de telefonía UCA Telecom:

/etc/asterisk/iax.conf

```
[ucatelecom]
type = user
host = dynamic
auth = rsa
inkeys = ucatelecom
context = entrantes

[proveedorip]
type = peer
outkey = ucaautos
username = ucaautos
host = home-asterisk.local
auth = rsa
qualify = yes
```

Por otro lado y para terminar, en el fichero de configuración del Plan de Marcación, solo tenemos que editar todos los puntos donde nos encontramos una comunicación a través del protocolo SIP, y pasarlo por el protocolo IAX2.

5.3.2. Resolución: Acceso al protocolo IAX

En este caso, la comprobación es muy sencilla, dado que realmente solo hemos reformulado lo que ya teníamos hecho anteriormente, pero con un estilo más eficaz y elegante.

Por ejemplo, podemos comprobar el efecto de una llamada a una extensión realizando la misma entre dos teléfonos, y también comprobando si el buzón de voz hace su papel como vimos en el apartado anterior.

En cuanto a la conexión IAX, ocurre exactamente lo mismo que con el proveedor IP a través de SIP. Simplemente llamando a cualquiera de los teléfonos DDI 9560011XX acabado en el número de extensión de las extensiones de Mánager deberíamos de poder recibir la llamada si la tenemos configurada en un teléfono sin problemas

Además es posible comprobar que las claves RSA están correctamente configuradas en el servidor ejecutando el comando en la interfaz CLI:

```
CLI> keys show
Key Name      Type      Status      Sum
-----
ucaautos      PUBLIC    [Loaded]    a18c6e5866db067d9c378fb3cafd967c
ucaautos      PRIVATE  [Loaded]    a46b82f019beb91702a73eec276bf6b6
ucatelecom    PUBLIC    [Loaded]    c3abd8244e94ddb27d02b4b9ed9f9236
```

5.4. Distribuidor de Piezas de Repuesto

Pasado un mes desde nuestra última instalación, pudimos observar como la mayor parte del personal se sentía satisfecha con la solución de telefonía ofrecida. Algunos Mánager de departamento se habían puesto en contacto con el gerente, para consultarle la posibilidad de sacar más partido al sistema. Justamente, en este periodo, se presentó un nuevo reto a la compañía UCA Autos, dado que fue nombrada Distribuidora oficial de piezas de Repuesto para toda la provincia que representaba.

Esto significaba tener que dar servicio a toda la red de talleres, y distribuidores de recambios locales, para la Marca que representaban, pero a nivel provincial, lo que supondría tener que ofrecer un nivel de servicio a sus nuevos clientes. La comunicación se establecía tanto presencialmente, por los gestores de estos negocios, o la vía más común, pedidos a nivel telefónico, dado que aunque existía una interfaz en Internet por la cual podrían cursar sus pedidos, la mayor parte de los clientes, no tenían demasiado control sobre las nuevas tecnologías (gente de relativa avanzada edad, propietarios de talleres acostumbrados a un servicio de toda la vida), y por otro lado, también era frecuente la consulta como atención al cliente, dada la amplia selección de piezas, que causaba relativa confusión entre los compradores.

Es en este momento, cuando el Sr. Doe contactó con nosotros, para ver si existía la posibilidad de montar una pequeña plataforma de Call Center, que fuera fiable, pero a ser posible, no demasiado cara, porque según había entendido por sus compañeros colaboradores, las que ellos conocían resultaban excesivamente costosas, pero ofrecían estadísticas de uso para su análisis de calidad de servicio, y en muchos casos tuvieron que optar por soluciones muy pobres que les aportaban escasa o nula información en la mayor parte de los casos.

Tras la conversación con Zutano, le explicamos que con el sistema Asterisk que había elegido implementar, podría tener el mismo sistema que le comentaron sus compañeros, pero a coste menor aproximado que el que ellos le reportaron, dado que solo se reflejaba el coste de implantación en horas por parte de nuestra empresa. El gerente de UCA Autos, no podía dar crédito a nuestras palabras y nos pidió que lo hiciéramos lo antes posible.

Las especificaciones fueron muy sencillas. Para el departamento de almacén existían tres puestos aparte del Mánager. Quería que esos tres puestos fueran los tres componentes de la primera instancia de Call-Center, ya que de momento no tenía intención de contratar más personal y ver si podía valerse para atender a sus clientes con lo que disponía en la actualidad, o plantear alguna ampliación en un futuro.

Por otro lado, quería tener un teléfono directo específico para el Call-Center, al entrar la llamada, diese un mensaje de bienvenida, y pusiera al cliente en la cola de espera hasta que pudiera ser atendido por un agente. Y si fuera posible también nos pidió que el sistema dijera eventualmente al cliente en espera, su posición en la cola. Le confirmamos que sus peticiones eran todas posibles, y que nos pondríamos manos a la obra de inmediato.

Le preguntamos como quería distribuir sus llamadas entre los agentes disponibles. Hicimos algunos comentarios sobre las estrategias disponibles, y eligió la estrategia más simple de momento, dado que todavía no conocía en profundidad como resultaría esta nueva operación, que sonase por igual el teléfono en todas las líneas disponibles a cada momento.

Finalmente, también le inquietaba, el hecho de poder controlar la calidad de las llamadas. En un principio le comentamos que existía la posibilidad tanto de escuchar las llamadas en tiempo real, como poder realizar grabaciones de llamadas en disco. Ambas opciones le parecían interesantes por igual, especialmente para el Responsable de Almacen y Marketing los cuales debían hacer un seguimiento intensivo por la forma en la que se realizaba la comunicación pudiera optimizarse en un futuro. En consecuencia, se planteo el hecho de tener los dos servicios disponibles

5.4.1. Colas en Asterisk

En primer lugar, crearíamos dentro de la máquina Asterisk, el fichero de configuración específico para el sistema de Colas. Pese a que existen muchos parámetros disponibles nos centraríamos en las peticiones del Sr. Zutano, y si en un futuro apreciaba deficiencias, intentaríamos ajustárselas con las mismas.

El fichero se muestra a continuación:

/etc/asterisk/queues.conf

```
[general]

; Opciones generales practicas
persistentmembers = yes
autofill = yes
monitor-type = MixMonitor

[almacen]

; Estrategia llamar a todos
strategy = ringall
; Intervalo de Servicio para
; estadísticas
servicelevel = 60
; Tiempo de reintento si el
; agente corta la llamada
retry = 5
; Tiempo de "reposicion" del Agente
wrapuptime = 20
; Autopausa si el agente no atiende
; las llamadas
autopause = yes
; Tiempo de anuncio periodico
announce-frequency = 15
; Anuncia la posicion solamente
announce-position = yes
; Tiempo minimo entre anuncios
min-announce-frequency = 10

member => SIP/32,0,Perico
member => SIP/33,0,Francis
member => SIP/34,0,Pep
```

La mayor parte de las opciones quedan comentadas, significativamente, hemos asociado tres agentes de la cola que hemos creado llamada “Almacén” con los respectivos nombres de usuario, y asociados a los tres dispositivos SIP de cada uno de ellos.

Por otro lado en el Plan de Marcación necesitamos hacer algunos ajustes a petición. Ajustamos un nuevo teléfono de tipo DDI ofrecido por la operadora:

/etc/asterisk/extensions.conf

```
;Numero General de UCA Autos
exten => 956001181,1,Goto(especiales,101,1)
```

Y por otro lado, creamos la extensión 101 en el contexto [especiales] para poder acceder a la cola **almacén** recién creada.

/etc/asterisk/extensions.conf

```
[especiales]

exten => 101,1,NoOp()
same => n,Playback(bienvenida)
same => n,Queue(almacen)
same => n,Hangup()
```

Finalmente queremos darle la opción a los usuarios del Call-Center para que se puedan poner en modo Pausa, por si tienen que atender otras necesidades. Para ello vamos a crear Macros que se ajusten a la función de poner y quitar la Pausa del Agente dentro de la cola dentro del plan de marcación de las respectivas extensiones:

/etc/asterisk/extensions.conf

```
[macro-pausa]
exten => s,1,NoOp()
same => n,PauseQueueMember(almacen,SIP/${MACRO_EXTEN:1})
same => n,Hangup()

[macro-sinpausa]
exten => s,1,NoOp()
same => n,UnpauseQueueMember(almacen,SIP/${MACRO_EXTEN:1})
same => n,Hangup()

[extensiones]

exten => _73[2-4],1,Macro(pausa)
exten => _83[2-4],1,Macro(sinpausa)
```

Con esto quedaría el sistema base de Colas completamente funcional. Ahora por otro lado, queremos ofrecerle a UCA Autos eso que la mayor parte, ofrece como una solución con licencias bastante caras, el sistema de Estadísticas de control y Analisis.

Para ello vamos a utilizar la herramienta Asternic Stats, en su formato Open Source. Siguiendo el esquema de instalación ofrecido en la Wiki ⁷ [14] tenemos montado todo en un tiempo relativamente corto, podemos ver en la Figura 5.9 una captura de la lista de los tres usuarios de Almacén y su respectiva disponibilidad.

En última instancia, quería preparar el sistema de Grabación de llamadas, y de escucha. Era importante hacer saber al Sr. Zutano que existían aspectos legales que tenía que tener controlados, con respecto a la grabación de llamadas (previo consentimiento del cliente, mediante una locución al inicio de la llamada, informando al respecto), para ello hacemos una pequeña modificación en la extensión 101, introduciendo esta locución específica por un lado.

Por otro lado, vamos a lanzar la aplicación de grabación Monitor, para que solo grabe cuando exista comunicación entre ambas partes de la comunicación. Aparte guardaremos los registros en ficheros indi-

⁷WIKIAsterisk, Asternic Stats, http://wikiasterisk.com/index.php?title=Asternic_Stats




Estado de Agentes		
Cola	Agente	
almacen	Pep	 No disponible
almacen	Francis	 Libre
almacen	Perico	 En Pausa

Figura 5.9: Captura de la Pantalla de Disponibilidad de Agentes de Asternic

viduales mezclados en formato wav, y tendran nombres distintos para poder identificarlos y que no se solapen unos con los otros:

/etc/asterisk/extensions.conf

```
exten => 101,1,NoOp()
same => n,Playback(bienvenida)
same => n,Playback(grabacion)
same => n,Set (GRABACION="grabacion-${STRFTIME (${EPOCH},, %Y %m %d- %H %M %S) })
same => n,Monitor(wav,\${GRABACION},mb)
same => n,Queue(almacen)
same => n,Hangup()
```

Vamos a crear un enlace simbolico a nuestro directorio web raíz, a la carpeta de grabaciones de Asterisk para que puedan acceder a las mismas de forma directa el responsable. Asi podría acceder directamente desde el servidor web <http://pfc-asterisk.local/monitor> y ver todas las grabaciones:

```
cd /var/www/
ln -s /var/spool/asterisk/monitor ./
```

Por otro lado, tenemos que informar por contrato a los empleados, que sus llamadas pueden ser escuchadas sin previa autorización, para ello remitimos al gerente a su responsable legal para que redacte el mismo. Con esto, habiendo allanado el terreno de lo legal, podemos proceder a realizar las modificaciones pertinentes

En cuanto a la posibilidad de ofrecer escucha, simplemente tenemos que añadir una nueva extensión para realizarla, en este caso lo haremos dentro del contexto [manager]

/etc/asterisk/extensions.conf

```
exten => 201,1,NoOp()
same => n,ChanSpy(SIP/3)
same => n,Hangup()
```


5.4.2. Resolución: Colas en Asterisk

En primer lugar, para comprobar el funcionamiento de las colas, es muy directo. Llamamos desde cualquier teléfono a la extensión específica de las colas 101, y si tenemos teléfonos del rango 3X (extensiones de almacén), deberíamos de poder escuchar la llamada entrante. Si tenemos varios teléfonos para probar, podríamos comprobar como el primero pasa a estar en primera posición de la cola, y además todos los teléfonos asociados a la cola suenan a la vez por la estrategia **ringall**. Además los llamantes deberían recibir mensajes de audio indicándoles la posición en la cola, si todo esta correctamente parametrizado tal y como podemos ver en el desarrollo.

Podemos acceder simultáneamente, cuando hay varias llamadas de prueba cursándose en la cola al sistema Asternic Stats, y en el apartado Realtime tenemos que ver como los eventos van surgiendo de forma inmediata y automática. Además si todo el registro de la cola va realizándose correctamente, empezaremos a ver estadísticas en los otros apartados correspondientes, un ejemplo de estadísticas podríamos ver en la Figura 5.10.

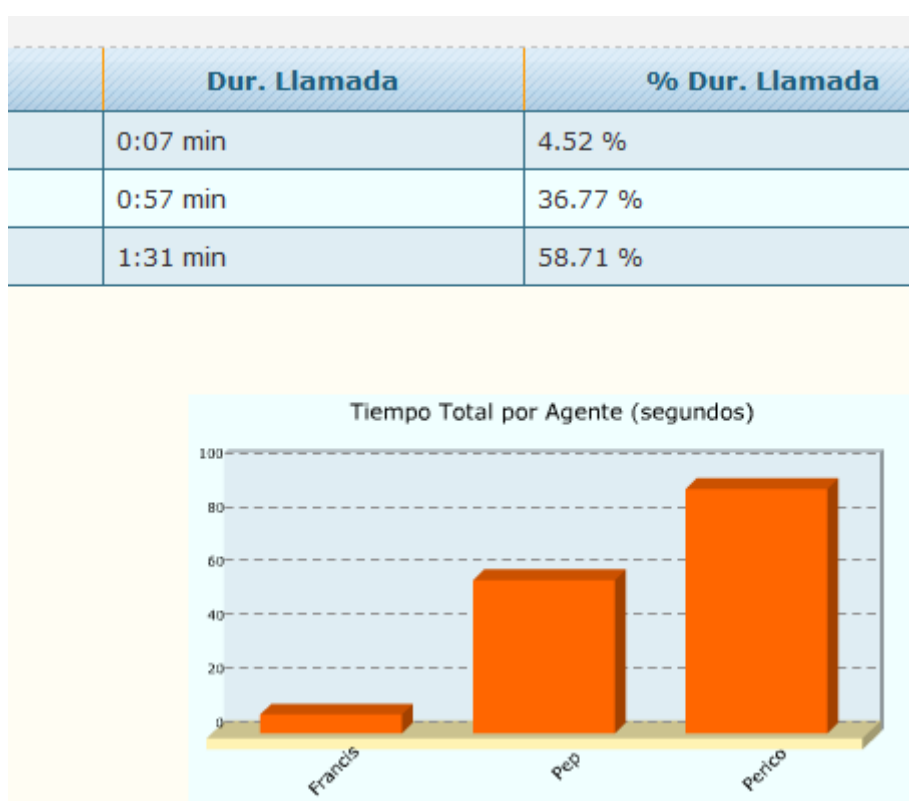


Figura 5.10: Ejemplo de Estadísticas Asternic Stats

Para la escucha activa, cuando un agente o varios entre las extensiones 3X este hablando, podemos ejecutar desde un dispositivo la extensión 201, y automáticamente deberíamos poder estar escuchando la conversación en ese canal.

Finalmente para comprobar que las grabaciones se estan haciendo correctamente, simplemente tenemos que acceder a la URL <http://pfc-asterisk.local/monitor> para comprobar que van apareciendo ficheros de audio en su interior.

5.5. Nuevas Incorporaciones

Poco tiempo después de incorporar el Centro de Llamadas para Almacén dentro de la empresa, el gerente Sr. Doe, vuelve a contactar con nosotros. Nos indica que el sistema ha prosperado y que el Mánager de Almacén ha conseguido controlar la situación sin mucha dificultad, próximamente contratarán a una persona más, y así modifiquemos un poco el sistema, porque se ven un poco desbordados dado que se les acumula el trabajo que ya previamente hacían con la nueva encomendación, aunque sus ventas estaban aumentando de forma bastante fuerte. Pero la llamada fue realmente para informarnos, que había contratado a una nueva persona, dentro del departamento de Marketing, ya que tenían intención de mejorar algunos aspectos a nivel tecnológicos dentro de la empresa, y montar un CRM para la gestión de la fuerza comercial dado que también estaba aumentando y tenía programado contratar dos vendedores más en los próximos días. Concertamos una cita con el nuevo empleado, llamado Pedro de los Palotes, unos días después de incorporarse.

La nueva figura, tenía conocimientos aceptables sobre el mundo del marketing, pero su principal manejo era las tecnologías de la información, y unos buenos conocimientos de informática en general. Durante la reunión que mantuvimos con el, nos comentó por encima alguna de las funciones que tenía que cumplir, y que tenían mucho que ver con nosotros, dado que necesitaba que introdujéramos algunas modificaciones en la central si fueran posibles. Además el Sr. Doe le dio carta blanca para ello, dado que tenía muy buenas referencias de él, y sabía bien lo que hacía. Lo que pudimos sacar fue:

1. Le habían ofrecido la oportunidad de montar en un servidor independiente para incorporar un sistema de CRM
2. Querían que controlara los gastos y la facturación de los conceptos de telefonía dado que habían observado ciertos usos indebidos, pero no sabían a quien achacar esta responsabilidad
3. Necesitaban que el Sr. de los Palotes, tuviera más autocontrol sobre Asterisk, aunque no tenía conocimientos de como funcionaba internamente, el sistema, para poder ser algo más autónomos dado los cambios que se avecinaban.
4. Finalmente, también el Mánager Comercial, el cual tenía dificultades a veces para establecer la comunicación con sus proveedores cuando necesitaban tener conversaciones entre más de dos usuarios simultáneamente, le había encomendado ver si era posible alguna pequeña mejora para poder solucionar esto.

Tras poner cada una de las tareas encima de la mesa, hicimos un primer planteamiento para dividir las nuevas encomendaciones en diferentes secuencias que iríamos incorporando en nuestra máquina Asterisk progresivamente para ir probando su funcionalidad con Pedro. Le pareció bien, y nos pusimos manos a la obra.

5.5.1. Generación Automática de Eventos

Antes de nada, creamos una nueva extensión para el Sr. Pedro de los Palotes, dentro del grupo de Marketing, la número 54, en el fichero sip.conf, y por otro lado, le habilitamos dicha extensión en el Plan de Marcación extensions.conf.

En primera instancia, hicimos un breve análisis de sus pretensiones de montar un servidor CRM propio, dado el mantenimiento y dedicación que esto llevaba, queríamos averiguar un poco su planteamiento y de que forma quería que nuestro sistema Asterisk trabajara con su sistema CRM. En primer lugar, no observamos que tuviera ningún conocimiento acerca de sistemas de motorización de sistemas informáticos, dado que en realidad aun sus buenos conocimientos del tema, realmente su función no era la mantener el sistema.

Hasta la fecha según nos comento, en su anterior empresa, había estado trabajando con un CRM llamado vTiger CRM [38], lo cual nos agradó bastante dado que conocíamos su buena integración con Asterisk a través del Asterisk Manager Interface. Pero en su caso, solo había utilizado vTiger en un servidor externalizado online, y todas las tareas de mantenimiento se las llevaba una empresa de hosting web, y el solo se encargaba de administrar y gestionar el sistema CRM. Según había oído, era posible con los CTI [45] de algunas PBX, era posible que directamente desde la interfaz web de vTiger, los usuarios pudieran cursar llamadas, y otras tantas funcionalidades.

Le informamos que esto era posible, siempre y cuando el servidor para su servicio, estuviera dentro de la misma red local, dado que “abrir” a Internet el CTI de Asterisk suponía serias implicaciones en la seguridad del sistema de las cuales no podíamos hacernos totalmente responsables. Le pusimos en contacto con una empresa colaboradora nuestra, Çadiz Computers"para que le facilitarán el trabajo de montaje, mantenimiento y administración del servidor CRM, pero de momento para que pudieran empezar a trabajar, le informamos de una funcionalidad de Asterisk que le serviría para poder establecer un mínimo de motorización sobre el estado de su servidor web, donde se contenía la aplicación vTiger, que realmente estaba basada en el lenguaje de programación PHP.

Para ello en primer lugar, considerando que el servidor web ya esta funcionando con su aplicación CRM en marcha, instalamos telnet en nuestro servidor Asterisk:

```
sudo aptitude install telnet
```

Y creamos un script bash que va a cumplir las función de motorización en un nivel muy básico, pero haciendo uso de nuestro sistema Asterisk como estaba previsto:

./apache_monitor.sh

```
#!/bin/bash

SERVER="home-asterisk.local"
PORT="80"
WORKDIR="/home/asterisk/repositorio/scripts"

echo "quit" | telnet $SERVER $PORT | grep "Escape character is"

if [ "$?" -ne 0 ]; then
    cp $WORKDIR/apache_warning.call $WORKDIR/apache_warning_temp.call
    mv $WORKDIR/apache_warning_temp.call /var/spool/asterisk/outgoing/
    exit 1
fi
```

Debemos darle privilegios de escritura:

```
chmod +x apache_monitor.sh
```

Y ahora tenemos que crear el fichero que generara un Llamada Automatica ⁸, que llamaremos **apache_warning.call**:

./apache_warning.call

```
Channel:SIP/54
Application:Playback
Data:alerta-web
CallerID:Emergencias CRM <111>
MaxRetries:3
WaitTime:90
RetryTime:10
```

Finalmente añadimos una linea en el fichero del proceso Cron, crontab, para que comprueba cada minuto, el estado de nuestro servidor gracias al script recién creado:

/etc/crontab

```
*/1 * * * * sh /home/asterisk/repositorio/scripts/apache_monitor.sh
```

En el momento que el servidor web deje de atender a las peticiones entrantes por el puerto 80, el Sr. de los Palotes recibirá una llamada de emergencia en su extensión personal, cuyo CallerID será Emergencias CRM <111>, hasta que se resuelva el problema.

5.5.2. Resolución: Generación Automática de Eventos

En el momento que paremos el servidor web en el servidor donde esta asociado el script deberían empezar a recibirse llamadas en la extensión 54 si la tenemos configurada en algun teléfono.

Para parar el servidor web, suponiendo por ejemplo que estemos utilizando apache en el servidor remoto, deberíamos ejecutar en el mismo:

```
# sudo service apache2 stop
```

En cuestión de un minuto máximo recibiremos la llamada. Podemos comprobar también en esa llamada que el Identificador de llamada hace referencia a “Emergencias Servidor” (Figura 5.11) Además deberíamos escuchar al responder una reproducción indicando la alerta.

⁸WikiAsterisk Generación Automática de Llamadas,
http://wikiasterisk.com/index.php?title=GeneraciC3B3n_Autom%C3A1tica_de_llamadas



Figura 5.11: Llamada Entrante de alerta

5.5.3. Probando nuestro CTI, Asterisk Manager Interface

Ahora vamos a montar el sistema para que el CRM vTiger pueda integrarse con nuestra máquina Asterisk. Tan pronto este preparado le daremos las credenciales al nuevo responsable IT para que pudiera poner en marcha lo necesario.

Cuando hicimos la primera instalación de Asternic Stats para la gestión de Colas ya tuvimos una pequeña aproximación al Manager de Asterisk, pero ahora se verá un poco más en detalle dados algunos problemas que surgieron durante la integración del sistema vTiger con nuestra máquina.

En primer lugar, necesitábamos definir el fichero `manager.conf`, encargado de toda la gestión de AMI y adaptarlo un poco más para diferenciar el proceso específico de Asternic, de la integración de vTiger, especialmente por si tenemos la necesidad de tener controlados los eventuales problemas posibles a nivel de depuración.

Para ello editamos el ficheros con la siguiente información:

`/etc/asterisk/manager.conf`

```
[general]
enabled = yes
port = 5038
webenabled = no
bindaddr = 0.0.0.0
allowmultiplelogin = yes
displayconnects = yes
timestampevents = yes
authlimit = 30

[qstats]
secret = qstats
read = all
permit = 127.0.0.1/255.255.255.0
permit = 192.168.101.151/255.255.255.0

[vtiger]
secret = vtiger
read = all
write = call
permit = 127.0.0.1/255.255.255.0
permit = 192.168.101.151/255.255.255.0

[meetme]
secret = meetme
read = call
```

```
write = command,originate
permit = 127.0.0.1/255.255.255.0
```

Viendo este fichero en detalle podemos observar por un lado, que hemos especificado un usuario concreto para Asterisk Stats, y le hemos retirado todos los privilegios de escritura, dado que al ser solo una aplicación de monitoreo no le va a hacer falta. Podríamos entender por el contexto de la aplicación que tipo de permisos de lectura le harían falta (seguramente los parámetros **agent** y **call** serían suficientes).

Y por otro lado, ya hemos definido un usuario específico para hacer la integración con vTiger, tampoco tenemos muy claro que permisos le harían falta, en este caso hemos definido la dirección IP local del servidor al que va a hacer referencia, para al menos establecer un punto más de seguridad y control al respecto.

El resto de los parámetros son los necesarios para el buen funcionamiento como podemos ver en detalle en el apartado de configuración del AMI ⁹ dentro de la WikiAsterisk.

A partir de aquí nos ponemos en contacto con Pedro para intentar coordinar el funcionamiento del sistema y ver si realiza la operación de manera adecuada. Ya hicimos otras integraciones de vTiger con nuestro sistema Asterisk en el pasado, pero dadas las constantes actualizaciones del CRM queríamos asegurarnos que todo iba bien.

En primer lugar, hicimos la parte de configuración más clásica del sistema para el usuario del Mánager Comercial que sería uno de los principales actores dentro de esta aplicación, el Sr. Douglas, según vemos en la siguiente figura 5.12

7. Asterisk Configuration			
Asterisk Extension	21	Receive Incoming Calls	yes

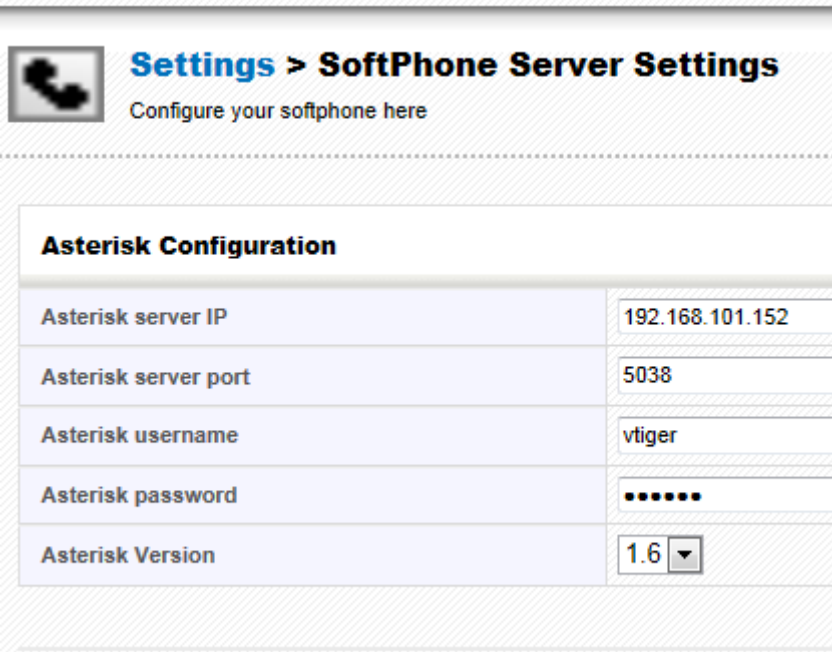
Figura 5.12: Configuración de la extensión en Vtiger

Por otro lado era necesario cubrir los aspectos concretos de configuración del sistema Vtiger para interconectar con nuestra interfaz AMI con las credenciales recién configuradas, reflejado en la Figura 5.13.

Con esto supuestamente le indicamos al Sr. de los Palotes que probara a realizar las operaciones que conocía para trabajar con el sistema vTiger, pero comprobamos que no funcionaba nada. Buscando en la web, pudimos observar que vTiger requería de algunas modificaciones a nivel interno para su correcta integración con nuestro Plan de Marcación, así que revisando la página oficial de configuración [37] nos dimos cuenta que era fundamental modificar un archivo propio del PBXManager (al parecer un módulo encargado de realizar toda esta gestión de interpretación de Mánager de Asterisk), y las modificaciones pertinentes las hicimos según podemos ver a continuación:

```
./www/vtigercrm/modules/PBXManager/utils/AsteriskClass.php
switch ($typeCalled) {
    case "SIP":
        $context = "extensiones";
        break;
    case "PSTN":
        $context = "extensiones"; // "outbound-dialing";
}
```

⁹Configuración AMI, <http://wikiasterisk.com/index.php?title=AMI>



Asterisk Configuration	
Asterisk server IP	192.168.101.152
Asterisk server port	5038
Asterisk username	vtiger
Asterisk password
Asterisk Version	1.6 ▼

Figura 5.13: Configuración de los parámetros AMI en Vtiger

```
break;
default:
    $context = "extensiones";
}
```

Con esto pudimos hacer que todas las llamadas salientes desde vTiger, se encaminaran a nuestro contexto de “Extensiones” principal, con suponía observar. Al parecer era posible realizar personalizaciones dentro de vTiger en función del tipo de teléfono al que deseáramos llamar (si fuera un teléfono tipo SIP, o por la línea de teléfono convencional (PSTN), pero como en nuestro caso, no hacemos esta diferenciación dentro de nuestro DialPlan, preferimos optar por utilizar el mismo contexto del DialPlan por defecto.

Y por último estuvimos comprobando, si llamando a la extensión 21 que acabábamos de configurar para el Sr. Douglas, se recibía dentro del sistema vTiger un mensaje pop-up indicando información referente al llamante, para comprobar que no era así. Al parecer las últimas versiones de vTiger provocaban serios fallos en la integración con instalaciones de Asterisk versiones 1.8 o superiores, de hecho como vimos en la Figura 5.13 en la versión solo podíamos indicar 1.6 como máximo.

Así que tuvimos que realizar ya algunas modificaciones más intensas de código, dentro del fichero que manejaba esta habilidad de gestionar las llamadas entrantes.

La incidencia más significativa que encontramos fue, dentro de la función **asterisk_handleResponse2** nos dimos cuenta, que no solo el condicional tampoco valía, la sintaxis de uno de los parámetros llamado AppData del comando NewExten, había cambiado significativamente, por esto adaptamos un poco el código PHP para que sirviera para esta nueva aplicación:

`./www/vtigercrm/cron/modules/PBXManager/AsteriskClient.php`

```
// if($mainresponse['Event'] == 'Newexten' &&
//(strstr($appdata, "__DIALED_NUMBER") ||
//strstr($appdata, "EXTTOCALL"))
if($mainresponse['Event'] == 'Newexten' &&
\ $mainresponse['Application'] == 'Dial')

// $splits = explode('=', $appdata);
// $extension = $splits[1];
$splits = explode('/', $appdata);
$longextension = $splits[1];
$splits = explode(',', $longextension);
$extension = $splits[0];
```

Esto ocurría porque concretamente, utilizando la función PHP “explode” para segmentar una cadena de texto, la cadena dentro Event Newexten, correspondiente al parámetro AppData, tenía una estructura del tipo, SIP/10,30, siendo SIP/10 el dispositivo SIP al que apuntaría la aplicación para mostrar el mensaje emergente, y 30, los segundos antes del "timeout". Con esto, retiraría todos estos argumentos y nos quedaríamos exclusivamente con el "10" para este ejemplo que podría corresponderse con la extensión asignada a un usuario en el sistema vTiger.

Y con estos dos cambios el sistema ya funcionaba a la perfección, y dimos por integrado vTiger con nuestra versión de Asterisk.

5.5.4. Comprobación: Probando nuestro CTI, Asterisk Manager Interface

Esta parte está principalmente orientada al uso de vTiger que se realizó en el servidor externo, habría que considerar que el correcto desarrollo de un sistema externo como vTiger realmente no está directamente relacionado con la idea de este proyecto, pero provee de un ejemplo interesante dentro del concepto de las Comunicaciones Unificadas.

Teniendo presentes algunos conceptos básicos de instalación del mismo que no se han cubierto dentro de este proyecto, y haciendo las pequeñas modificaciones necesarias podríamos conseguir los resultados.

Igualmente hemos provisto de una instalación de vTiger de ejemplo con las credenciales **hdouglas**, y la contraseña **asterisk** haciendo referencia al Jefe Comercial del Caso de Estudio.

A continuación en la Figura 5.14 y la Figura 5.15 podemos comprobar los mensajes emergentes, para visualizar el resultado de una correcta puesta en marcha.

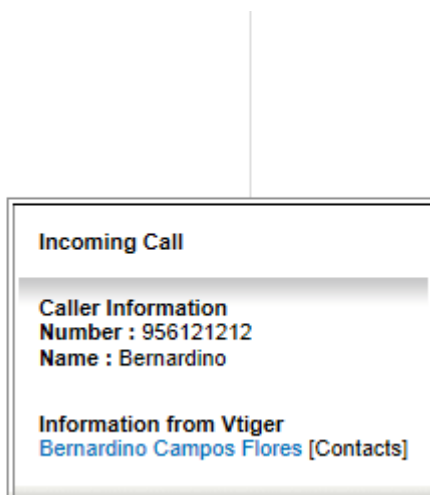


Figura 5.14: Llamada Entrante en Vtiger

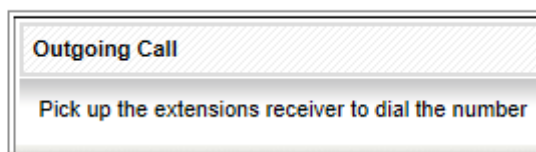


Figura 5.15: Llamada Saliente en Vtiger

5.5.5. Devolviendo el Control al Cliente

Atendiendo a la petición de nuestro cliente para poder ofrecerle la posibilidad de ser más autónomo en la gestión de sus propias extensiones, con la voluntad de darlas de alta y baja por cuenta propia, era fundamental incorporar en nuestro sistema Asterisk un nuevo elemento, el sistema Asterisk RealTime, equivalente a “trasvasar” todo el sistema de configuración interno de los ficheros de texto plano, estáticos por naturaleza, a un sistema más dinámico, basado en Bases de Datos, cuya interpretación de las mismas cara a la configuración del sistema, se haría en tiempo real. Esta funcionalidad que ofrece Asterisk, es muy práctica, si tenemos la intención de querer brindar algún tipo de interfaz, por ejemplo, en entorno web, para modificar los parámetros de una manera mucho más cómoda y confiable.

Aunque es cierto que existen interfaces web ya creadas a medida para Asterisk, que no requieren de Bases de datos, ya que también permiten interpretar los ficheros de texto plano directamente, y estos adaptarlos a la misma, para mostrar su contenido y poder a su vez ser editado y reintroducido en los ficheros de configuración, el nivel de control que brindan al usuario que las utiliza es prácticamente total, y esto significa que un usuario inexperto podría provocar un problema en el sistema de forma casi involuntaria.

Además, por la propia definición de estas interfaces, su sistema radica, primero, en una lectura línea

por línea del fichero de configuración, para interpretar su contenido, plasmarlo en la interfaz cara al usuario, y luego en caso de edición por parte de este, volver a reescribir literalmente ese fichero de texto de forma automáticamente, en el formato más conveniente por el sistema interfaz. Esto supone que la introducción y configuración de nuevos parámetros por un usuario más avanzado puede suponer una interferencia e incluso una inconveniencia, aunque dado el caso, existe la posibilidad de adaptar ficheros de configuración a medida en comunión a los “automáticos” generados por estas interfaces, que en un peor de los casos, incluso podrían provocar una redundancia en la funcionalidad (y poder derivar en errores de difícil detección)

En general este tipo de interfaces, solo son utilizadas por usuarios noveles del sistema Asterisk y el nivel de personificación de la máquina esta limitada realmente al número de posibilidades que ofrezcan estas interfaces por defecto independientemente de las posibilidades de “personalización” que puedan llegar a aportar. Algunos de los contras que he comentado, suelen ser las críticas que alegan, la mayor parte de los usuarios más experimentados en la comunidad Asterisk.

En consecuencia a esto, la idea para poder ofrecer autonomía sería la siguiente:

1. Vamos a ofrecer la posibilidad de configurar pares SIP a voluntad del usuario, como extensiones posibles dentro del rango 10 al 59
2. Tendremos que pasar la configuración del sistema SIP a Asterisk RealTime realizando las configuraciones en el sistema oportunas
3. Tendremos que modificar el plan de marcación para ofrecer el número máximo de opciones que contemplen el rango limitado de creación de extensiones que vamos a ofrecer
4. Finalmente crearemos una aplicación en PHP para la creación y eliminación de extensiones SIP de forma dinámica

Considerando esto, en primer lugar, tocaría realizar la configuración de los ficheros correspondientes, para poder establecer un puente entre Asterisk, y nuestra base de datos SQL donde alojaremos la configuración de los pares SIP (sippeers). Para ello utilizaremos el driver ODBC de nuestro sistema Linux que asociaremos con el recurso ODBC que ofrece Asterisk.

El primer fichero de configuración sería extconfig.conf como vemos a continuación:

/etc/asterisk/extconfig.conf

```
[settings]

sippeers => odbc,asterisk_odbc,pares_sip
```

Un fichero muy sencillo, indicamos que el módulo que vamos a convertir es el encargado de gestionar los pares del canal SIP, utilizando el driver ODBC, para enlazarlo con el recurso ODBC de Asterisk en el contexto **asterisk_odbc**, y que la tabla se llamará **pares_sip**.

Ahora el segundo fichero, sería res_odbc.conf:

/etc/asterisk/res_odbc.conf

```
[asterisk_odbc]
enabled => yes
dsn => asterisk
username => asterisk
password => asterisk
pre-connect => yes

[meetme_odbc]
```

```
enabled => yes
dsn => meetme
username => asterisk
password => asterisk
pre-connect => yes
```

En este caso también una estructura muy sencilla. Especificamos que el DSN del Driver ODBC será llamado asterisk, y queremos que conecte utilizando unas credenciales específicas que vamos a añadir en nuestra base de datos MySQL del paquete LAMP que instalamos al principio

Ahora vamos a modificar el fichero sip.conf donde se encuentran los pares en formato “estático”, para que se adapte a la nueva configuración en formato Asterisk Realtime “dinámico”. Nos quedaría tan simple de la siguiente forma:

/etc/asterisk/sip.conf

```
[general]
; Nuestro idioma
language = es
; Cambiamos el puerto para ofrecer seguridad por ocultacion
bindport = 35060
allowguest = no
register => ucaautos:12345678@home-asterisk.local:35060
srvlookup = yes
; Parametro especifico de Asterisk RealTime
rtcachefriends = yes
```

Con estos pasos tan sencillos, ya tenemos la parte Asterisk funcionando.

Ahora nos toca preparar el Driver ODBC a nivel Linux. Considerando que estamos trabajando con Ubuntu, es importante, dado que las librerías ODBC para MySQL se ubican en directorios diferentes que en otras distribuciones como RedHat o CentOS. Por ello necesitamos especificar la ruta exacta utilizando el fichero de configuración ubicado en el directorio /etc/, llamado **odbcinst.ini**:

/etc/odbcinst.ini

```
[MySQL]
Description = MySQL ODBC MyODBC Driver
Driver = /usr/lib/odbc/libmyodbc.so
Setup = /usr/lib/odbc/libmyodbcS.so
```

Es importante considerar que tenemos los paquetes unixodbc y libmyodbc, para el correcto funcionamiento:

```
sudo aptitude install unixodbc libmyodbc
```

Y finalmente configuramos el DSN en el fichero odbc.ini:

/etc/odbc.ini

```
[asterisk]
Description = ODBC para MySQL
Driver = MySQL
```

```

Server = localhost
Database = asterisk
Socket = /var/run/mysqld/mysqld.sock

[meetme]
Description = ODBC para MeetMe
Driver = MySQL
Server = localhost
Database = meetme
Socket = /var/run/mysqld/mysqld.sock

```

En este caso, dado que las credenciales las especificamos en el recurso ODBC, vamos a realizar la conexión utilizando un socket específico para mysql que en Ubuntu server se encuentra en el directorio especificado dentro de la configuración de este fichero.

Ahora, necesitamos prepara la Base de Datos SQL, utilizando mySQL, concretamente la interfaz grafica phpMySQL basada en el lenguaje de programación PHP. Pero eventualmente para las primeras tareas básicas de configuración vamos a utilizar la interfaz por linea de comandos que incorpora MySQL en su instalación.

En primer lugar vamos a crear la base de datos **asterisk**, especifica para el cometido que estamos buscando:

```
mysqladmin -u root -p create asterisk
```

```
GRANT ALL PRIVILEGES ON 'asterisk' . * TO 'asterisk'@'localhost';
```

Ahora vamos a crear un usuario llamado asterisk, para que pueda operar exclusivamente en esta base de datos:

usuarioasterisk.sql

```

CREATE USER 'asterisk'@'localhost' IDENTIFIED BY 'asterisk';
GRANT USAGE ON * . * TO 'asterisk'@'localhost' IDENTIFIED BY 'asterisk'
WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR
0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;
GRANT ALL PRIVILEGES ON 'asterisk' . * TO 'asterisk'@'localhost';

```

Y lo introducimos en nuestro sistema:

```
mysql -u root -p < usuarioasterisk.sql
```

Ahora finalmente vamos a crear la tabla pares_sip dentro de la base de datos llamada asterisk, para ello creamos un fichero y lo cargamos:

pares_sip.sql

```

CREATE TABLE 'pares_sip' (
'id' int(11) NOT NULL auto_increment,
'name' varchar(80) NOT NULL default '',
'host' varchar(31) NOT NULL default '',
'nat' varchar(5) NOT NULL default 'no',
'type' enum('user','peer','friend') NOT NULL default 'friend',

```

```

`callerid` varchar(80) default NULL,
`context` varchar(80) default NULL,
`defaulttip` varchar(15) default NULL,
`dtmfmode` varchar(7) default NULL,
`fromuser` varchar(80) default NULL,
`fromdomain` varchar(80) default NULL,
`insecure` varchar(20) default NULL,
`mailbox` varchar(50) default NULL,
`md5secret` varchar(80) default NULL,
`deny` varchar(95) default NULL,
`permit` varchar(95) default NULL,
`qualify` char(3) default NULL,
`secret` varchar(80) default NULL,
`disallow` varchar(100) default 'all',
`allow` varchar(100) default 'alaw;ulaw;gsm',
`port` smallint(5) unsigned NOT NULL default '0',
`regserver` varchar(100) default NULL,
`regseconds` int(11) NOT NULL default '0',
`lastms` int(11) NOT NULL default '0',
`username` varchar(80) NOT NULL default '',
`defaultuser` varchar(80) NOT NULL default '',
`subscribecontext` varchar(80) default NULL,
`useragent` varchar(20) default NULL,
`useradmin` int(1) NOT NULL DEFAULT '0',
PRIMARY KEY (`id`),
UNIQUE KEY `name` (`name`),
KEY `name_2` (`name`)
) ENGINE=MyISAM ROW_FORMAT=DYNAMIC;

```

```
mysql -u root -p asterisk < pares_sip.sql
```

Notese que existe un campo, que no es creado por defecto en este tipo de tabla de sippeers. Estamos hablando del campo **useradmin**, que nos va a servir a posteriori, cuando diseñemos una pequeña aplicación para poder ofrecer un entorno visual de edición de usuarios de esta tabla, para configurar el sistema de pares al completo. Así que le damos este privilegio (1) a la extensión 10 y 54 que serán en principio mientras no se diga lo contrario, los dos “administradores” del sistema.

Con toda la estructura creada, solo nos queda empezar a insertar los pares con los que veníamos trabajando anteriormente dentro de la tabla y comprobar que todo funciona correctamente. Para ello creamos otro fichero sql donde insertamos toda la información al respecto, en este caso vamos a utilizar 3 pares de ejemplo, el resto los haremos utilizando la herramienta phpMyAdmin:

ejemplo_pares_sip.sql

```

INSERT INTO `asterisk`.`pares_sip` (`id`, `name`, `host`, `nat`, `type`,
`callerid`, `context`, `defaulttip`, `dtmfmode`, `fromuser`, `fromdomain`,
`insecure`, `mailbox`, `md5secret`, `deny`, `permit`, `qualify`, `secret`,
`disallow`, `allow`, `port`, `regserver`, `regseconds`, `lastms`,
`username`, `defaultuser`, `subscribecontext`, `useragent`, `useradmin`)
VALUES (NULL, 'provedorip', 'home-asterisk.local', 'no', 'friend', NULL,
`entrantes`, NULL, NULL, 'ucaautos', NULL, 'invite', NULL, NULL, NULL,
NULL, 'yes', '12345678', 'all', 'g729;ilbc;gsm;ulaw;alaw', '35060', NULL,
'0', '0', '', 'ucaautos', NULL, NULL, '0');

```

```

INSERT INTO `asterisk`.`pares_sip` (`id`, `name`, `host`, `nat`, `type`,
`callerid`, `context`, `defaultip`, `dtmfmode`, `fromuser`, `fromdomain`,
`insecure`, `mailbox`, `md5secret`, `deny`, `permit`, `qualify`, `secret`,
`disallow`, `allow`, `port`, `regserver`, `regseconds`, `lastms`,
`username`, `defaultuser`, `subscribecontext`, `useragent`, `useradmin`)
VALUES (NULL, '21', 'dynamic', 'no', 'friend', NULL, 'manager', NULL,
`rfc2833`, NULL, NULL, NULL, '21@default',
`5dbe664c6d998967ad9bfe4ea37521a1`,
NULL, NULL, 'yes', '', 'all', 'g729;ilbc;gsm;ulaw;alaw', '0', NULL, '0',
'0', '', '', NULL, NULL, '0');

INSERT INTO `asterisk`.`pares_sip` (`id`, `name`, `host`, `nat`, `type`,
`callerid`, `context`, `defaultip`, `dtmfmode`, `fromuser`, `fromdomain`,
`insecure`, `mailbox`, `md5secret`, `deny`, `permit`, `qualify`, `secret`,
`disallow`, `allow`, `port`, `regserver`, `regseconds`, `lastms`,
`username`, `defaultuser`, `subscribecontext`, `useragent`, `useradmin`)
VALUES (NULL, '54', 'dynamic', 'no', 'friend', NULL, 'resto', NULL,
`rfc2833`, NULL, NULL, NULL, '54@default',
`037f3c092ae05f73054d2d2dd0e7d5dc`,
NULL, NULL, 'yes', '', 'all', 'g729;ilbc;gsm;ulaw;alaw', '0', NULL, '0',
'0', '', '', NULL, NULL, '1');

```

```
mysql -u root -p asterisk < ejemplo_pares_sip.sql
```

Ahora es el momento de diseñar una aplicación en PHP muy sencillo para poder llevar una pequeña gestión de los pares SIP de forma descentralizada por parte de nuestro usuario. La idea sería, basandonos en la tabla recién creada “pares_sip”, poder considerarla como un sistema de acceso de usuarios para ciertos usuarios, y además, será la tabla que realmente pretendemos que nuestros usuarios de UCA Autos puedan editar a voluntad. Este script de momento es una primera instancia para poder ofrecer lo que nuestros usuarios necesitan, pero quizá en un futuro sería mejorable, optimizando la seguridad de acceso al mismo, ya que con técnicas poco ortodoxas como la Inyección de código SQL podrían acceder al mismo sin grandes dificultades.

El script se irá describiendo a continuación ¹⁰, primero conectamos a la base de datos **asterisk**:

./www/usuarios.php

```

<?php

// Conexion a la base de Datos

$base = "asterisk";
$host = "localhost";
$user = "asterisk";
$password = "asterisk";
$conexion = mysql\connect($host,$user,$password);
$result = mysql\select\_db($base,$conexion)
or die ("Error en la Conexion a BD");

```

¹⁰Montando un Sistema Asterisk Autonomo I,
<http://10000horas.com/asterisk/2011/12/08/montando-un-sistema-asterisk-autonomo-i>

A continuación, registramos una sesión con una Cookie, y hacemos una consulta condicional, a la base de datos en caso que no estemos logeados (es decir, en nuestra Cookie, no aparezca la variable de sesión **login** que indica que estamos registrados), y con esta consulta, comprobamos si el usuario esta registrado o no, en caso que lo este, registramos las variables de sesión incluyendo la variable **login**, para que no nos vuelva a solicitar inicio de sesión en adelante.

Como vemos, en caso que no hubiesemos accedido al sistema, nos aparecería un pequeño formulario pidiendo las credenciales, que al rellenarlo, lanzaría las credenciales como variables para una consulta SQL contra nuestra Base de Datos:

./www/usuarios.php

```
session_start();

// Si no estamos logeados

if (!$_SESSION['login'])
{
    if(isset($_POST['loginsubmit']))
    {
        $usuario = $_POST['usuario'];
        $password = $_POST['password'];

        if((!$usuario) || (!$password))
        {
            echo "Error 1<br>";
            exit();
        }
        $password = $usuario.":asterisk:". $password;
        $password = md5($password);
        $query = mysql_query("SELECT * FROM pares_sip WHERE
name='$usuario' AND md5secret = '$password' AND useradmin = '1'");
        if (mysql_num_rows($query) > 0)
        {
            session_register('login');
            $_SESSION['login'] = '1';
        }
        else
            echo "Error 2<br><br>";

        echo "<a href='usuarios.php'>Home</a>";
    }
    else
    {
        echo "<form method='post' action='?'>";
        echo "Usuario: <input name='usuario' type='text'><br>";
        echo "Contrase&ntilde;a: <input name='password' type='password'><br>";
        echo "<input type='submit' name='loginsubmit'>";
        echo "</form>";
    }
}
```

En caso que ya estemos logeados, nos ofrece un pequeño menu por si queremos avanzar hacia el formulario para insertar o destruir registros SIP o si queremos salir, en el segundo caso, destruiría la Cookie

con la sesión y nos daría la opción a volver a la pantalla de inicio de la aplicación donde nos vuelve a solicitar las credenciales de acceso.

A partir de aquí, considerando, que hayamos lanzado el formulario que veremos al final de la descripción de esta aplicación, en caso que hayamos rellenado el formulario de insertar un registro, recibiríamos los parámetros, y crearíamos una consulta SQL con los mismos a medida, para insertarla en nuestra base de datos. Con esto quedaría registrado nuestro usuario y podríamos volver a nuestro panel principal.

./www/usuarios.php

```
// Si ya estamos logeados

else
{
    // Salida del Sistema
    if(isset($_REQUEST['exit']))
    {
        session_destroy();

        if(!session_is_registered('login'))
            echo "<a href='usuarios.php'>Home</a>";

    }
    // Insercion de un nuevo Registro
    elseif(isset($_POST['insertsubmit']))
    {
        $sipuser = $_POST['sipuser'];
        $sippass = $_POST['sippass'];
        $contexto = $_POST['contexto'];
        $sippass = $sipuser.":asterisk:". $sippass;
        $sippass = md5($sippass);
        $buzon = $sipuser."@default";
        $query = mysql_query("INSERT INTO pares_sip
        ('name', 'host', 'nat', 'type', 'context', 'md5secret',
        'qualify', 'disallow', 'allow', 'port', 'regseconds',
        'lastms', 'username', 'defaultuser', 'mailbox')
        VALUES ('$sipuser', 'dynamic', 'no', 'friend', '$contexto',
        '$sippass', 'yes', 'all', 'alaw;gsm;ulaw',
        '0','0', '0', '', '', '$buzon')");

        echo "<a href='usuarios.php'>Home</a><br>";
        echo "<a href='usuarios.php?exit'>Exit</a>";
    }
}
```

La segunda posibilidad, es que hayamos optado por pulsar el botón del formulario principal para borrar un registro SIP. En este caso, haría el borrado lanzando la consulta SQL correspondiente, asociándolo al identificador concreto del Par SIP que hayamos decidido eliminar:

./www/usuarios.php

```
// Borrado de un Registro
elseif(isset($_POST['deletesubmit']))
{

```



```

$sipid = $_POST['sipid'];
$query = mysql_query("DELETE FROM pares_sip WHERE id = '$sipid'");

echo "<a href='usuarios.php'>Home</a><br>";
echo "<a href='usuarios.php?exit'>Exit</a>";
}

```

Finalmente, están los dos posibles formularios, tanto el de inserción como el de borrado, los cuales recogen las variables para pasárselas a las consultas anteriormente explicadas:

./www/usuarios.php

```

// Formularios de Insercion y Borrado
else
{
echo "Insertar Registro:<br>";
echo "<p><form method='post' action='?'>";

echo "Usuario: <input name='sipuser' type='text'><br>";
echo "Contrase&ntilde;a: <input name='sippass' type='password'><br>";
echo "Tipo: <select name='contexto'>";
echo "<option value='extensiones'>Resto</option>";
echo "<option value='manager'>Manager</option>";
echo "<option value='gerencia'>Gerencia</option>";
echo "</select><br>";
echo "<input type='submit' name='insertsubmit' value='Insertar'>";
echo "</form></p>";

echo "<table border='1'>";
echo "<tr><td colspan = '2' align='center'>SIP Peers Activos</td></tr>";
echo "<tr><td>Usuario</td><td>Borrar</td></tr>";

$query = mysql_query("SELECT * FROM pares_sip WHERE type = 'friend'");
$rows = mysql_num_rows($query);
for ($i=0;$i<$rows;$i++)
{
$sippeersarray = mysql_fetch_array($query);
$sipuser = $sippeersarray['name'];
$sipid = $sippeersarray['id'];

echo "<tr>";
echo "<td>".$sipuser."</td>";
echo "<td>";
echo "<form method='post' action='?'>";
echo "<input type=hidden name='sipid' value='\$sipid'>";
echo "<input type=submit name='deletesubmit' value='Borrar'>";
echo "</form>";
echo "</td>";
echo "</tr>";

}
echo "</table>";
echo "<a href='usuarios.php?exit'>Exit</a>";

}

```

```
}  
?>
```

Con esta aplicación somos capaces de crear todas las extensiones que creamos antes, exactamente con la misma configuración, en una cuestión de 1 minuto, siendo un ejemplo de escalabilidad y flexibilidad.

Finalmente hemos de modificar el Plan de Marcación para que acepte todo tipo de extensiones dentro del rango que establecimos (de la extensión 10 a la extensión 59 hasta nuevo aviso). Para ello realizamos la siguiente pequeña modificación:

/etc/asterisk/extensions.conf

```
; Todas las Extensiones para Asterisk RealTime  
exten => _[1-5]X,1,Macro(telefonos)  
exten => _9[1-5]X,1,Macro(buzon)
```

Y con esto ya quedaría terminada por completo esta parte del trabajo.

5.5.6. Resolución: Devolviendo el Control al Cliente

En primera instancia debemos acceder al servidor web, a la url <http://pfc-asterisk.local/usuarios.php> para comprobar que podemos acceder a la interfaz de creación de usuarios, acceder con nuestras credenciales (en el ejemplo las de la extensión 54, contraseña 1234), y probar a crear y borrar usuarios a voluntad.

Una vez hayamos creado un usuario SIP considerando el rango permitido que hablábamos antes (del 10 al 59), podemos crear una extensión que no estuviera antes, configurarla en un dispositivo SIP, y probar a llamar al mismo para comprobar que recibe la llamada correctamente.

El mismo efecto, para comprobar que el sistema de buzón de voz funciona, teniendo presente, que esto tendríamos que configurarlo de momento, manualmente dado que el sistema de Buzones de Voz, no se ha migrado al sistema RealTime, práctica que quizá también resultaría interesante para obtener una independencia integral.

5.5.7. Análisis de Asterisk con Lupa

Continuando con la serie de trabajos encomendados por el señor Pedro, era el momento de hacer frente a la gestión y el control del registro de llamadas, dado que según nos comentó nuestro cliente, había observado en la factura de teléfono que se estaban haciendo usos desmedidos del mismo, pero no podía controlar bien, quien y porque, estaba incurriendo en este gasto desmesurado. Quizá fuese simplemente porque se sabía que hasta el momento no existía un mecanismo de control eficiente, y era raro tener este tipo de controles para las PYME del sector, en las PBX convencionales dado que suponían unos sistemas de licenciamiento bastante costosos.

En este caso, sabemos que Asterisk ofrece un registro de llamadas que podemos extraer desde diversas fuentes, pasando por ficheros de texto hasta las bases de datos de diferentes tipos, siguiendo un estilo muy parecido de almacenamiento, al que describimos anteriormente cuando hicimos el trasvase de configuración del módulo SIP al modelo de Tiempo Real (Asterisk Realtime).

En primer lugar vamos a configurar el sistema de almacenamiento del Registro de Llamadas (CDR), en la base de Datos que creamos anteriormente para nuestro sistema RealTime, solo que vamos a tener que crear una tabla nueva específica para almacenar todos los registros CDR.

Por ello, la primera parte sería la de configurar el fichero `cdr_adaptive_odbc.conf` y dotarlo de una estructura genérica para que conecte a nuestro anterior “conector” para el registro DSN del Driver ODBC que especificamos en la anterior sección (el llamado `.asterisk_odbc` de `res_odbc.conf`). Con esto configuramos el fichero `cdr_adaptive_odbc.conf` de la siguiente forma:

`/etc/asterisk(cdr_adaptive_odbc.conf`

```
[odbc_cdr]
connection=asterisk_odbc
table=cdr
```

En segundo lugar, vamos a crear la tabla dentro de nuestra anteriormente creada base de datos Asterisk, con la siguiente estructura:

`sistemacdr.sql`

```
CREATE TABLE cdr (
  calldate datetime NOT NULL default '0000-00-00 00:00:00',
  clid varchar(80) NOT NULL default '',
  src varchar(80) NOT NULL default '',
  dst varchar(80) NOT NULL default '',
  dcontext varchar(80) NOT NULL default '',
  channel varchar(80) NOT NULL default '',
  dstchannel varchar(80) NOT NULL default '',
  lastapp varchar(80) NOT NULL default '',
  lastdata varchar(80) NOT NULL default '',
  duration int(11) NOT NULL default '0',
  billsec int(11) NOT NULL default '0',
  disposition varchar(45) NOT NULL default '',
  amaflags int(11) NOT NULL default '0',
  accountcode varchar(20) NOT NULL default '',
  uniqueid varchar(32) NOT NULL default '',
  userfield varchar(255) NOT NULL default ''
);

ALTER TABLE `cdr` ADD INDEX ( `calldate` );
```

```
ALTER TABLE `cdr` ADD INDEX ( `dst` );
ALTER TABLE `cdr` ADD INDEX ( `accountcode` );
```

Y cargamos la tabla dentro de nuestra base de datos “asterisk”:

```
mysql -u root -p asterisk < sistemacdr.sql
```

Con esto ya tendríamos el sistema CDR mandando registros automáticamente.

Por otro lado, necesitaríamos proveer de algún tipo de interfaz, para que nuestro cliente, pudiera observar fácilmente lo que tiene intención (en este caso, controlar el mal uso de las líneas telefónicas y detectar los abusos por parte del personal). Para ello realizaríamos la instalación de CDR-Stats como entorno web ¹¹ [1], según viene perfectamente descrito en la WikiAsterisk.

La instalación de este sistema, habría que considerar que ha ido bastante rápido, debido a que la preparación del sistema Asterisk Realtime, dio lugar a que las bases ya estuvieran sentadas, para la implementación de todos tipo de servicios que involucrarán la interrelación entre nuestra máquina Asterisk, y un sistema de Bases de Datos.

5.5.8. Resolución: Análisis de Asterisk con Lupa

En primer lugar, tenemos que intentar acceder al sistema CDR-Stats desde la URL <http://pfc-asterisk.local/cdr-stats> y comprobar que no hay ningún problema al menos en el acceso aunque pueda estar todo sin datos.

Para comprobar que esto funciona correctamente, en primer lugar tenemos que realizar múltiples llamadas aleatorias. Luego accediendo al sistema CDR-Stats nuevo deberían aparecer las llamadas prácticamente de inmediato

En la Figura 5.16 y la Figura 5.17 mostramos las posibilidades que ofrecerá para el Sr. De los Palotes la Interfaz que acabamos de preparar para el manejo de todo el Detalle de Llamadas según hubiera pretendido. Dado que la instalación todavía es muy prematura sin uso ninguno, solo se ve como quedarían los ejemplos, pero sin suficiente información para que resulten lo suficientemente esclarecedores:

Es importante considerar que si las llamadas no aparecen, podríamos comprobar si el sistema CDR esta en funcionamiento. Para ello podemos ejecutar la orden en la CLI:

```
CLI> cdr show status

Call Detail Record (CDR) settings
-----
Logging:                               Enabled
Mode:                                   Simple
Log unanswered calls:                  No

* Registered Backends
-----
Adaptive ODBC
cdr-custom
```

¹¹CDR-Stats en WikiAsterisk, <http://wikiasterisk.com/index.php?title=CDR-Stats>

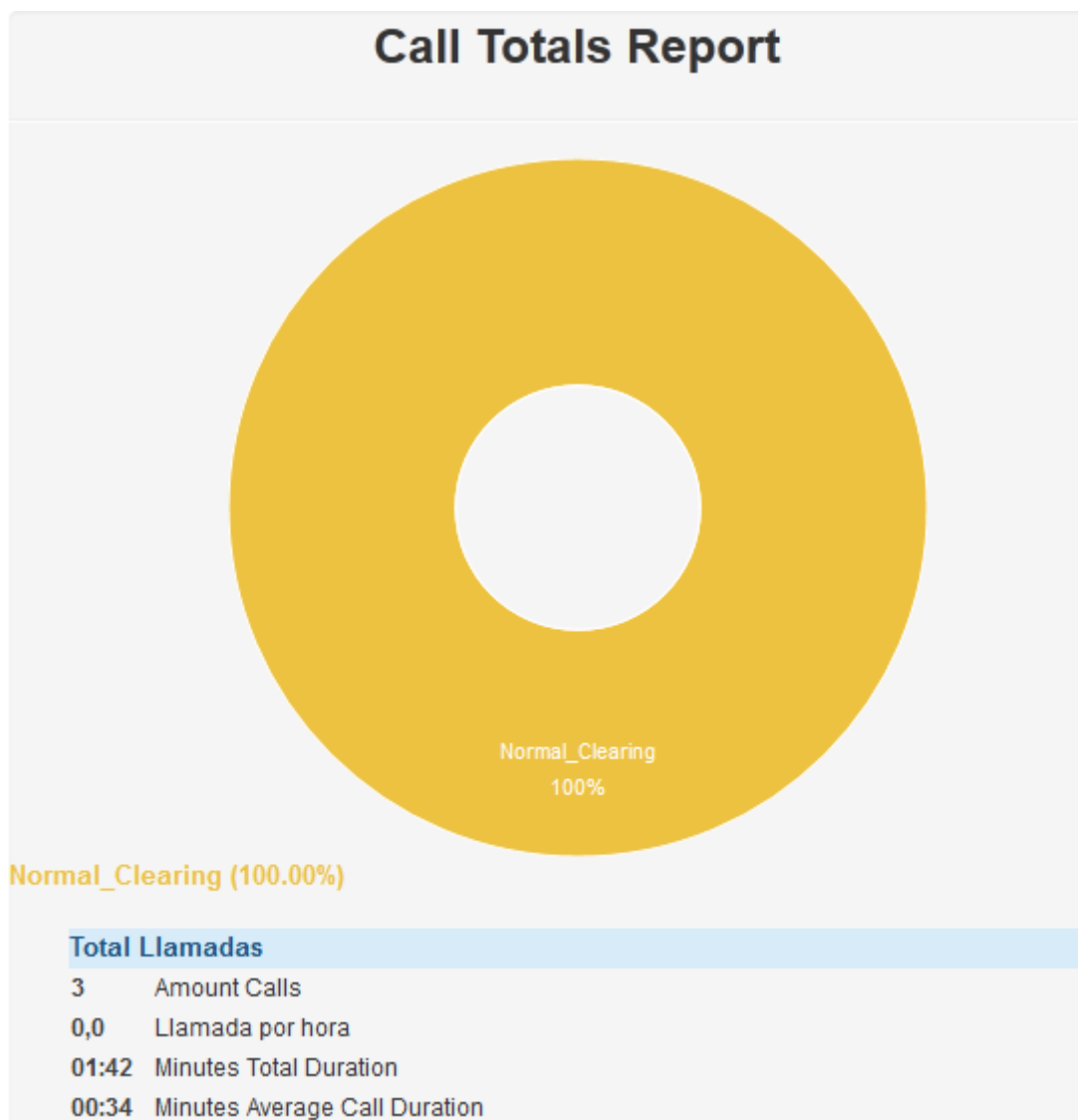


Figura 5.16: Gráfico Circular de Llamadas Totales

5.5.9. Conferencias Corporativas

Como último trabajo, nos tocaría introducir el sistema de Conferencing que incorpora Asterisk entre sus módulos generales, llamado MeetMe. También existe la opción de utilizar ConfBridge que es la nueva evolución del sistema, pero para este caso resulta suficiente el primero, y además integraremos una nueva interfaz web muy práctica, que seguirá en la línea de las anteriores configuraciones, aprovechando además la posibilidad de ofrecer configuraciones del sistema MeetMe en tiempo real (con Asterisk RealTime).

Nuestra intención dado el potencial que nos ofrece el sistema, es crear por un lado, una sala de conferencias estática, para uso del Jefe de Ventas, y luego un sistema general de salas de conferencia dinámica que ofreceremos al Sr. Pedro, para que pudiera crear futuras salas de conferencia a voluntad y ofrecerlas al resto de los empleados.

Para ello, vamos a asociar cada uno de los sistemas, a un número Direct Dial-In (DDI) cada una, y así

Call Statistics - 6th jun. 2012 with previous 2 días

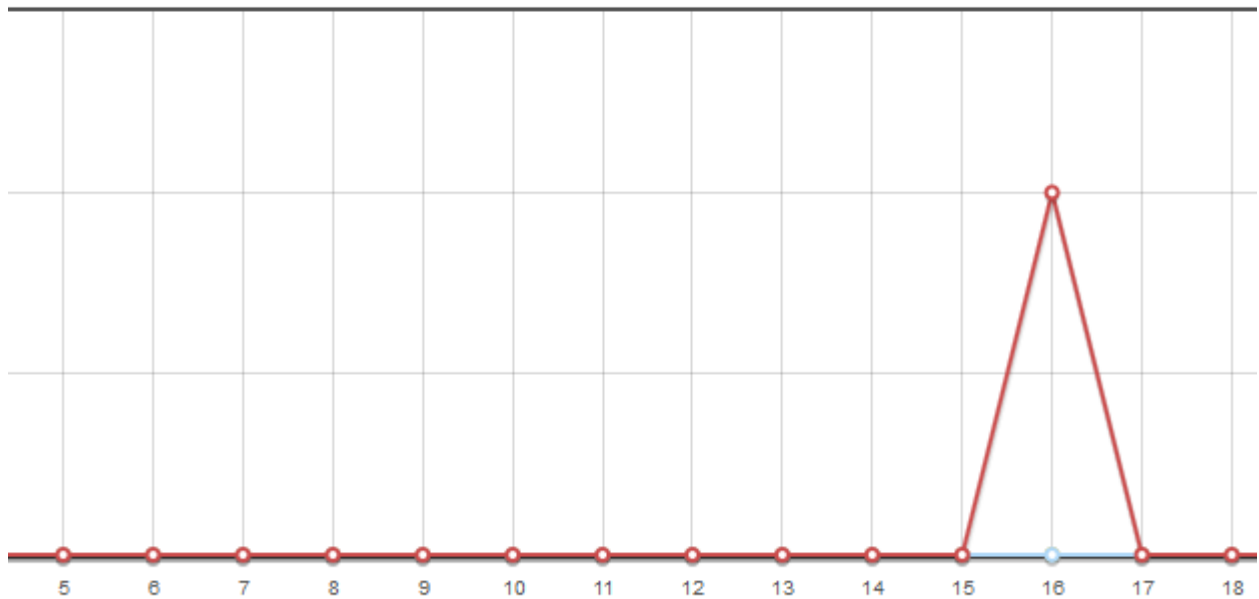


Figura 5.17: Histograma de Llamadas por días

podrán accederse a las mismas en cualquier momento. Solicitamos los respectivos números a nuestro operador UCA Telecom, y los asociamos al Plan de Marcación:

/etc/asterisk/extensions.conf

```
exten => 956001171,1,Goto(especiales,102,1)
exten => 956001172,1,Goto(especiales,103,1)
```

En este caso, el 956001171, sería para el sistema general de salas de conferencia, y el 956001172, para la sala de conferencias específica para el Jefe de Ventas.

Ahora, en primer lugar vamos a crear la sala de Conferencias sencilla, basándonos en la Aplicación Meetme, y la asociaremos a un número Direct Dial-In (DDI) para su correspondiente acceso desde el exterior, para los posibles proveedores, o clientes, según nos comentaba nuestro cliente.

Para ello, vamos a configurar por un lado el fichero meetme.conf, con los parámetros suficientes para conseguir este resultado, según podemos ver a continuación:

/etc/asterisk/meetme.conf

```
[general]
audiobuffers=32
logmembercount=yes
fuzzystart=300
earlyalert=3600
endalert=120
```

```

schedule=yes

[rooms]

conf => 100,1234,4321

```

En este caso, hemos definido, una única sala, la número 100, con dos códigos de acceso, uno para administrador, y otro para el resto de los usuarios, además de algunos parámetros específicos del sistema MeetMe.

Para poder acceder a la misma, simplemente definimos en nuestro plan de marcación la parte correspondiente:

/etc/asterisk/extensions.conf

```

exten => 103,1,NoOp()
same => n,MeetMe(100)
same => n,Hangup()

```

Por otro lado vamos a configurar el sistema MeetMe en formato Realtime, para poder utilizarlo con una interfaz web específica, llamada Web-Meetme ¹².

Así que debemos crear una base de datos específica para este contexto, que llamaremos meetme:

```
mysqladmin -u root -p create meetme
```

Y crearemos la estructura de tablas utilizando el siguiente script, extraído y adaptado, del fichero del paquete de instalación de Web-Meetme:

./tablasmeetme.sql

```

connect meetme;

CREATE TABLE `booking` (
  `bookId` int(10) unsigned NOT NULL auto_increment,
  `clientId` int(10) unsigned default '0',
  `confno` varchar(30) default '0',
  `pin` varchar(30) NOT NULL default '0',
  `adminpin` varchar(30) NOT NULL default '0',
  `starttime` datetime NOT NULL default '0000-00-00 00:00:00',
  `endtime` datetime NOT NULL default '0000-00-00 00:00:00',
  `dateReq` datetime NOT NULL default '0000-00-00 00:00:00',
  `dateMod` datetime NOT NULL default '0000-00-00 00:00:00',
  `maxUser` varchar(30) NOT NULL default '10',
  `status` varchar(30) NOT NULL default 'A',
  `confOwner` varchar(30) NOT NULL default '',
  `confDesc` varchar(100) NOT NULL default '',
  `adminopts` varchar(10) NOT NULL default '',
  `opts` varchar(10) NOT NULL default '',
  `sequenceNo` int(10) unsigned default '0',
  `recurInterval` int(10) unsigned default '0',
  `recordingfilename` varchar(128) default NULL,

```

¹²Instalación Web-Meetme en WIKIAsterisk, <http://wikiasterisk.com/index.php?title=Web-MeetMe>

```

        PRIMARY KEY  (`bookId`)
    ) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=145 ;

CREATE TABLE `cdr` (
  `bookId` int(11) default NULL,
  `duration` varchar(12) default NULL,
  `CIDnum` varchar(32) default NULL,
  `CIDname` varchar(32) default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE `notifications` (
  `id` int(11) NOT NULL auto_increment,
  `user_id` int(11) NOT NULL default '0',
  `book_id` int(11) NOT NULL default '0',
  `ntype` char(10) default NULL,
  `ndate` timestamp NOT NULL default CURRENT_TIMESTAMP,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

CREATE TABLE `participants` (
  `id` int(11) NOT NULL auto_increment,
  `user_id` int(11) NOT NULL default '0',
  `book_id` int(10) NOT NULL default '0',
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=782 ;

CREATE TABLE `user` (
  `id` int(11) NOT NULL auto_increment,
  `email` varchar(100) NOT NULL default '',
  `password` varchar(25) default NULL,
  `first_name` varchar(50) default NULL,
  `last_name` varchar(50) default NULL,
  `telephone` varchar(15) default NULL,
  `admin` varchar(5) NOT NULL default 'User',
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=20 ;

```

Y procedemos a insertarla dentro de nuestra base de datos “asterisk”:

```
mysql -u root -p meetme < tablasmeetme.sql
```

Añadimos un usuario administrador a la tabla “user” para nuestro cliente Don Pedro:

```
mysql -u root -p meetme -e "INSERT INTO meetme.user (first_name, last_name,
email, telephone, password, admin) VALUES ('Pedro', 'de los Palotes',
'ppalotes@wikiasterisk.com', '956001154', 'asterisk', 'Admin');"
```

Y también permitir en la base de datos **meetme** acceso total al usuario asterisk:

```
mysql -u root -p meetme -e "mysql -u root -p -e
\"GRANT ALL PRIVILEGES ON meetme.* TO 'asterisk'@'localhost';"
```

Por otro lado añadimos la siguiente línea en el fichero **extconfig.php** donde se definen los módulos convertibles a Asterisk Realtime y las tablas que lo asocian en las bases de datos:

/etc/asterisk/extconfig.conf

```
meetme => odbc,meetme_odbc,booking
```

También tenemos que incorporar algunas modificaciones en nuestro fichero `res_odbc.conf` para incorporar una nueva configuración, para que conecte al Driver ODBC con el DSN `meetme`, para la nueva base de datos con el mismo nombre:

/etc/asterisk/res_odbc.conf

```
[meetme_odbc]
enabled => yes
dsn => meetme
username => asterisk
password => asterisk
pre-connect => yes
```

Ya por último, terminar de configurar el driver ODBC, y especificar el DSN específico `meetme` con el que se realizará toda la transmisión de información entre MySQL y Asterisk:

/etc/odbc.ini

```
[meetme]
Description = MeetMe para MySQL
Driver = MySQL
Server = localhost
Database = meetme
Socket = /var/run/mysqld/mysqld.sock
```

Y ya culminando esta parte, solo queda la instalación propiamente dicha, del sistema Web-Meetme siguiendo los pasos de WikiAsterisk.com, considerar que modificaríamos el fichero `manager.conf` de la interfaz AMI para adaptarlo a la configuración específica de esta interfaz añadiendo lo siguiente:

/etc/asterisk/manager.conf

```
[meetme]
secret = meetme
read = call
write = command,originate
permit = 127.0.0.1/255.255.255.0
```

Una vez permitido el acceso al Manager con un usuario y contraseña específico es necesario que recurramos a un fichero concreto, que ofrece la instalación de Web-Meetme.

Considerando que Web-Meetme, incorpora una versión de PHP-AGI ¹³, solo necesitamos configurar el fichero `phpagi.conf` para dar la funcionalidad a la clase `AGI_AsteriskManager`:

/etc/asterisk/phpagi.conf

¹³Interfaz AGI, WIKIAsterisk, <http://wikiasterisk.com/index.php?title=AGI>

```
[phpagi]
debug=true
error_handler=true
admin=sir.louen@gmail.com
hostname=pfc-asterisk.local
tempdir=/var/spool/asterisk/tmp/

[festival]
text2wave=/usr/bin/text2wave

[asmanager]
server=localhost
port=5038
username=meetme
secret=meetme

[cepstral]
swift=/opt/swift/bin/swift
voice=David
```

Y tambien editar un fichero de configuración para la conexión a la base de datos de la aplicación MeetMe propia:

/var/www/web-meetme/lib/database.php

```
include_once 'DB.php';
$database = 'meetme';
$host = 'localhost';
$username = 'asterisk';
$password = 'asterisk';
```

Finalmente vamos a especificar algunos parámetros concretos, para que nuestro sistema Web-MeetMe funcione con plenitud ¹⁴.

/var/www/web-meetme/lib/defines.php

```
define ("WEBROOT", "http://pfc-asterisk.local/web-meetme/");
define ("FSROOT", "/var/www/web-meetme/");
define ("LIBDIR", FSROOT."lib/");

define ("OUT_CONTEXT", "extensiones");
define ("OUT_CALL_CID", "Invitacion Meetme <102>");

define ("AUTH_TYPE", "sqldb");
define ("AUTH_TIMEOUT", "3");

define ("LOCAL_SUPPORT", "Departamento de Soporte");
define ("LOCAL_PHONE", "956001151");
define ("PHONENUM", "956001171");
```

¹⁴Configuración Adicional Web-MeetMe, WikiAsterisk.com,
http://wikiasterisk.com/index.php?title=Web-MeetMe#Configuraci.C3.B3n_Adicional

Con todo esto, ya quedaría el sistema Web-Meetme completamente configurado, podemos ver una imagen con distintas conferencias creadas dinámicamente en la interfaz, en la Figura 5.18. Ya solo quedaría realizar un pequeño ajuste en el Plan de Marcación, para incluir un nuevo número DDI específico para conectar a nuestro sistema de salas dinámicas:


- Scheduled Conference :		
Conference #	Conference Name	Starts 
123	123	06/07/12 12:11pm
75470	fdasfasdf	06/07/12 2:06pm
38108	Pruebas	06/07/12 2:14pm

Figura 5.18: Lista de Conferencias Dinámicas Web-MeetMe

/etc/asterisk/extensions.conf

```
exten => 102,1,NoOp()
same => n,MeetMe()
same => n,Hangup()
```

Con esto ya habríamos terminado todas las tareas que nos encomendó Pedro de los Palotes y podríamos proceder a la entrega según previsto.

5.5.10. Resolución: Conferencias Corporativas

Por un lado tenemos el sistema MeetMe estático. Su comprobación es directa. Desde dos terminales accedemos a la extensión 103, marcamos la clave 1234 y deberíamos de estar en comunicación entre estos dos teléfonos de manera inmediata.

Podemos agregar un tercero para comprobar que la conferencia es correcta aunque con dos realmente es suficiente.

En el caso del sistema Realtime, podríamos comprobar que funciona correctamente utilizando el comando CLI, vemos que aparece **Connected: Yes**, simboliza que la conexión esta establecida. Desde la interfaz web podríamos crear una sala de conferencias, y llamando a la extensión 102, acceder a la misma, siguiendo las instrucciones de voz que nos va indicando el sistema.

```
CLI> odbc show meetme_odbc

ODBC DSN Settings
```

Name: meetme_odbc

DSN: meetme

Last connection attempt: 1970-01-01 01:00:00

Pooled: No

Connected: Yes

5.6. Bienvenidos a la Telefonía 2.0

Varias semanas después de haber entregado los trabajos al Sr. de los Palotes, y habiendo comprobado que todo respondía según lo previsto, recibimos una llamada del Sr. Doe porque quería comentarnos un proyecto que tenía en mente y quería llevar adelante, así que concretamos un día de visita en la misma semana.

En la reunión tuvimos por primera vez contacto con el responsable de Posventa, el cual se encontraba junto a Zutano, hablando en el momento de nuestra llegada. Muy satisfechos con el nivel de funcionalidad que habían logrado alcanzar gracias a nuestro sistema Asterisk, querían proponernos la posibilidad de llevar a cabo un desarrollo interesante, que les resultaba muy útil y también como un precedente para poder ver si realmente nuestro sistema Asterisk podía llegar hasta donde le habíamos dicho

La idea surgía a raíz de una necesidad por parte del departamento de Posventa, dado que observaron que existía un amplio número de llamadas de clientes, pidiendo información acerca de la garantía de su vehículo y el tiempo que les quedaba para que expirase la misma, hacía perder tiempo al personal al teléfono, ya que tenían que consultar la base de datos de clientes, en el apartado de Garantías, y ver la fecha de compra, y los años contratados, para decirles cuanto tiempo les restaba. Pensando al respecto, se nos ocurrió la idea, de asociar en el teléfono directo, diese la opción al cliente, de marcando un dígito, de solicitarle el DNI, y que hiciese una consulta a la base de datos donde estaba conectado su ERP, buscando el cliente por DNI, y extrayendo la información relativa a la garantía, y con un sistema de síntesis de voz, decirle al cliente los días que le quedaban para que finalizase el contrato.

Tanto Zutano, como el responsable de Posventa, quedaron más que satisfechos con la idea propuesta, y nos pidieron que la lleváramos adelante con prontitud.

5.6.1. Respuesta de Voz Interactiva

Después de todo el despliegue, ya íbamos a desarrollar la primera aplicación que hace nuestro sistema Asterisk que destaque por encima de otras soluciones estándar. El concepto subyacente, era muy sencillo.

En primer lugar, íbamos a montar una pequeña Operadora Automática en una extensión general para el departamento de Posventa, íbamos a requerir un teléfono directo (DDI), para mismo por un lado, y por otro lado, desarrollar el sistema de respuesta de voz interactiva (IVR) [47], para que realizase la gestión que nos solicitaban.

La idea sería aprovechar la funcionalidad de Reconocimiento de Voz Automático (ASR) para capturar los dígitos del DNI, y en caso de intentarlo dos veces sin éxito, solicitar la marcación vía teclado numérico. Una vez capturado el numero de DNI, pasaríamos a hacer la consulta en la base de datos, y recogeríamos tanto la Fecha de Compra como los años de garantía, para sacar la fecha de finalización, y restando los días a la fecha actual, comprobar si ya esta finalizada, o los días que quedan, remitírselos los al cliente, con un mensaje síntesis de voz (TTS) expresamente, diciendo el número de días restantes.

Una vez establecido el Algoritmo que vamos a seguir, procedemos a su construcción en nuestro sistema.

Vamos a empezar con la construcción de la aplicación de recogida de información, en el lenguaje PHP, utilizando la interfaz que proporciona Asterisk AGI, y para establecer la comunicación bidireccional utilizaremos el API, PHP-AGI ¹⁵.

En primer lugar, necesitamos una función para convertir fechas de formato SQL a un formato que vamos a manejar a voluntad en este caso DDMMAAAA

/var/lib/asterisk/agi-bin/fingarantia.php

```
#!/usr/bin/php -q
<?php

require_once("/var/lib/asterisk/agi-bin/phpagi/phpagi.php");

function fecha_normal($fecha)
{
    ereg( " ([0-9]{2,4})-([0-9]{1,2})-([0-9]{1,2})", $fecha, $mifecha);
    $lafecha=$mifecha[3].$mifecha[2].$mifecha[1];
    return $lafecha;
}
```

En segundo lugar, otra función, calcular los días de diferencia entre dos fechas, en el formato que hemos establecido DDMMAAAA, la idea sería partiendo del formato de fecha, dividirla en un vector de tres elementos con la función `ereg`, convertirla a un formato de tipo marca de tiempo, en formato UNIX y los valores restarlos mutuamente y quitarle las horas, minutos y segundos, para conseguir como resultado la diferencia de días.

/var/lib/asterisk/agi-bin/fingarantia.php

```
function resta_fecha($fecha_inicial, $fecha_final)
{
    ereg( " ([0-9]{1,2}) ([0-9]{1,2}) ([0-9]{2,4})", $fecha_inicial, $fini);
```

¹⁵ AGI y PHP en WIKIAsterisk, <http://wikiasterisk.com/index.php?title=AGI>

```

ereg( "([0-9]{1,2})([0-9]{1,2})([0-9]{2,4})", $fecha_final, $ffin);

$fecha1 = mktime(0,0,0,$fini[2], $fini[1], $fini[3]);
$fecha2 = mktime(0,0,0,$ffin[2], $ffin[1], $ffin[3]);

return round(($fecha2 - $fecha1) / (60 * 60 * 24));
}

```

La siguiente función sirve para calcular la letra del DNI [2]:

/var/lib/asterisk/agi-bin/fingarantia.php

```

function letra_nif($dni)
{
    return substr("TRWAGMYFPDXBNJZSQVHLCKE", $dni%23,1);
}

```

A continuación, recogemos una información acerca del sistema AGI, y argumentos que recibimos, en este caso el DNI del cliente. Aparte también realizamos la conexión a la base de datos del ERP.

/var/lib/asterisk/agi-bin/fingarantia.php

```

$agi = new AGI();
$dni = $argv[1];
$dni = str_replace(" ", "", $dni);
$dni = $dni.letra_nif($dni);

// Conexion a la base de Datos
$base = "erp";
$host = "home-asterisk.local";
$user = "asterisk";
$password = "asterisk";
$conexion = mysql_connect($host, $user, $password);
$result = mysql_select_db($base, $conexion)
or die ("Error en la Conexion a BD");

```

A partir de aquí realizamos algunas consultas, para localizar los datos que necesitamos, es decir, fecha de compra y años de garantía, para así poder calcular los días de garantía que quedan. En caso de que encontremos el cliente por el DNI, devolveremos 0 días de garantía

/var/lib/asterisk/agi-bin/fingarantia.php

```

if(!mysql_num_rows($query))
{
    $diasgarantia = '0';
}
else
{
    $cliente = mysql_fetch_array($query);
    $idcliente = $cliente['id'];
}

```

```

$query2 = mysql_query("SELECT * FROM vehiculos
WHERE cliente = '$idcliente'");
$vehiculo = mysql_fetch_array($query2);
$fcompra = $vehiculo['fcompra'];
$garantia = $vehiculo['garantia'];
$ahora = date("dmY",strtotime("-$garantia year"));
$fechacompra = fecha_normal($fcompra);
$diasgarantia = resta_fecha($ahora,$fechacompra);
}

\begin{lstlisting}[language=php,title={/var/lib/asterisk/agi-bin/fingarantia.php}]

$query = mysql_query("SELECT * FROM clientes WHERE dni = '$dni'");

```

Finalizamos devolviendo al flujo de la llamada una variable con los días de garantía obtenidos

/var/lib/asterisk/agi-bin/fingarantia.php

```

$agi->verbose($diasgarantia);

$agi->set_variable("textotts",\$diasgarantia);
?>

```

Por otro lado necesitamos definir algunos scripts específicos para tratar el sistema Automatic Speech Recognition y Text-To-Speech, basados también en AGI. Podemos encontrar los scripts en el Anexo.

Ahora solo faltaría definir el IVR dentro de nuestro DialPlan. Para ello hemos creado una extensión específica especial que cumpliera dicho proposito, ejecutando las aplicaciones oportunas según descrito a continuación:

(/etc/asterisk/extensions.conf

```

/ IVR de GARANTIAS
exten => 105,1,Answer()
same => n,Set(contador=1)
// Recogida de Informacion
same => n(repetir),GotoIf($[${contador}>2]?operador)
same => n,AGI(asterisk-tts.php,"Diga los numeros
de su DNI y pulse almohadilla")
same => n,Playback(google-tts)
same => n,Record(grabacion-asr.wav,2)
same => n,AGI(asterisk-asr.php)
same => n,AGI(asterisk-tts.php,"Su DNI es")
same => n,Playback(google-tts)
same => n,AGI(asterisk-tts.php,${TEXTTOASR})
same => n,Playback(google-tts)
same => n,AGI(asterisk-tts.php,"Marque 1 si es correcto")
same => n,read(varok,google-tts,1)
same => n,Set(contador=${[${contador}+1]})
same => n,GotoIf(\${[${varok}]!=1}?repetir)

```

La secuencia sería, como tambien puede verse en la figura 5.19

1. Respondemos la llamada
2. Tenemos un contador para evitar que el cliente se pase indefinidamente intentado decir su DNI, en caso que lo haga dos veces sin éxito, le pasaríamos con un operador
3. Lanzamos mensajes utilizando el sistema TTS de Google, ya no necesitamos grabar los mensajes expresamente, de forma estática
4. Recogemos la grabación, tanto si marca la almohadilla (#) como si pasan dos segundos de silencio
5. Lanzamos la aplicación AGI asterisk-tts para convertir un texto a audio, y reproducimos el mismo
6. Recogemos los números del DNI utilizando la aplicación AGI asterisk-asr
7. Seguimos lanzando la aplicación AGI TTS y la aplicación para reproducir todos los textos y los números del DNI.
8. Confirmamos con el cliente si es correcta la grabación de los números. En caso que no lo sea, volvemos a empezar, si lo es, continuamos a la siguiente parte

(/etc/asterisk/extensions.conf

```
// Comprobacion de Garantia
same => n,AGI(fingarantia.php,${TEXTTOASR})
same => n,GotoIf($[${texttotts}<1]?caducada)
same => n,AGI(asterisk-tts.php,"Le quedan")
same => n,Playback(google-tts)
same => n,AGI(asterisk-tts.php,${texttotts})
same => n,Playback(google-tts)
same => n,AGI(asterisk-tts.php,"dias de garantia")
same => n,Playback(google-tts)
same => n,Hangup()
same => n(caducada),AGI(asterisk-tts.php,"La garantia ha caducado")
same => n,Playback(google-tts)
same => n,Hangup()
```

La secuencia seria:

1. Ejecutamos el script fingarantia.php descrito anteriormente.
2. En caso que la garantía sea inferior a 1 día, lanzaríamos la reproducción de garantía caducada
3. En caso que no, lanzaríamos reproducciones utilizando el sistema TTS para dar el número de días restantes y terminaría

(/etc/asterisk/extensions.conf

```
// Salto a Extensiones Posventa
same => n(operador),AGI(asterisk-tts.php,"Le pasamos con un operador")
same => n,Playback(google-tts)
same => n,Goto(extensiones,400,1)
```

Finalmente, en caso que hubieran pasado 2 intentos de reconocer el DNI con el sistema ASR sin éxito, lanzaríamos la llamada al grupo de líneas de Posventa. Para definir el grupo de líneas de Posventa, necesitamos hacer algunas modificaciones. Por un lado vamos a crear una variable globales, que agrupe los dispositivos SIP de las extensiones actuales:

(/etc/asterisk/extensions.conf

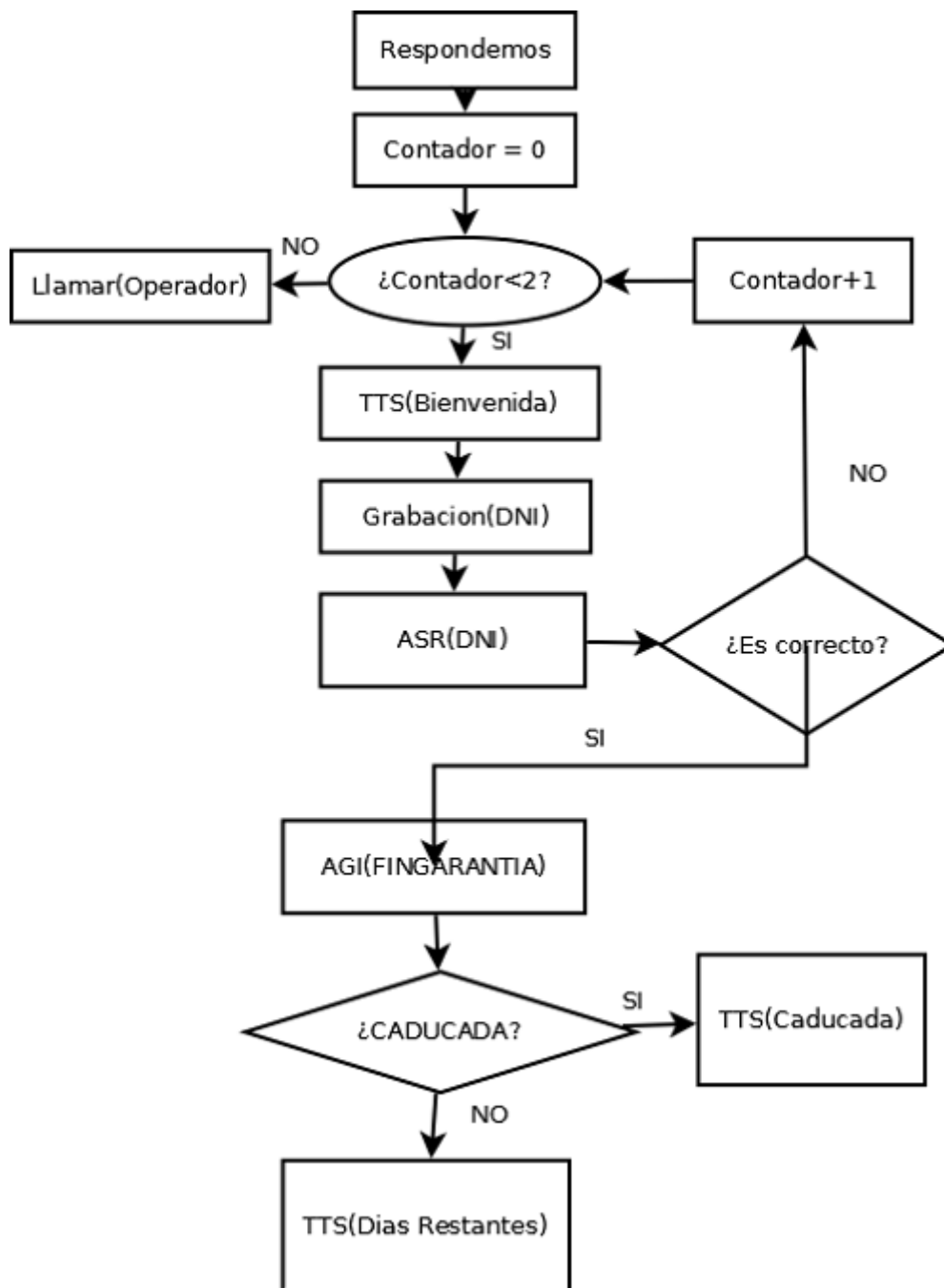


Figura 5.19: Diagrama de una llamada en el sistema a IVR

```
[globals]
```

```
GRUPOPOSVENTA=SIP/41&SIP/42&SIP/43
```

Por otro lado, definiríamos en el contexto [extensiones] la extensión 400 a la que hacemos referencia, utilizando esta variable global declarada.

```
(/etc/asterisk/extensions.conf
```

```
; Extensiones Grupo Posventa
exten => 400,1,Answer()
same => n,Dial(\${GRUPOPOSVENTA},30,m)
same => n,VoiceMail(41)
same => n,Hangup()
```

Finalmente vamos a necesitar de un último DDI, que dará la opción de hablar con el grupo Posventa, o saber la garantía. Para ello vamos a crear otra pequeña Operadora Automática en el DDI número 956001161, que se desarrollará de la siguiente forma:

(/etc/asterisk/extensions.conf

```
exten => 956001161,1,NoOp()
same => n,AGI(asterisk-tts.php,"Si quiere saber el tiempo
restante de su garantía marque el 1")
same => n,Playback(google-tts)
same => n,AGI(asterisk-tts.php,"en otro caso, espere y le
pondremos con un operador")
same => n,Read(opciones,google-tts,1)
same => n,GotoIf($[${opciones}=1]?especiales,105,1:extensiones,400,1)
```

Con esto ya tendríamos el sistema IVR listo para ser entregado a nuestro cliente UCA Autos.

5.6.2. Respuesta de Voz Interactiva

Muy parecido a la sección donde configurábamos una operadora automática basada en diferentes estados y diferentes combinaciones, en este caso, tendríamos que realizar las mismas comprobaciones llamando al a extensión 105 y observar que se efectúan correctamente, siguiendo la secuencia de estados mostrado en la figura 5.19

Aparte llamando al número directo 956001161, deberíamos de poder probar también si en caso de no marcar nada, nos desvía a una extensión del grupo de Posventa siempre y cuando tengamos como mínimo uno de los teléfonos configurados de ese grupo (4X). En caso que marquemos la opción 1, deberíamos de poder escuchar el menú IVR correspondiente.

Si en la base de datos tenemos configurados varios usuarios y vehículos con diferentes fechas de compra podremos comprobar además que el sistema AGI efectúa la comprobación de forma correcta. Incluso podríamos probar a marcar un DNI erróneo para comprobar que también los fallos están contemplados como posibles combinaciones.

En este caso, los ejemplos de DNI son 79253592P con una garantía en vigor hasta el año 2013, y 12345678Z con una garantía caducada.

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

Tras mostrar a la comunidad de Asterisk el proyecto, y recibir una amplia aceptación ha resultado una gratificación en términos generales. Desde el inicio del proyecto de investigación y la puesta en marcha del Blog de Difusión 10000 Horas 6.3 ya podían observarse múltiples atisbos de interés. Las estadísticas hablan hoy y tengo firme convicción que el proyecto no ha hecho más que empezar.

Poder realizar un primer proyecto a nivel formal que integra múltiples mecanismos de desarrollo y documentación ha sido una idea interesante planteada desde el inicio por el tutor Manuel Palomo,

6.1.1. Sobre WIKIAsterisk

La estructura esta diseñada, para servir como una guía práctica para el lector, y que dado el caso, pudiera seguirse como un manual de referencia. Existen guías y documentos en formato papel, y no se descarta en un futuro poder editar la misma si así resultase en una difusión aún más distendida.

Podríamos considerar que el desarrollo sigue un camino, iniciando al lector en los aspectos generales, pasando por aspectos prácticos como la instalación, y conceptuales generales, como el aprendizaje de la generación y creación de un plan de marcación, desde niveles básicos a niveles más sofisticados. A partir de ahí se vuelcan todas las herramientas en orden de importancia relativa en forma de artículos y bajo mi visión personal puede servir como traza para el lector pueda ir tomando los aspectos que más le interesen en función de sus necesidades.

El aspecto visual me resulta impecable y a diferencia de un libro, el cual suele estar basado en un índice, y recorrer las páginas aun intuitivo puede resultar engorroso, todo el sistema referencial que aporta un sistema WIKI aporta una potencia que hasta la fecha, no pensaba que pudiera resultar tan potente.

Particularmente tengo que reconocer que con diferencia, lo que más me ha sorprendido ha sido la facilidad con la que podremos crear referencias de un punto a otro de la WIKI, y la facilidad con la que podemos entrelazar en consecuencia los artículos. Un caso curioso, podría decir, que el Artículo que habla sobre SIP es de lejos, el artículo más referenciado en toda la WIKI dado que como explicaba en la Introducción, realmente es el protocolo de Telefonía IP más distendido en la actualidad y prácticamente todos los aspectos de Asterisk hacen referencia al mismo.

6.1.2. Método del Caso y Caso de Estudio

Una de las cosas que tenía clara desde el primer momento, era diseñar una metodología de explicación de los contenidos prácticos de una forma que fuera lo más apartada de lo convencional posible, que

hasta la fecha, me había resultado algo escabrosa especialmente en el despliegue relacionado a sistemas informáticos.

Además ha servido para mi por primera vez, en aplicarlo de manera “formal” y quizá me plantearía en un futuro una prueba piloto para aplicarlo con carácter docente extraoficialmente. Me ha sorprendido gratamente, que habiendo utilizado un caso, basado en una mediana empresa dentro del sector de la automoción al cual me he sentido fuertemente asociado hasta una fecha relativamente cercana, se me han ido ocurriendo problemas de una manera prácticamente espontánea, y me veía aprovechando mis ratos libres, para idear de manera ingeniosa, como podía desarrollar un planteamiento práctico, en un supuesto caso real de implantación, para que resultara lo más creíble y realista posible.

En términos generales, puedo afirmar ahora que el caso ha quedado bastante completo y cubre todos los aspectos generales como para poder integrar a un usuario en Asterisk, enfocándolo en el marco conceptual de las Comunicaciones Unificadas.

La prueba de fuego puede que realmente se encuentre, en medir con terceras personas si realmente este sistema pudiera ser lo suficientemente didáctico para implementarse de como manera dinámica de implantar una metodología de enseñanza en las formaciones relativas a Asterisk y otros sistemas semejantes.

6.1.3. Máquina Virtual

Considerando las limitaciones de Hardware tuve que plantearme como desplegar un caso práctico completo, y funcional para demostrarme a mi mismo que era posible integrar, probar y desarrollar toda la funcionalidad de Asterisk a pequeña escala en mi domicilio particular. Al principio, pensaba que con una máquina virtual iba a poder cubrir todo la práctica, pero al poco tiempo surgió la necesidad de crear una segunda máquina virtual, y que afortunadamente, podía convivir sin problemas de rendimiento dentro del mismo servidor, y a la necesitábamos considerar como “Proveedor de telefonía IP” o ITSP (Internet Telephony Service Provider).

Con esto y en esencia, todo quedaba cubierto y la máquina del Proveedor, quedaría como una “Caja Negra” para demostrar como la interoperabilidad con el mismo, no era dependiente a excepción de los parámetros de configuración que un supuesto Proveedor de Telefonía IP nos proporcionase con el contrato.

Este tipo de configuraciones son muy comunes en los cursos de Asterisk, donde el ordenador del docente, sirve como “Proveedor de Telefonía” y los equipos de alumnos son las máquinas Asterisk que interoperan con el mismo.

En general este planteamiento ya lo tenía tan asimilado desde el primer momento, que no me resulto demasiado complejo, la implementación de sucesivas mejoras para adaptar las necesidades del proyecto a cada momento presente en la línea del tiempo.

6.2. Dificultades Técnicas Superadas

La familiarización con herramientas nuevas, suele ser una tarea ardua y compleja según el momento en el que estamos tratando de lidiar ellas.

Debo reconocer que uno de los aspectos que más tiempo me costó integrar fue en el uso de las mismas, principalmente las enfocadas a la documentación, como fueron el manejo de sistemas WIKI basados en MediaWiki, y la documentación a través de L^AT_EX.

Creo que esto suponía más un problema de falta de adecuación, y conceptual, en el sentido que a priori no entendía los principios básicos de estas herramientas considerando que estaba acostumbrado a utilizar otros medios más distendidos (en general hablando de sistemas WYSIWYG que a pesar que de mayor o menor medida).

Pasado el primer mes de uso intensivo de estas herramientas pude observar como mis dificultades fueron quedado atrás, y acepté que realmente para escribir un documento no era necesario que resultara estéticamente vistoso desde el primer momento algo que hasta la fecha era totalmente ajeno a mi experiencia.

Por otro lado en cuanto a lo que el desarrollo del proyecto hace referencia debo reconocer, que a pesar que me costó tiempo entender el funcionamiento del sistema Asterisk, gracias a la formación específica, y el tiempo que permanecí en la primera fase en la que se desarrollo todo el proceso de aprendizaje e investigación en la materia, facilitó de una gran manera el poder diseñar el grueso del proyecto técnico en un intervalo limitado de tiempo aceptable.

Afortunadamente, el uso de lenguajes de programación aplicados en la materia, como PHP [13] [10], durante el transcurso del desarrollo, ha sido uno de mis puntos fuertes, lo que no me supuso mayor dificultad, que aprender a integrar este conocimiento dentro de las posibilidades que me ofrecía Asterisk a nivel de Interfaz.

Finalmente, también debo destacar que al final, el hecho de haber conseguido integrar todas las herramientas aquí utilizadas, me ha servido para dar un paso adelante en varios niveles, y en la actualidad ya me esta sirviendo para otros ámbitos foráneos al presente proyecto, quizá resultado de la ventaja de incorporar tecnologías eficientes en nuestro conocimiento.

6.3. Planteamientos Futuros

Durante la creación de este proyecto, iban surgiendo ideas nuevas, sobre posibles planteamientos que pudieran llevarse a corto, medio y largo plazo. De forma generalizada, podría decirse que la mayor parte de los planteamientos que surgieron a corto, decidieron llevarse adelante, aún no estando previstos dentro de la idea original, y exceptuando contados casos, tuvieron que dejarse a un lado. La mayor parte de estos conceptos a corto, están fundamentados en determinadas secciones de la WIKI, que aún comentadas por encima, deberían quedar más detalladas en profundidad, dado la repercusión que pueden suponer en ofrecer el máximo de contenidos de calidad. En realidad quizá haya sido por cuestiones de previsiones de tiempo que alguna instancia se hubiera quedado en el tintero.

A medio Plazo, mis ideas se enfocan principalmente en un tema que surge levemente en varias ubicaciones de este Proyecto, y tiene que ver con una plataforma de relativa reciente creación, llamada Asterisk Communications Framework (Asterisk SCF), y esta basada en la idea de poder desplegar una arquitectura distribuida, que ofrezca la flexibilidad semejante al P2P de Skype, pero basada exclusivamente en sistemas de Servidores escalables. Esto esta más orientado a Operadoras, y al nivel Carrier-Grade [41] ya que se observó con el tiempo, que Asterisk podía convertirse en un cuello de botella, dada su baja escalabilidad y era frecuente recurrir a SIP Proxys [11] fuera de Asterisk específicos que si ofrecían estas posibilidades de escalabilidad de forma nativa como Kamailio [27]

A largo Plazo, mi pretensión se encuentra focalizada, en poder desarrollar una WIKI adjunta a WIKIAsterisk.com, pero exclusivamente basada en el entorno de desarrollo de Asterisk, para ofrecer una introducción y profundización paulatina, y poder extender la comunidad de desarrollo a más ámbitos. Además al igual que la WIKI original, esta originalmente orientada a mostrar todo el contenido disponible para prepararse cara a la certificación **Digium Certified Asterisk Professional** (dCAP) según noticias de Digium, plantean lanzar una certificación **Digium Certified Asterisk Developer**, que irá enfocada a los desarrolladores del mismo.

Además como se comenta en el capítulo de Difusión 6.3, otro de los aspectos obligados, es la creación de comunidad, dado que la verdadera eficiencia de un sistema de este tipo es difícilmente manejable por una única persona, dado que los sistemas se actualizan regularmente, van surgiendo novedades en varios niveles y conceptos, y esto repercute en una “desactualización” general, que acaba degenerando en la obsolescencia de la creación. A no ser que en este caso, yo personalmente, me encontrará exclusivamente enfocado en el entorno de Asterisk, cosa que resultase poco probable, lo más típico sería que pequeñas aportaciones de mucha gente fueran similares, a una gran aportación de una sola persona como es este proyecto.

Finalmente uno de los aspectos más interesantes que también han quedado abiertos, es el desarrollo de Asterisk para otras plataformas diferentes a Linux, y concretamente a Ubuntu Server/Debian, el cual ha estado muy enfocado el proyecto en general. Una de los grandes atributos de Asterisk es que se integra en todo tipo de sistemas basados en la plataforma UNIX perfectamente.

Software utilizado

MiKTeX + T_EXnic Center

Debo reconocer que por cuestiones de la vida para desarrollar este proyecto he trabajado principalmente bajo un entorno Microsoft Windows. Aún así esto no quita para utilizar grandes herramientas Open Source como son el caso de la distribución T_EX para Windows llamada MiKTeX[29].

Por otro lado, para la edición existe un IDE de L^AT_EX muy popular en Windows y bastante completo a mi parecer, llamado T_EXnic Center (Figura 6.1, el proyecto se mantiene en la conocida web SourceForge [9], posee un licenciamiento GPL también y se integra con MiKTeX a la perfección.

Finalmente para la configuración de este documento, he utilizado una plantilla específica para PFC que me ha proporcionado mi Director de Proyecto, y para la documentación referente a las guías de edición, he de hacer referencia a la página web que me ha ido guiando en el proceso. [24]. Gracias a esta guía he introducido cambios generales de estructura para optimizarla un poco más y facilitarme el trabajo en general.

Inkscape + GIMP + DIA

Siguiendo un poco en la Línea de Aplicaciones Open Source, para editar las edición de imágenes he utilizado principalmente estos. Hay que considerar que los efectos tampoco han sido de gran nivel, dado que mis habilidades no se encuentran especialmente desarrolladas en este ámbito.

GIMP [35] ha sido principalmente práctico, para desarrollar los logotipos, adaptar imágenes al tamaño de la memoria y corregir algunos defectos en las imágenes de poca repercusión

Inkscape [12] ha resultado muy útil para componer gráficos y esquemas explicativos dada su naturaleza de diseño vectorial esta especialmente orientado a esta faceta.

Para algunos gráficos concretamente los diagramas de flujo la herramienta que he utilizado ha sido DIA [5].

Softphones: Zoiper + X-Lite

Un Softphone hace referencia a Software Telephone, es decir, un teléfono basado en Software. Desafortunadamente, debo decir que en contrapartida, trabajando en un entorno Windows, estos en formato Software de Código Libre no son especialmente funcionales. Dado que el protocolo SIP ha pasado por más de 10 drafts antes de convertirse en un RFC medianamente estandarizado, existen muchas implementaciones y disposiciones específicas en el intercambio de mensajes, que pueden resultar fatales para

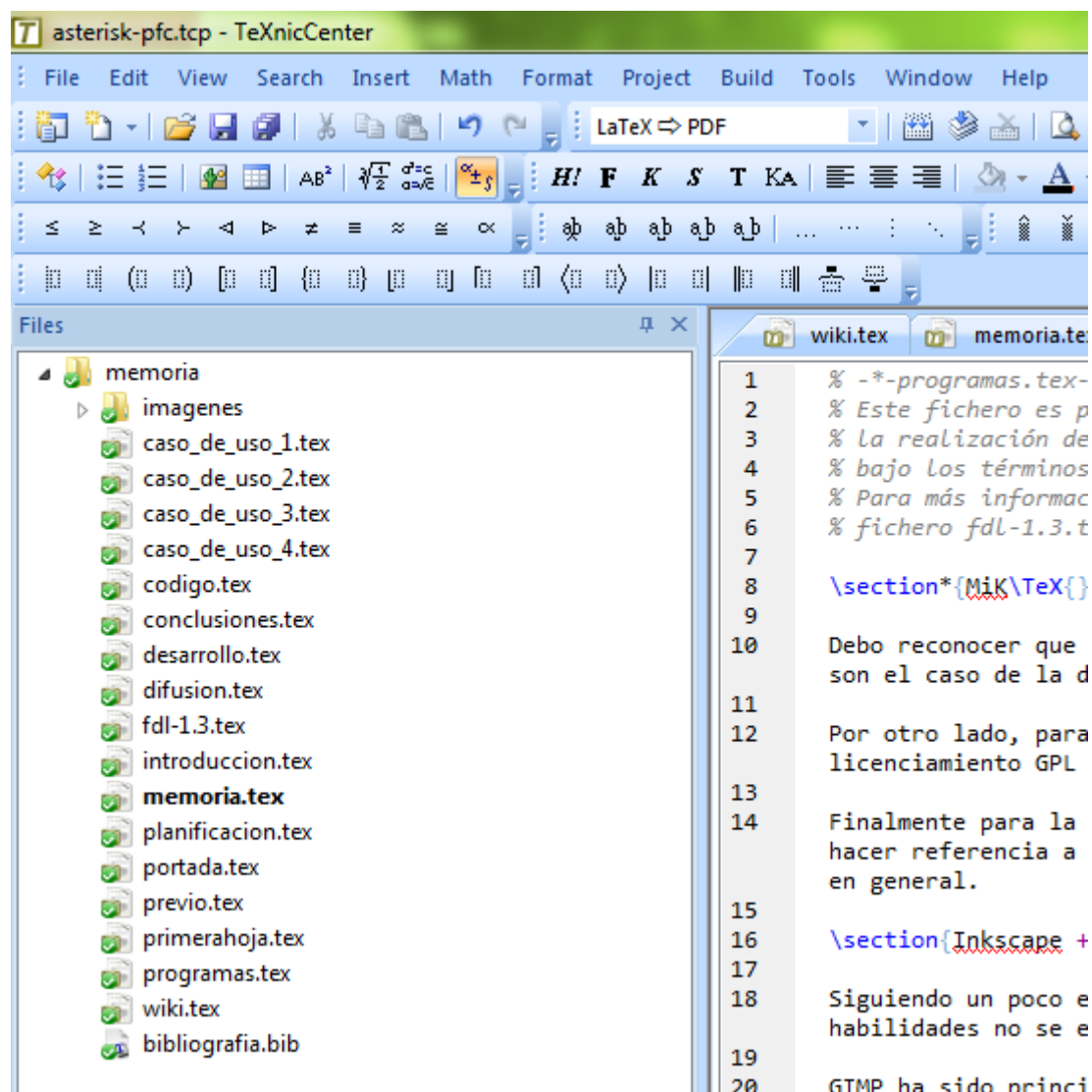


Figura 6.1: Captura del Software T_EXnic Center

ciertos programas. Está altamente desaconsejado utilizar Softphones de bajo desarrollo, y justamente los softphones Open Source en el entorno Windows, suelen ser malos “ports”¹ de su versión original en Linux.

Yendo más allá, justamente Zoiper [21] (Figura 6.2) y X-Lite [4] son versiones básicas de uso para consumo a nivel particular con niveles muy restringidos de uso, pero totalmente funcionales a nivel de llamadas. Podrían considerarse poco más que un teléfono básico físico a nivel de prestaciones, pero cumplirían bien su objetivo cara al desarrollo de este proyecto.

PuTTY

Este programa es por excelencia la puerta de acceso a todos los servidores Linux más reconocido en el entorno Windows. Permite el acceso a consola utilizando el sistema Secure Shell (SSH). Haciendo

¹La denominación de recompilar el código fuente originalmente ideado para un Sistema Operativo específico, en otro sistema diferente



Figura 6.2: Captura del Softphone Zoiper

una búsqueda con los terminos SSH y Windows en Google es curioso observar como la mayor parte de los enlaces llevan directamente a este programa lo que podría considerarlo el Software más popular dentro de este sector. Yo personalmente llevo utilizando por necesidad desde hace bastantes años, y por continuidad durante el desarrollo de este proyecto también he requerido su uso

MinGW

Sus siglas hacen referencia a Minimalist GNU for Windows [33], es decir, una plataforma para Windows que trae una gran cantidad de aplicaciones destacadas de GNU, basadas en librerías propias de la misma, y especialmente orientadas a entornos de programación. Concretamente yo he dado uso a esta herramienta para poder hacer uso del cliente GIT [32], para conectar al repositorio GIT de GitHub.com donde se almacena toda la estructura de ficheros relacionados directamente con este proyecto.

No he observado ninguna carencia con respecto al cliente GIT nativo de Linux, ni siquiera en el intercambio de claves RSA, con lo cual supone una gran alternativa en caso que estemos forzamos a utilizar un sistema Windows como ha resultado mi caso específico.

MinGW establece una interfaz tipo shell, equivalente a cualquier otra Consola GUI ofrecida en entornos de escritorio Linux como puede observarse en la figura 6.3

```

$ git pull origin master
Enter passphrase for key '/c/Users/mcamargo/.ssh/id_rsa':
remote: Counting objects: 25, done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 17 (delta 12), reused 13 (delta 8)
Unpacking objects: 100% (17/17), done.
From github.com:SirLouen/Asterisk-PFC
 * branch                master      -> FETCH_HEAD
Updating a4c8596..4ccc9fb
Fast-forward
 conf/extensions.conf      | 53 ++++++++
 scripts/asterisk-asr.php  |  2 +-
 scripts/asterisk-tts.php  |  4 +-
 scripts/{clave_md5 => clave_md5.sh} |  0
 scripts/fingarantia.php   | 63 ++++++++
 .../{permisos_asterisk => permisos_asterisk.sh} |  0
6 files changed, 103 insertions(+), 19 deletions(-)
rename scripts/{clave_md5 => clave_md5.sh} (100%)
rename scripts/{permisos_asterisk => permisos_asterisk.sh} (100%)
mcamargo@PCMANUEL /f/Dropbox/Dropbox/PFC/repositorio (master)
$

```

Figura 6.3: Captura de la Plataforma MinGW ejecutando comandos GIT

Difusión de WIKIAsterisk

Blog de Investigación: 10000 Horas

El paso previo a la creación de WIKIAsterisk fue el proceso de generación de un Blog con fines de Investigación y Desarrollo con el sistema Asterisk. [3]. Una de las primeras inconveniencias que encontré, era la dificultad de entender el funcionamiento de un sistema de Telecomunicaciones y enfocarlo más como un sistema Informático. Pensé que si desarrollaba mi conocimiento paso a paso, podría resultar útil en un futuro a un recién llegado, o incluso en este caso, me serviría incluso de autoreferencia para apoyar este proyecto.

En la actualidad recibe aproximadamente unas 80-100 visitas diarias como puede verse en la Figura 6.4, curiosamente, incluso en Google Reader posee varios subscriptores, pese a no ser un Blog demasiado activo.

Eventualmente, también me ha servido para recibir feedback por parte de la comunidad en mis desarrollos y así poder perfeccionarlos y depurarlos hasta conseguir niveles de utilidad aceptables. Además se han tratado temas que nunca fueron difundidos antes, y ha servido para sentar algunas bases y como punto de conocimiento para temas concretos.

La idea de ponerle 10000 Horas, fue por el hecho que existe un concepto que basa la idea, que se necesitan esas horas para conseguir la maestría en algo. Yo pretendía alcanzar la maestría en Asterisk pasando esas horas delante del sistema, y gracias al blog, mejorar la curva de aprendizaje de futuros lectores, para reducir la cantidad de tiempo a invertir drásticamente.

	Total	Mes anterior May, 2012	Este mes Jun, 2012	Target Este mes Jun, 2012	Ayer 12 Jun, 2012
■ Visitantes	17854	1784	784 (-56.1%)	1809 (+1.4%)	127
■ Páginas vistas	85033	6121	2168 (-64.6%)	5003 (-18.3%)	317
■ Spiders	27315	2002	835 (-58.3%)	1927 (-3.7%)	124
■ Feeds	3605	120	46 (-61.7%)	106 (-11.7%)	7

Figura 6.4: Estadísticas de acceso del Blog 10000horas

Estadísticas de Acceso WikiAsterisk

Para poder conocer las estadísticas de acceso del sistema MediaWiki conseguí dar con una aplicación complementaria llamada Firestats [52] (Figura 6.5 bastante completa y mucho más intuitiva y versátil

que las propias extensiones de Mediawiki que podían agregarse (y la pobre interfaz de estadísticas que incluye de serie) [51]

Con esto podríamos en tiempo real, el impacto producido por el trabajo realizado a nivel de Difusión 6.3.

La familiarización con herramientas nuevas, suele ser una tarea ardua y compleja según el momento en el que estamos tratando de lidiar con las mismas. Uno de los puntos en el que más tiempo costó integrarme fue en el uso de las mismas, principalmente las enfocadas a la documentación, como fueron el manejo de sistemas WIKI basados en MediaWiki, y la documentación a través de Latex.



Figura 6.5: Estadísticas WIKIAsterisk con Firestats

Certificaciones Digium

Durante el transcurso del desarrollo de la WIKI y finales de la primera fase de investigación, tuve la oportunidad de obtener la primera y más básica certificación de Digium, **Digium Certified Asterisk Administrator**, la cual está orientada a ofrecer a demostrar competencias básicas pero no esta ampliamente reconocida por el sector.

En segundo y último nivel en la actualidad, se encuentra el Digium Certified Asterisk Professional (Figura 6.6), el cual demuestra alta competencia en el manejo del sistema Asterisk, y está ampliamente reconocida por todos los operadores de Telefonía a nivel mundial, más considerando que Asterisk esta siendo implantada no solo a niveles puros, sino incluso, existen referencia de sistemas Asterisk funcionando bajo el nombre de PBX de marcas reconocidas. Este auge podría suponer un gran interés especialmente en la comunidad orientada al mercado de las telecomunicaciones, de empezar a interesarse cada vez mas en Asterisk, y todo este proyecto, servir como medio para canalizar, ese interés especialmente dentro del entorno hispanohablante.

Una de las ideas concebidas tras el surgimiento de la WIKI era enfocarla para cubrir todos los aspectos relevantes de la certificación dCAP y aprovechar la comunidad para dar cabida a toda la información recursiva necesaria para ser una fuente de información lo suficientemente fidedigna para servir como punto de referencia.



Figura 6.6: Logotipo de la Certificación dCAP

Anexo I: Código Referenciado

Ficheros de Configuración

Protocolo SIP

Antes de Convertirlo a configuración RealTime:

/etc/asterisk/sip.conf

```
[general]
; Nuestro idioma
language = es
; Cambiamos el puerto para ofrecer seguridad por ocultacion
bindport = 35060
allowguest = no

register => ucaautos:12345678@home-asterisk.local:35060

[proveedorip]
defaultuser = ucaautos
fromuser = ucaautos
secret = 12345678
host = home-asterisk.local
port = 35060
type = friend
insecure = invite
qualify = yes
context = entrantes

[telefonos](!)
; Vamos a impedir el Transcoding utilizando solo el codec Alaw
disallow = all
allow = alaw
allow = ulaw
dtmfmode = rfc2833
host = dynamic
qualify = yes
type = friend

[gerencia](!)
context = gerencia

[manager](!)
context = manager
```

```

[resto](!)
context = extensiones

; 1 - Administracion
; 2 - Comercial
; 3 - Almacen
; 4 - Posventa
; 5 - Marketing

; Gerente
[10] (telefonos, gerencia)
md5secret=2b45660a7b1155943f0132f05bd0e34d
mailbox = 10@default

; Responsable Administracion
[11] (telefonos, manager)
md5secret=6acb3eb4a4939b22ea6709e87ec79311
mailbox = 11@default

[12] (telefonos, resto)
md5secret=bb24a2de8a36fdd14e358493cd5abac5
mailbox = 12@default

[13] (telefonos, resto)
md5secret=501b420ea96938f8ba34ecce52a4c654
mailbox = 13@default

[14] (telefonos, resto)
md5secret=6745f8a4a34c38a2f6b554334bb76483
mailbox = 14@default

; Responsable Comercial
[21] (telefonos, manager)
md5secret=5dbe664c6d998967ad9bfe4ea37521a1
mailbox = 21@default

[22] (telefonos, resto)
md5secret=2f9ddd39a552c3c7709f07b288cf1db3
mailbox = 22@default

[23] (telefonos, resto)
md5secret=0a4bb4d4b4f92ba79f48b28a706bbe7b
mailbox = 23@default

[24] (telefonos, resto)
md5secret=6736a7be4a1ef7a6f111441d5d59187e
mailbox = 24@default

[25] (telefonos, resto)
md5secret=10f36f1ed829ec2a8fac56d0b9c3080d
mailbox = 25@default

[31] (telefonos, manager)
md5secret=400e3d6f04bc6e13bb9a58f0f53c1a26
mailbox = 31@default

```

```

[32] (telefonos, resto)
md5secret=cafff9208a95aa83357ae5e5d3949a681
mailbox = 32@default

[33] (telefonos, resto)
md5secret=58a1e10c20da84d4c323f5018b1f7f26
mailbox = 33@default

[34] (telefonos, resto)
md5secret=902dc9faff73e90187292a036a8c1d86
mailbox = 34@default

[41] (telefonos, manager)
md5secret=21c85ea8cc8b45b430742f1cf9eb55bb
mailbox = 41@default

[42] (telefonos, resto)
md5secret=d7e2c8ec48e1647412503d82659dd9d2
mailbox = 42@default

[43] (telefonos, resto)
md5secret=988cd566697d6310e7a26f21ecc7007f
mailbox = 43@default

[51] (telefonos, manager)
md5secret=2bbcc664bdf4e1bb29cd7f1bdbf91327
mailbox = 51@default

[52] (telefonos, resto)
md5secret=0c32d549b9c60fa6a7df077e4bb545e5
mailbox = 52@default

[53] (telefonos, resto)
md5secret=7b099d95a97bb4c8d6a1d08c7c047167
mailbox = 53@default

[54] (telefonos, resto)
md5secret=037f3c092ae05f73054d2d2dd0e7d5dc
mailbox = 54@default

```

Después de Convertirlo a configuración RealTime:

/etc/asterisk/sip.conf

```

[general]
; Nuestro idioma
language = es
; Cambiamos el puerto para ofrecer seguridad por ocultacion
bindport = 35060
allowguest = no
register => ucaautos:12345678@home-asterisk.local:35060
srvlookup = yes
; Parametro especifico de Asterisk RealTime
rtcachefriends = yes

```

Plan de Marcación

/etc/asterisk/extensions.conf

```
[globals]

GRUPOPOSVENTA=SIP/41&SIP/42&SIP/43

[especiales]

exten => 101,1,NoOp()
same => n,Playback(bienvenida)
same => n,Playback(grabacion)
same => n,Set (GRABACION="grabacion-${STRFTIME (${EPOCH},, %Y %m %d- %H %M %S) })
same => n,Monitor (wav, ${GRABACION},mb)
same => n,Queue (almacen)
same => n,Hangup()

exten => 102,1,NoOp()
same => n,MeetMe()
same => n,Hangup()

exten => 103,1,NoOp()
same => n,MeetMe(100)
same => n,Hangup()

// IVR de GARANTIAS
exten => 105,1,Answer()
same => n,Set(contador=1)
// Recogida de Informacion
same => n(repetir),GotoIf($[${contador}>2]?operador)
same => n,AGI(asterisk-tts.php,"Diga los
numeros de su DNI y pulse almohadilla al terminar")
same => n,Playback(google-tts)
same => n,Record(grabacion-asr.wav,2)
same => n,AGI(asterisk-asr.php)
same => n,AGI(asterisk-tts.php,"Su DNI es")
same => n,Playback(google-tts)
same => n,AGI(asterisk-tts.php,${TEXTTOASR})
same => n,Playback(google-tts)
same => n,AGI(asterisk-tts.php,"Marque 1 si es correcto")
same => n,read(varok,google-tts,1)
same => n,Set(contador=${[${contador}+1]})
same => n,GotoIf($[${varok}!=1]?repetir)
// Comprobacion de Garantia
same => n,AGI(fingarantia.php,${TEXTTOASR})
same => n,GotoIf($[${texttotts}<1]?caducada)
same => n,AGI(asterisk-tts.php,"Le quedan")
same => n,Playback(google-tts)
same => n,AGI(asterisk-tts.php,${texttotts})
same => n,Playback(google-tts)
same => n,AGI(asterisk-tts.php,"dias de garantia")
same => n,Playback(google-tts)
same => n,Hangup()
same => n(caducada),AGI(asterisk-tts.php,"La garantia ha caducado")
```

```

same => n,Playback(google-tts)
same => n,Hangup()
// Salto a Extensiones Posventa
same => n(operador),AGI(asterisk-tts.php,"Le pasamos con un operador")
same => n,Playback(google-tts)
same => n,Goto(extensiones,400,1)

; --- Macro para Extensiones
; --- Con Buzones de Voz Mejorados
[macro-telefonos]
exten => s,1,NoOp()
same => n,Set(buzon=${MACRO_EXTEN})
same => n,Set(CHANNEL(tonezone)=starwars)
same => n,Dial(SIP/${buzon},30)
same => n,Gotoif("${DIALSTATUS}" = "BUSY"?ocupado)
same => n,VoiceMail(${buzon},u)
same => n,Hangup()
same => n(ocupado),VoiceMail(${buzon},b)
same => n,Hangup()

[macro-buzon]
exten => s,1,NoOp()
same => n,Set(buzon=${MACRO_EXTEN:1})
same => n,VoiceMailMain(${buzon})
same => n,Hangup()

[macro-pausa]
exten => s,1,NoOp()
same => n,PauseQueueMember(almacen,SIP/${MACRO_EXTEN:1})
same => n,Hangup()

[macro-sinpausa]
exten => s,1,NoOp()
same => n,UnpauseQueueMember(almacen,SIP/${MACRO_EXTEN:1})
same => n,Hangup()

; --- Plan General de Extensiones
[extensiones]

include => exterior
include => especiales

; Minimizando las combinaciones para
; ofrecer justo las extensiones disponibles

; Todas las Extensiones para Asterisk RealTime
exten => _[1-5]X,1,Macro(telefonos)
exten => _9[1-5]X,1,Macro(buzon)

; Extensiones para Colas de Almacen
exten => _73[2-4],1,Macro(pausa)
exten => _83[2-4],1,Macro(sinpausa)

; Extensiones Grupo Posventa
exten => 400,1,Answer()
same => n,Dial(${GRUPOPOSVENTA},30,m)

```

```

same => n,VoiceMail(41)
same => n,Hangup()

[exterior]
; --- Contexto Especifico para Llamadas al exterior

exten => _[89]ZXXXXXXX,1,NoOp()
same => n,Dial(IAX2/proveedorrip/${EXTEN})
;same => n,Dial(SIP/proveedorrip/${EXTEN})
same => n,Hangup()

exten => _6XXXXXXX,1,NoOp()
same => n,Dial(IAX2/proveedorrip/${EXTEN})
same => n,Hangup()

exten => _90[0-2]XXXXXX,1,NoOp()
same => n,Dial(IAX2/proveedorrip/${EXTEN})
same => n,Hangup()

; --- Contexto especifico de los Managers
[manager]

include => extensiones

; --- Llamadas a numeros de informacion
exten => _118XX,1,NoOp()
same => n,Dial(IAX2/proveedorrip/${EXTEN})
same => n,Hangup()

exten => 201,1,NoOp()
same => n,ChanSpy(SIP/3)
same => n,Hangup()

; --- Contexto especifico del Gerente
[gerencia]

include => manager
include => grabaciones

; --- Los numeros en Alemania son de longitud variable
; --- Para evitar plantear todas las posibilidades
; --- Hacemos uso del comodin "."

exten => _0049.,1,NoOp()
same => n,Dial(IAX2/proveedorrip/${EXTEN})
same => n,Hangup()

; --- Otros Sistemas Adicionales de Prueba,
; --- solo puede acceder el Gerente.

exten => 1111,1,NoOp()
same => n,Progress()
same => n,MusicOnHold()

[entrantes]

```

```

;Numero General de UCA Autos
exten => 956001191,1,Goto(s,1)
;Numero del Callcenter de Almacen
exten => 956001181,1,Goto(especiales,101,1)
exten => 956001171,1,Goto(especiales,102,1)
exten => 956001172,1,Goto(especiales,103,1)
exten => 956001101,1,Goto(extensiones,10,1)
exten => 956001111,1,Goto(extensiones,11,1)

exten => 956001121,1,NoOp()
same => n,Set(CALLERID(num)={EXTEN})
same => n,Goto(extensiones,21,1)

exten => 956001131,1,Goto(extensiones,31,1)
exten => 956001141,1,Goto(extensiones,41,1)
exten => 956001151,1,Goto(extensiones,51,1)

exten => 956001161,1,NoOp()
same => n,AGI(asterisk-tts.php,"Si quiere saber
el tiempo restante de su garantia marque el 1")
same => n,Playback(google-tts)
same => n,AGI(asterisk-tts.php,"en otro caso,
espere y le pondremos con un operador")
same => n,Read(opciones,google-tts,1)
same => n,GotoIf(${ ${opciones}=1}?especiales,105,1:extensiones,400,1)

; Operadora Automatica
exten => s,1,NoOp()
same => n,Set(ivrcont=0)
same => n,Answer()
same => n(bienvenida),Playback(bienvenida)
same => n,GotoIfTime(16:00-23:30,mon-fri,*,*?abierto)
same => n,Playback(cerrado&horario)
same => n,VoiceMail(1@market)
same => n,Hangup()
same => n(abierto),Background(menu-ivr)
same => n,WaitExten(1)

exten => t,1,NoOp()
same => n,Set(ivrcont=${ ${ivrcont}+1})
same => n,GoToIf(${ ${ivrcont}<2}?s,bienvenida:0,1)

; 1 a Ventas
exten => 1,1,Goto(extensiones,21,1)
; 2 a Almacen
exten => 2,1,Goto(extensiones,31,1)
; 3 a Posventa
exten => 3,1,Goto(extensiones,41,1)
; 0 a Marketing
exten => 0,1,Goto(extensiones,51,1)

[grabaciones]

; Este contexto sirve para especificar las extensiones
; Dedicadas a Grabacion de mensajes especiales

```

```
exten => 511,1,Record(bienvenida.wav)
exten => 512,1,Record(menu-ivr.wav)
```

Protocolo IAX

/etc/asterisk/iax.conf

```
[general]
;register => ucaautos:1234@192.168.101.151
bindport=4569
srvlookup=yes
language=es
encryption=yes
; Si usamos claves RSA es conveniente desactivar
; esta opcion.
forceencryption=no
trunk=yes
requirecalltoken=yes

[ucatelecom]
type = user
host = dynamic
auth = rsa
inkeys = ucatelecom
context = entrantes

[proveedorip]
type = peer
outkey = ucaautos
username = ucaautos
host = home-asterisk.local
;host = 192.168.101.151
auth = rsa
;auth = plaintext
;secret = 1234
qualify = yes
```

Buzones de Voz

/etc/asterisk/voicemail.conf

```
;!
;! Automatically generated configuration file
;! Filename: voicemail.conf (/etc/asterisk/voicemail.conf)
;! Generator: AppVoicemail
;! Creation Date: Tue May 22 15:02:20 2012
;!
[general]
format = wav49|gsm|wav
serveremail = mcamargo@wikiasterisk.com
```



```

attach = yes
maxmsg = 50
maxsecs = 120
minsecs = 5
skipms = 3000
maxsilence = 10
silencethreshold = 128
maxlogins = 2
pbxskip = no
fromstring = VoiceMail Asterisk-PFC
emaildateformat = %A, %B %d, %Y at %r
minpassword = 4
backupdeleted = 20

; Parametros por defecto del sistema de mensajes de Asterisk
[zonemessages]
eastern = America/New_York|'vm-received' Q 'digits/at' IMp
central = America/Chicago|'vm-received' Q 'digits/at' IMp
central24 = America/Chicago|'vm-received' q 'digits/at' H N 'hours'
military = Zulu|'vm-received' q 'digits/at' H N 'hours' 'phonetic/z_p'
european = Europe/Madrid|'vm-received' a d b 'digits/at' HM

[default]
10 => 10,John Doe,jdoe@wikiasterisk.com,,tz=european|
dialout=extensiones|sendvoicemail=yes|callback=extensiones|
review=yes|forcename=yes|forcegreetings=yes
11 => 11,Robert Mackencie,rmackencie@wikiasterisk.com,,tz=european|
dialout=extensiones|sendvoicemail=yes|callback=extensiones|
review=yes|forcename=yes|forcegreetings=yes
21 => 21,Hugh Douglas,hdouglas@wikiasterisk.com,,tz=european|
dialout=extensiones|sendvoicemail=yes|callback=extensiones|
review=yes|forcename=yes|forcegreetings=yes
31 => 31,Mary Roberts,mroberts@wikiasterisk.com,,tz=european|
dialout=extensiones|sendvoicemail=yes|callback=extensiones|
review=yes|forcename=yes|forcegreetings=yes
41 => 41,Philip Hudson,phudson@wikiasterisk.com,,tz=european|
dialout=extensiones|sendvoicemail=yes|callback=extensiones|
review=yes|forcename=yes|forcegreetings=yes
51 => 1234,Daniela Mayhem,sir.louen@gmail.com,,tz=european|
dialout=extensiones|sendvoicemail=yes|callback=extensiones|
review=yes|forcename=yes|forcegreetings=yes

```

Sistema de Colas

/etc/asterisk/queues.conf

```

[general]

; Opciones generales practicas
persistentmembers = yes
autofill = yes
monitor-type = MixMonitor

[almacen]

```

```

; Estrategia llamar a todos
strategy = ringall
; Intervalo de Servicio para
; estadísticas
servicelevel = 60
; Tiempo de reintento si el
; agente corta la llamada
retry = 5
; Tiempo de "reposición" del Agente
wrapuptime = 20
; Autopausa si el agente no atiende
; las llamadas
autopause = yes
; Tiempo de anuncio periódico
announce-frequency = 15
; Anuncia la posición solamente
announce-position = yes
; Tiempo mínimo entre anuncios
min-announce-frequency = 10

member => SIP/32,0,Perico
member => SIP/33,0,Francis
member => SIP/34,0,Pep

```

Sistema de Conferencias

/etc/asterisk/meetme.conf

```

[general]
audiobuffers=32
logmembercount=yes
fuzzystart=300
earlyalert=3600
endalert=120
schedule=yes

[rooms]

conf => 100,1234,4321

```

Música en Espera

/etc/asterisk/musiconhold.conf

```

[general]

[default]
mode=files
directory=moh

```

Asterisk RealTime

/etc/asterisk/res_odbc.conf

```
[asterisk_odbc]
enabled => yes
dsn => asterisk
username => asterisk
password => asterisk
pre-connect => yes

[meetme_odbc]
enabled => yes
dsn => meetme
username => asterisk
password => asterisk
pre-connect => yes
```

/etc/asterisk/extconfig.conf

```
[settings]

sippeers => odbc,asterisk_odbc,pares_sip
meetme => odbc,meetme_odbc,booking
```

/etc/odbc.ini

```
[asterisk]
Description = ODBC para MySQL
Driver = MySQL
Server = localhost
Database = asterisk
Socket = /var/run/mysqld/mysqld.sock

[meetme]
Description = ODBC para MeetMe
Driver = MySQL
Server = localhost
Database = meetme
Socket = /var/run/mysqld/mysqld.sock
```

/etc/odbcinst.ini

```
[MySQL]
Description = MySQL ODBC MyODBC Driver
Driver = /usr/lib/odbc/libmyodbc.so
Setup = /usr/lib/odbc/libmyodbcS.so
```

Generación Automática de Llamadas

/etc/asterisk/cdr_adaptive_odbcd.conf

```
[odbc_cdr]
connection=asterisk_odbcd
table=cdr
```

Interfaz AMI

/etc/asterisk/manager.conf

```
[general]
enabled = yes
port = 5038
webenabled = no
bindaddr = 0.0.0.0
allowmultiplelogin = yes
displayconnects = yes
timestampedevents = yes
authlimit = 30

[qstats]
secret = qstats
read = all
permit = 127.0.0.1/255.255.255.0
permit = 192.168.101.151/255.255.255.0

[vtiger]
secret = vtiger
read = all
write = call
permit = 127.0.0.1/255.255.255.0
permit = 192.168.101.151/255.255.255.0

[meetme]
secret = meetme
read = call
write = command,originate
permit = 127.0.0.1/255.255.255.0
```

Interfaz AGI

/etc/asterisk/phpagi.conf

```
[phpagi]
debug=true
error_handler=true
admin=sir.louen@gmail.com
hostname=pfc-asterisk.local
tempdir=/var/spool/asterisk/tmp/

[festival]
text2wave=/usr/bin/text2wave
```

```
[asmanager]
server=localhost
port=5038
username=meetme
secret=meetme

[cepstral]
swift=/opt/swift/bin/swift
voice=David
```

Scripts de Gestión

Instalación y Configuración de Asterisk

./permisos_asterisk.sh

```
#!/bin/bash
# Uso: ./permisos_asterisk usuario grupo
sudo chown -R $1:$2 /usr/lib/asterisk/
sudo chown -R $1:$2 /var/lib/asterisk/
sudo chown -R $1:$2 /var/spool/asterisk/
sudo chown -R $1:$2 /var/log/asterisk/
sudo chown -R $1:$2 /var/run/asterisk
sudo chown $1:$2 /usr/sbin/asterisk
sudo chown -R $1:$2 /etc/asterisk/
```

Protocolo SIP

./clave_md5.sh

```
#!/bin/bash
REALM=":asterisk:"
echo -n $1$REALM$2 | md5sum
```

Asterisk RealTime

/var/www/usuarios.php

```
<?php

// Conexion a la base de Datos

$base = "asterisk";
$host = "localhost";
$user = "asterisk";
$password = "asterisk";
```

```

$conexion = mysql_connect($host,$user,$password);
$result = mysql_select_db($base,$conexion)
or die ("Error en la Conexion a BD");

session_start();

// Si no estamos logeados

if (!$_SESSION['login'])
{
    if(isset($_POST['loginsubmit']))
    {
        $usuario = $_POST['usuario'];
        $password = $_POST['password'];

        if((!$usuario) || (!$password))
        {
            echo "Error 1<br>";
            exit();
        }
        $password = $usuario.":asterisk:". $password;
        $password = md5($password);
        $query = mysql_query("SELECT * FROM pares_sip WHERE
            name='$usuario' AND md5secret = '$password' AND useradmin = '1'");
        if (mysql_num_rows($query) > 0)
        {
            session_register('login');
            $_SESSION['login'] = '1';
        }
        else
            echo "Error 2<br><br>";

        echo "<a href='usuarios.php'>Home</a>";
    }
    else
    {
        echo "<form method='post' action='?'>";
        echo "Usuario: <input name='usuario' type='text'><br>";
        echo "Contrase&ntilde;a: <input name='password' type='password'><br>";
        echo "<input type='submit' name='loginsubmit'>";
        echo "</form>";
    }
}

// Si ya estamos logeados

else
{
    // Salida del Sistema
    if(isset($_REQUEST['exit']))
    {
        session_destroy();

        if(!session_is_registered('login'))
            echo "<a href='usuarios.php'>Home</a>";
    }
}

```

```

}
// Insercion de un nuevo Registro
elseif(isset($_POST['insertsubmit']))
{
$sipuser = $_POST['sipuser'];
$sippass = $_POST['sippass'];
$contexto = $_POST['contexto'];
$sippass = $sipuser.":asterisk:". $sippass;
$sippass = md5($sippass);
$buzon = $sipuser."@default";
$query = mysql_query("INSERT INTO pares_sip (`name`, `host`, `nat`,
`type`, `context`, `md5secret`, `qualify`, `disallow`, `allow`,
`port`, `regseconds`, `lastms`, `username`, `defaultuser`, `mailbox`)
VALUES ('$sipuser', `dynamic`, `no`, `friend`, `$contexto`, `$sippass`,
`yes`, `all`, `alaw;gsm;ulaw`, `0`,`0`,`0`,``,``,``,`buzon`)");

echo "<a href='usuarios.php'>Home</a><br>";
echo "<a href='usuarios.php?exit'>Exit</a>";
}
// Borrado de un Registro
elseif(isset($_POST['deletesubmit']))
{
$sipid = $_POST['sipid'];
$query = mysql_query("DELETE FROM pares_sip WHERE id = '$sipid'");

echo "<a href='usuarios.php'>Home</a><br>";
echo "<a href='usuarios.php?exit'>Exit</a>";
}
// Formularios de Insercion y Borrado
else
{
echo "Insertar Registro:<br>";
echo "<p><form method='post' action='?'>";
echo "Usuario: <input name='sipuser' type='text'><br>";
echo "Contrase&ntilde;a: <input name='sippass' type='password'><br>";
echo "Tipo: <select name='contexto'>";
echo "<option value='extensiones'>Resto</option>";
echo "<option value='manager'>Manager</option>";
echo "<option value='gerencia'>Gerencia</option>";
echo "</select><br>";
echo "<input type='submit' name='insertsubmit' value='Insertar'>";
echo "</form></p>";

echo "<table border='1'>";
echo "<tr><td colspan ='2' align='center'>SIP Peers Activos</td></tr>";
echo "<tr><td>Usuario</td><td>Borrar</td></tr>";

$query = mysql_query("SELECT * FROM pares_sip WHERE type = `friend`");
$rows = mysql_num_rows($query);
for ($i=0;$i<$rows;$i++)
{
$sippeersarray = mysql_fetch_array($query);
$sipuser = $sippeersarray['name'];
$sipid = $sippeersarray['id'];

echo "<tr>";

```

```

echo "<td>".$sipuser."</td>";
echo "<td>";
echo "<form method='post' action='?'>";
echo "<input type=hidden name='sipid' value='$sipid'>";
echo "<input type=submit name='deletesubmit' value='Borrar'>";
echo "</form>";
echo "</td>";
echo "</tr>";

}
echo "</table>";
echo "<a href='usuarios.php?exit'>Exit</a>";

}
}

?>

```

./usuarioasterisk.sql

```

CREATE USER 'asterisk'@'localhost' IDENTIFIED BY 'asterisk';
GRANT USAGE ON * . * TO 'asterisk'@'localhost' IDENTIFIED BY 'asterisk'
WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR
0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;
GRANT ALL PRIVILEGES ON 'asterisk' . * TO 'asterisk'@'localhost';

```

./pares_sip.sql

```

CREATE TABLE `pares_sip` (
  `id` int(11) NOT NULL auto_increment,
  `name` varchar(80) NOT NULL default '',
  `host` varchar(31) NOT NULL default '',
  `nat` varchar(5) NOT NULL default 'no',
  `type` enum('user','peer','friend') NOT NULL default 'friend',
  `callerid` varchar(80) default NULL,
  `context` varchar(80) default NULL,
  `defaulttip` varchar(15) default NULL,
  `dtmfmode` varchar(7) default NULL,
  `fromuser` varchar(80) default NULL,
  `fromdomain` varchar(80) default NULL,
  `insecure` varchar(20) default NULL,
  `mailbox` varchar(50) default NULL,
  `md5secret` varchar(80) default NULL,
  `deny` varchar(95) default NULL,
  `permit` varchar(95) default NULL,
  `qualify` char(3) default NULL,
  `secret` varchar(80) default NULL,
  `disallow` varchar(100) default 'all',
  `allow` varchar(100) default 'alaw;ulaw;gsm',
  `port` smallint(5) unsigned NOT NULL default '0',
  `regserver` varchar(100) default NULL,
  `regseconds` int(11) NOT NULL default '0',
  `lastms` int(11) NOT NULL default '0',
  `username` varchar(80) NOT NULL default '',

```



```

'defaultuser' varchar(80) NOT NULL default '',
'subscribecontext' varchar(80) default NULL,
'useragent' varchar(20) default NULL,
'useradmin' int(1) NOT NULL DEFAULT '0',
PRIMARY KEY ('id'),
UNIQUE KEY 'name' ('name'),
KEY 'name_2' ('name')
) ENGINE=MyISAM ROW_FORMAT=DYNAMIC;

```

./ejemplo_pares_sip.sql

```

INSERT INTO `asterisk`.`pares_sip` (`id`, `name`, `host`, `nat`, `type`,
`callerid`, `context`, `defaultip`, `dtmfmode`, `fromuser`, `fromdomain`,
`insecure`, `mailbox`, `md5secret`, `deny`, `permit`, `qualify`, `secret`,
`disallow`, `allow`, `port`, `regserver`, `regseconds`, `lastms`,
`username`, `defaultuser`, `subscribecontext`, `useragent`, `useradmin`)
VALUES (NULL, 'provedorip', 'home-asterisk.local', 'no', 'friend', NULL,
`entrantes`, NULL, NULL, 'ucaautos', NULL, 'invite', NULL, NULL, NULL,
NULL, 'yes', '12345678', 'all', 'g729;ilbc;gsm;ulaw;alaw', '35060', NULL,
'0', '0', '', 'ucaautos', NULL, NULL, '0');

```

```

INSERT INTO `asterisk`.`pares_sip` (`id`, `name`, `host`, `nat`, `type`,
`callerid`, `context`, `defaultip`, `dtmfmode`, `fromuser`, `fromdomain`,
`insecure`, `mailbox`, `md5secret`, `deny`, `permit`, `qualify`, `secret`,
`disallow`, `allow`, `port`, `regserver`, `regseconds`, `lastms`,
`username`, `defaultuser`, `subscribecontext`, `useragent`, `useradmin`)
VALUES (NULL, '21', 'dynamic', 'no', 'friend', NULL, 'manager', NULL,
`rfc2833`, NULL, NULL, NULL, '21@default',
'5dbe664c6d998967ad9bfe4ea37521a1',
NULL, NULL, 'yes', '', 'all', 'g729;ilbc;gsm;ulaw;alaw', '0', NULL, '0',
'0', '', '', NULL, NULL, '0');

```

```

INSERT INTO `asterisk`.`pares_sip` (`id`, `name`, `host`, `nat`, `type`,
`callerid`, `context`, `defaultip`, `dtmfmode`, `fromuser`, `fromdomain`,
`insecure`, `mailbox`, `md5secret`, `deny`, `permit`, `qualify`, `secret`,
`disallow`, `allow`, `port`, `regserver`, `regseconds`, `lastms`,
`username`, `defaultuser`, `subscribecontext`, `useragent`, `useradmin`)
VALUES (NULL, '54', 'dynamic', 'no', 'friend', NULL, 'resto', NULL,
`rfc2833`, NULL, NULL, NULL, '54@default',
'037f3c092ae05f73054d2d2dd0e7d5dc',
NULL, NULL, 'yes', '', 'all', 'g729;ilbc;gsm;ulaw;alaw', '0', NULL, '0',
'0', '', '', NULL, NULL, '1');

```

Sistema de Conferencias

./tablasmeetme.sql

```

connect meetme;

CREATE TABLE `booking` (
  `bookId` int(10) unsigned NOT NULL auto_increment,
  `clientId` int(10) unsigned default '0',
  `confno` varchar(30) default '0',

```

```

    'pin' varchar(30) NOT NULL default '0',
    'adminpin' varchar(30) NOT NULL default '0',
    'starttime' datetime NOT NULL default '0000-00-00 00:00:00',
    'endtime' datetime NOT NULL default '0000-00-00 00:00:00',
    'dateReq' datetime NOT NULL default '0000-00-00 00:00:00',
    'dateMod' datetime NOT NULL default '0000-00-00 00:00:00',
    'maxUser' varchar(30) NOT NULL default '10',
    'status' varchar(30) NOT NULL default 'A',
    'confOwner' varchar(30) NOT NULL default '',
    'confDesc' varchar(100) NOT NULL default '',
    'adminopts' varchar(10) NOT NULL default '',
    'opts' varchar(10) NOT NULL default '',
    'sequenceNo' int(10) unsigned default '0',
    'recurInterval' int(10) unsigned default '0',
    'recordingfilename' varchar(128) default NULL,
    PRIMARY KEY ('bookId')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=145 ;

CREATE TABLE `cdr` (
  'bookId' int(11) default NULL,
  'duration' varchar(12) default NULL,
  'CIDnum' varchar(32) default NULL,
  'CIDname' varchar(32) default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE `notifications` (
  'id' int(11) NOT NULL auto_increment,
  'user_id' int(11) NOT NULL default '0',
  'book_id' int(11) NOT NULL default '0',
  'ntype' char(10) default NULL,
  'ndate' timestamp NOT NULL default CURRENT_TIMESTAMP,
  PRIMARY KEY ('id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

CREATE TABLE `participants` (
  'id' int(11) NOT NULL auto_increment,
  'user_id' int(11) NOT NULL default '0',
  'book_id' int(10) NOT NULL default '0',
  PRIMARY KEY ('id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=782 ;

CREATE TABLE `user` (
  'id' int(11) NOT NULL auto_increment,
  'email' varchar(100) NOT NULL default '',
  'password' varchar(25) default NULL,
  'first_name' varchar(50) default NULL,
  'last_name' varchar(50) default NULL,
  'telephone' varchar(15) default NULL,
  'admin' varchar(5) NOT NULL default 'User',
  PRIMARY KEY ('id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=20 ;

```

Generación Automática de Llamadas

`./apache_monitor.sh`

```
#!/bin/bash

SERVER="home-asterisk.local"
PORT="80"
WORKDIR="/home/asterisk/repositorio/scripts"

echo "quit" | telnet $SERVER $PORT | grep "Escape character is"

if [ "$?" -ne 0 ]; then
    cp $WORKDIR/apache_warning.call $WORKDIR/apache_warning_temp.call
    mv $WORKDIR/apache_warning_temp.call /var/spool/asterisk/outgoing/
    exit 1
fi
```

`./apache_warning.call`

```
Channel:SIP/54
Application:Playback
Data:alerta-web
CallerID:Emergencias CRM <111>
MaxRetries:3
WaitTime:90
RetryTime:10
```

Detalle de Llamadas y Eventos

`./sistemacdr.sql`

```
CREATE TABLE cdr (
    calldate datetime NOT NULL default '0000-00-00 00:00:00',
    clid varchar(80) NOT NULL default '',
    src varchar(80) NOT NULL default '',
    dst varchar(80) NOT NULL default '',
    dcontext varchar(80) NOT NULL default '',
    channel varchar(80) NOT NULL default '',
    dstchannel varchar(80) NOT NULL default '',
    lastapp varchar(80) NOT NULL default '',
    lastdata varchar(80) NOT NULL default '',
    duration int(11) NOT NULL default '0',
    billsec int(11) NOT NULL default '0',
    disposition varchar(45) NOT NULL default '',
    amaflags int(11) NOT NULL default '0',
    accountcode varchar(20) NOT NULL default '',
    uniqueid varchar(32) NOT NULL default '',
    userfield varchar(255) NOT NULL default ''
);

ALTER TABLE `cdr` ADD INDEX ( `calldate` );
ALTER TABLE `cdr` ADD INDEX ( `dst` );
ALTER TABLE `cdr` ADD INDEX ( `accountcode` );
```

Interactive Voice Response

/var/lib/asterisk/agi-bin/asterisk-asr.php

```
#!/usr/bin/php -q
<?php
require_once("/var/lib/asterisk/agi-bin/phpagi/phpagi.php");
$agi = new AGI();
// $agi->answer();
// Primero convertimos el fichero que generamos desde Asterisk
// a formato FLAC (que es el que admite Google)
shell_exec("sox /var/lib/asterisk/sounds/grabacion-asr.wav
/tmp/grabacion-asr.flac");

// Despues lanzamos una peticion a Google para que nos convierta
// el fichero FLAC en texto, nos devuelve una cadena de texto formato JSON
$linea = shell_exec("wget --post-file /tmp/grabacion-asr.flac
--header='Content-Type: audio/x-flac; rate=8000' -O
- 'http://www.google.com/speech-api/v1/recognize?lang=es_ES'");
// Convertimos esa cadena JSON gracias a PHP en un objeto directamente
// $obj = json_decode(utf8_encode($linea));
$obj = json_decode($linea);
// Sacamos el valor utterance
$valorasr = $obj->{"hypotheses"}[0]->{"utterance"};

$agi->verbose($valorasr);
// Pasamos el texto a Asterisk en forma de variable de canal
$agi->set_variable("TEXTTOASR", $valorasr);
$agi->verbose($retString);
?>
```

/var/lib/asterisk/agi-bin/asterisk-tts.php

```
#!/usr/bin/php -q
<?php

require_once("/var/lib/asterisk/agi-bin/phpagi/phpagi.php");
$agi = new AGI();
$traduccion = $argv[1];

$agi->verbose($traduccion);

// Aqui descargamos el fichero Mp3 haciendo pensar a Google
// que somos un Navegador
shell_exec("wget --header='User-Agent:Mozilla/4.0'
'http://translate.google.com/translate_tts?tl=es&ie=UTF-8
&q=".$traduccion."' -O /tmp/google-tts.mp3");

// Y convertimos el fichero con .lame. a wav que es reconocible
// por Asterisk
shell_exec("sox /tmp/google-tts.mp3 -r 8000 -c 1 -t raw -s
/var/lib/asterisk/sounds/google-tts.sln");

$retString = "Fin de la conversion";
$agi->verbose($retString);
```

?>

/var/lib/asterisk/agi-bin/fingarantia.php

```
#!/usr/bin/php -q
<?php

require_once("/var/lib/asterisk/agi-bin/phpagi/phpagi.php");

function fecha_normal($fecha)
{
    ereg( "([0-9]{2,4})-([0-9]{1,2})-([0-9]{1,2})", $fecha, $mifecha);
    $lafecha=$mifecha[3].$mifecha[2].$mifecha[1];
    return $lafecha;
}

function resta_fecha($fecha_inicial, $fecha_final)
{
    ereg( "([0-9]{1,2}) ([0-9]{1,2}) ([0-9]{2,4})", $fecha_inicial, $fini);
    ereg( "([0-9]{1,2}) ([0-9]{1,2}) ([0-9]{2,4})", $fecha_final, $ffin);

    $fecha1 = mktime(0,0,0,$fini[2], $fini[1], $fini[3]);
    $fecha2 = mktime(0,0,0,$ffin[2], $ffin[1], $ffin[3]);

    return round(($fecha2 - $fecha1) / (60 * 60 * 24));
}

// http://es.wikibooks.org/wiki/Algoritmo_para_obtener_la_letra_del_NIF
function letra_nif($dni)
{
    return substr("TRWAGMYFPDXBNJZSQVHLCKE",$dni%23,1);
}

$agi = new AGI();
$dni = $argv[1];
$dni = str_replace(" ", "", $dni);
$dni = $dni.letra_nif($dni);

// Conexion a la base de Datos
$base = "erp";
$host = "home-asterisk.local";
$user = "asterisk";
$password = "asterisk";
$conexion = mysql_connect($host,$user,$password);
$result = mysql_select_db($base,$conexion)
or die ("Error en la Conexion a BD");

$query = mysql_query("SELECT * FROM clientes WHERE dni = '$dni'");

if(!mysql_num_rows($query))
{
    $diasgarantia = '0';
}
else
{
```

```

$cliente = mysql_fetch_array($query);
$idcliente = $cliente['id'];

mysql_query("SELECT * FROM vehiculos WHERE cliente = '$idcliente'");
$vehiculo = mysql_fetch_array($query2);
$fcompra = $vehiculo['fcompra'];
$garantia = $vehiculo['garantia'];
$ahora = date("dmY",strtotime("-$garantia year"));
$fechacompra = fecha_normal($fcompra);
$diasgarantia = resta_fecha($ahora,$fechacompra);
}

$agi->verbose($diasgarantia);
$agi->set_variable("textotts",$diasgarantia);
?>

```

Bibliografía

- [1] Areski Belaid. CDR-Stats. <http://www.cdr-stats.org/>.
- [2] WIKI Books. Algoritmo para obtener letra DNI. http://es.wikibooks.org/wiki/Algoritmo_para_obtener_la_letra_del_NIF.
- [3] Manuel Camargo. Asterisk en 10000 Horas. <http://10000horas.com/asterisk>.
- [4] Counterpath Corp. Softphone X-Lite. <http://www.counterpath.com/x-lite.html>.
- [5] Varios Desarrolladores. Diagramas de flujo con dia w32. <http://projects.gnome.org/dia/>.
- [6] DNS Exit. Configuración Relay SMTP Smarthost en Postfix con SASL. <http://www.dnsexit.com/support/mailrelay/postfix.html>.
- [7] Harvard Business Publishing for Educators: Higher Education. Harvard Business Publishing. <http://hbsp.harvard.edu/>.
- [8] WIKIMedia Foundation. MediaWiki software para crear WIKIs. <http://www.mediawiki.org/wiki/MediaWiki/es>.
- [9] Inc. Geeknet. Página oficial sourceforge. <http://sourceforge.net/>.
- [10] W. Jason Gilmore. *Beginning PHP and MySQL: From Novice to Professional*. Apress, 2010.
- [11] Flavio E. Goncalves. *Building Telephony Systems with OpenSIPS 1.6*. Packt Publishing, 2010.
- [12] Ted Gould. Gráficos vectoriales con inkscape. <http://inkscape.org>.
- [13] The PHP Group. Lenguaje de programación php. <http://www.php.net/>.
- [14] Nicolás Gudiño. Asternic Stats. <http://www.asternic.biz/>.
- [15] Digium Inc. Asterisk now! distribución asterisk. <http://www.asterisk.org/asterisknow>.
- [16] Digium Inc. Pagina oficial de Asterisk. <http://www.asterisk.org>.
- [17] Digium Inc. Pagina oficial de Digium. <http://www.digium.com>.
- [18] GitHub Inc. Página oficial github. <https://github.com/>.
- [19] Alan B. Johnston. *SIP: Understanding the Session Initiation Protocol (Artech House Telecommunications)*. Artech House, 2009.

- [20] Jim Van Meggelen y Russell Bryant Leif Madsen. *Asterisk The Definitive Guide*. O'Reilly Media, 2011.
- [21] Zoiper LLC. Softphone Zoiper. <http://www.zoiper.com/>.
- [22] Manuel Camargo Lominchar. Wikiasterisk. <http://www.wikiasterisk.com>.
- [23] Leif Madsen and Russell Bryant. *Asterisk Cookbook (Oreilly Cookbooks)*. O'Reilly Media, 2011.
- [24] Wikimedia Foundation Múltiples Editores. Latex en Wikibooks. <http://en.wikibooks.org/wiki/LaTeX>.
- [25] Colin Perkins. *RTP: Audio and Video for the Internet*. Addison-Wesley Professional, 2003.
- [26] Planner. Web oficial de planner. <http://live.gnome.org/Planner>.
- [27] Kamailio SIP Server Project. Servidor SIP Proxy Kamailio. <http://www.kamailio.org>.
- [28] Telefónica S.A.U. Características de las interfaces analógicas de telefonía de España. <http://info.telefonica.es/ext/interfaces/pdf/ITE-CA-001-V05.pdf>.
- [29] Christian Schenk. Pagina oficial de MikTeX. <http://miktex.org>.
- [30] PaloSanto Solutions. Elastix, distribución basada en asterisk. <http://www.elastix.org/>.
- [31] Daniel Stenberg. Librería curl. <http://curl.haxx.se/>.
- [32] GIT Team. Git fast version control. <http://git-scm.com/>.
- [33] MinGW Team. Minimalist gnu for windows. <http://www.mingw.org/>.
- [34] OpenWRT Team. Open wrt y asterisk. <http://wiki.openwrt.org/doc/howto/voip.asterisk>.
- [35] The GIMP Team. Diseño gráfico con gimp. <http://www.gimp.org/>.
- [36] VOIP-Info LCC. Wiki sobre VoIP en Inglés. <http://www.voip-info.org>.
- [37] vTiger Systems PLD. PBXManager Wiki de Vtiger. https://wiki.vtiger.com/index.php/PBX_Manager_Module.
- [38] vTiger Systems PLD. Sistema vTiger CRM. <http://www.vtiger.com>.
- [39] Gary Walker. *IT Problem Management*. Prentice Hall, 2001.
- [40] Theodore Wallingford. *Switching to VoIP*. O'Reilly Media, 2005.
- [41] Wikipedia. Carrier grade. http://en.wikipedia.org/wiki/Carrier_grade.
- [42] Wikipedia. Computer telephony integration. http://es.wikipedia.org/wiki/Computer_Telephony_Integration.
- [43] Wikipedia. Comunicaciones unificadas. http://es.wikipedia.org/wiki/Comunicaciones_unificadas.
- [44] Wikipedia. Concepto de wiki. <http://es.wikipedia.org/wiki/Wiki>.
- [45] Wikipedia. CTI en Wikipedia. http://es.wikipedia.org/wiki/Computer_Telephony_Integration.

- [46] Wikipedia. Do it yourself. http://es.wikipedia.org/wiki/H%C3%A1lgalo_usted_mismo.
- [47] Wikipedia. Interactive voice response. http://es.wikipedia.org/wiki/Interactive_Voice_Response.
- [48] Wikipedia. Private branch exchange. <http://es.wikipedia.org/wiki/PBX>.
- [49] Wikipedia. Voz sobre ip. http://es.wikipedia.org/wiki/Voz_sobre_Protocolo_de_Internet.
- [50] Stefan Wintermeyer y Stephen Bosch. *Practical Asterisk, From Beginner to Expert*. Addison-Wesley Professional, 2009.
- [51] Omry Yadan. Instalación de FireStats. <http://firestats.cc/wiki/MediaWiki>.
- [52] Omry Yadan. Sistema de Estadísticas Firestats. <http://firestats.cc>.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall

subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to

the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
 - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
 - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
 - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
 - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
 - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
 - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the

Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.