



UNIVERSIDAD DEL AZUAY
FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
ESCUELA DE INGENIERIA DE SISTEMAS

**IMPLEMENTACIÓN DE UNA CENTRAL TELEFONICA PBX BASADA
EN ASTERISK**

**TRABAJO DE GRADUACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO DE SISTEMAS**

AUTOR:

CÉSAR ESTALIN AGUILAR SAMANIEGO.

DIRECTOR:

ING. LUIS CALDERÓN

CUENCA - ECUADOR

2011

Dedicatoria

Dedico el trabajo de graduación a toda mi familia, que me ha apoyado en el transcurso de toda mi vida estudiantil, con un especial agradecimiento a mis padres por el apoyo tanto afectivo como económicamente, a mis hermanos, sobrino y demás familiares, especialmente a mi abuelita Mercedes Mora.

Agradecimientos

Agradezco primeramente a mis padres por el apoyo brindado, al Ing. Luis Calderón mi director de tesis por haberme brindado la ayuda necesaria para poder concluir con el trabajo, igualmente a todos los profesores que me han brindado los conocimientos necesarios en el transcurso de la vida universitaria.

GRACIAS

Índice de Contenido

Dedicatoria.....	ii
Agradecimientos.....	iii
Índice de Contenido	iv
Índice de Ilustraciones	vi
Índice de Tablas	viii
Resumen.....	ix
Abstract.....	x
Introducción	1
CAPITULO 1: INTRODUCCIÓN	2
1.1 Breve Historia de la telefonía	2
1.2 El desarrollo de la tecnología telefónica	2
1.3 Telefónica VOIP	3
1.3.1 Elementos de la VOIP	3
1.3.2 Arquitectura de VOIP	4
1.3.3 Parámetros de la VOIP	14
CAPITULO 2: ANTECEDENTES.....	23
2.1 Asterisk.....	23
2.2 Software	23
2.2.1 Elastix	23
2.2.2 Softphone.....	24
2.3 Hardware: Requerimientos del Sistema	24
2.4 Sintetización de Voz.....	25
2.4.1 Funcionamiento	25
2.4.1.1 Análisis de texto	25
2.4.1.2 Análisis lingüístico	26
2.4.1.3 Generación de audio	26
CAPITULO 3: DEFINICIÓN DEL PROYECTO	27
3.1 Descripción detallada del proyecto	27
3.2 Definición del plan de Mercado	27
3.3 Diagrama de Conexiones	28
CAPITULO 4: CONFIGURACION DEL PROYECTO	29
4.1 Configuración de Archivos Asterisk	29

4.1.1 Configuración de Extensiones	29
4.1.2 Configuración de Sintetización de Voz (FESTIVAL)	34
4.1.3 Configuración de IVR	36
4.1.4 Configuración de Troncal	42
4.1.4.1 Configuraciones generales de troncales.	42
4.2 Configuración de Elastix	46
4.2.1 Configuración de Mensajería Instantánea (OPENFIRE)	46
4.2.2 Configuración de FAX (HYLAFAX)	55
4.2.3 Configuración de Call Center	62
CAPITULO 5: PLAN DE MERCADO AVANZADO	70
5.1 Concepto.....	70
5.2 Commandos AGI.....	70
5.3 Configuración.....	71
5.4 Ejemplo.....	73
CAPITULO 6: PRUEBAS.....	76
6.1 Prueba de llamada entre extensiones.....	76
6.2 Prueba de llamadas entrantes y salientes.....	77
6.4 Prueba de Hylafax	80
6.5 Prueba de Sintetización de Voz.....	81
6.6 Prueba de CallCenter.....	81

Índice de Ilustraciones

Figura 1-1 Teléfono de Reis.....	2
Figura 1-2 Esquema de una Arquitectura.....	5
Figura 1-3 Protocolos.....	5
Figura 1-4 Ejemplo de una zona H.323	7
Figura 1-5 Ejemplo de H.428.....	8
Figura 1-6 Ejemplo de Protocolo SIP	10
Figura 1-7 Ejemplo de Protocolo IAX2	12
Figura 2-1 Sofphone ZOIPER.....	24
Figura 2-2 Conversion de Texto a Voz	26
Figura 3-1 Diagrama de Conexiones.....	28
Figura 4-1 Nombre de la Cuenta SIP	33
Figura 4-2 Configuración de Cuenta SIP	34
Figura 4-3 Habilitar Open Fire.....	47
Figura 4-4 Configuración de Idioma.....	47
Figura 4-5 Configuración de Servidor	48
Figura 4-6 Configuración de la Fuente de Datos	49
Figura 4-7 Seteo de Perfil	49
Figura 4-8 Configuración de la Cuenta del Administrador.....	50
Figura 4-9 Configuración Finalizada	50
Figura 4-10 Configuración de Usuarios	51
Figura 4-11 Icono de Software Spark	51
Figura 4-12 Ejecutar el Programa	52
Figura 4-13 Configuración de Instalación de Spark 2.5.8	52
Figura 4-14 Configuración del Directorio para la instalación	53
Figura 4-15 Configuración del Nombre en el Menu Principal	53
Figura 4-16 Configuración de Creación de Iconos	54
Figura 4-17 Finalización de la Instalacion de Spark 2.5.8.....	54
Figura 4-18 Spark 2.5.8.....	55
Figura 4-19 Configuración en Elastix de Fax	56
Figura 4-20 Icono Agregar Impresora.....	56
Figura 4-21 Asistente para Agregar Impresora.....	57
Figura 4-22 Descripción de la Impresora a Utilizar.....	57
Figura 4-23 Configuración del Puerto de Impresora.....	58

Figura 4-24 Nombre del Puerto.....	58
Figura 4-25 Configuración del Modelo de la Impresora.....	59
Figura 4-26 Nombre de la Impresora	59
Figura 4-27 Configuración de Puerto.....	60
Figura 4-28 Configuración de Cliente Hylafax.....	61
Figura 4-29 Enviar Fax	62
Figura 4-30 Configuración de Cola.....	64
Figura 4-31 Configuración de Opciones de Cola.....	64
Figura 4-32 Configuración de Agente.....	65
Figura 4-33 Crear Formulario	66
Figura 4-34 Visualización de Formulario	66
Figura 4-35 Visualización de Call Center	67
Figura 4-36 Creación de Campaña.....	69
Figura 4-37 Listado de Campaña	69
Figura 6-1 Muestra las extensiones SIP Conectadas.....	76
Figura 6-2 Extension 3010 realizando llamada a extensión 1010.....	76
Figura 6-3 Extensión 1010 recibiendo llamada de extensión 3010	77
Figura 6-4 Consola Asterisk muestra ruteo de llamada	77
Figura 6-5 Prueba de llamada entrante.....	78
Figura 6-6 Llamada Saliente desde zoiper	78
Figura 6-7 Consola Asterisk Prueba llamada Saliente	79
Figura 6-8 Computadora #1 conectada a SPARK	80
Figura 6-9 Computadora #2 conectada a SPARK	80
Figura 6-10 Enviar Fax	81
Figura 6-11 Conectarse al Agente.....	81
Figura 6-12 Conectando Elastix con Zoiper	82
Figura 6-13 Consola del Agente	82

Índice de Tablas

Tabla 1-1 URI de IAX.....	11
Tabla 1-2 Comparativas de Protocolos VOIP más populares.....	13
Tabla 1-3 Principales Codécs.....	15
Tabla 4-1 Configuración de extensión SIP	33
Tabla 4-2 Parámetros de Configuración de una troncal.....	44
Tabla 4-3 Configuración de Fax Virtual	54
Tabla 4-4 Configuración de Cola.....	63
Tabla 4-5 Configuración del Agente.....	65
Tabla 4-6 Configuración del Formulario	65
Tabla 4-7 Configuración de Call Center	66

Resumen

El trabajo se trata de realizar una guía, de cómo configurar desde Asterisk lo que son las extensiones, la troncal y la respuesta interactiva de voz (IVR) por sus siglas en ingles, para lo cual utilizaremos la sintetización de voz con el módulo Festival, además de las configuraciones los sistemas de Fax (HYLAFAX), Mensajería Instantánea (OPENFIRE), configuración del Call Center para poder realizar de una manera más eficiente lo que son las campañas de entrada y de salida desde Elastix.

Para la realización de las guías y los ejemplos de OpenFire, HylaFax, CallCenter que hace referencia a los Plug in que tiene Elastix procedemos a:

- Realizar una Guía y ejemplo de cómo realizar una configuración de OpenFire (Mensajería Instantánea) en Elastix
- Realizar una Guía y ejemplo de cómo realizar una configuración de HylaFax (Fax) en Elastix
- Realizar una Guía y ejemplo de cómo realizar una configuración de Sintetización de voz en Elastix
- Realizar una guía y ejemplo de Plan de marcado avanzado en Asterisk.
- Realizar una Guía y ejemplo de cómo realizar la configuración de CallCenter en Elastix

ABSTRACT

This work is a guide on how to configure Asterisk software for the implementation of the branches, the central line and the Interactive Voice Response (IVR). For this purpose, the Festival Speech Synthesis System will be employed, as well as the Fax System (HYLAFAX), Instant Messaging (OPENFIRE), and a Call Center in order to manage inbound and outbound calls more efficiently through Elastix software.

The guide and the examples of OpenFire, HylaFax, Call center which are part of the Plug-in system of Elastix software will contain:

- A guide and an example of how to configure OpenFire (Instant Messaging) in Elastix.
- A guide and example of how to configure HylaFax (Fax) in Elastix.
- A guide and example of how to configure Speech Synthesis in Elastix.
- A guide and example of the advanced dialing plan in Asterisk.
- A guide and example of how to configure a Call center in Elastix.



Diana Lee Rodas
Translated by,
Diana lee Rodas

Introducción

Las telecomunicaciones han avanzado en los últimos años, con lo cual sus costos se han incrementado, debido a la fuerte inversión en infraestructura y a los dispositivos necesarios para permitir el acceso a la comunicación en las distintas áreas. Por lo cual las empresas se han visto en la necesidad de buscar nuevos métodos y tecnologías para reducir costos y mantener la eficiencia y eficacia.

Internet ha sido una pieza clave para que podamos alcanzar este objetivo, ya que gracias al constante desarrollo de las redes IP, combinado con técnicas avanzadas de digitalización de voz y los protocolos de control y transmisión en tiempo real, han hecho que la telefonía sobre IP se convierta en un tema estratégico, ya que permite la calidad de servicio a bajo costo, y con el apareamiento del software Asterisk se ha masificado la tecnología Voz sobre IP y uno de los mayores atractivos es que soporta múltiples protocolos.

La comunicación de una centralita (PBX) basadas en Asterisk es una buena opción para poder implementar en todo tipo de negocios desde los más pequeños a los más grandes, para que todas las necesidades se cumplan a cabalidad se debe de realizar una correcta instalación, configuración y administración de estas centralitas; además surgen otras necesidades las cuales se pueden satisfacer con Elastix como la mensajería instantánea, fax, Call Center las cuales ayudan en la productividad del negocio.

Dado la Carencia de un manual para la realización de una buena implementación de una centralilla PBX basada en Asterisk y configuración de los sistemas de OpenFire (Mensajería Instantánea), HylaFax (Fax), Call Center basados en Elastix he desarrollado la siguiente guía para la realización de las configuraciones tanto en Elastix como en Asterisk.

CAPITULO 1: INTRODUCCIÓN

1.1 Breve Historia de la telefonía

La idea de la existencia de un aparato parlante, rondaba por la cabeza de muchos inventores desde tiempos inmemoriales, pero fue a principios del siglo XIX cuando la idea se materializó, pues ya se había descubierto la electricidad, inventado el telégrafo y algunos ya experimentaban con ondas de radio.

En 1849 Antonio Meucci, médico italiano considerado por muchos como el inventor del teléfono, hizo una demostración de un dispositivo capaz de transmitir voz en La Habana. Pocos años después, en 1854, el mismo Meucci hace una nueva demostración de su invención en la ciudad de Nueva York. Otros también perseguían la idea de construir un “telégrafo parlante” y es así como en 1860 el alemán Johann Philip Reis construye un dispositivo capaz de transmitir voz basada en la idea original de Charles Bourseul, quien a su vez describió la construcción de dicho dispositivo en 1854 pero nunca lo construyó. (Landivar, pág. 64)

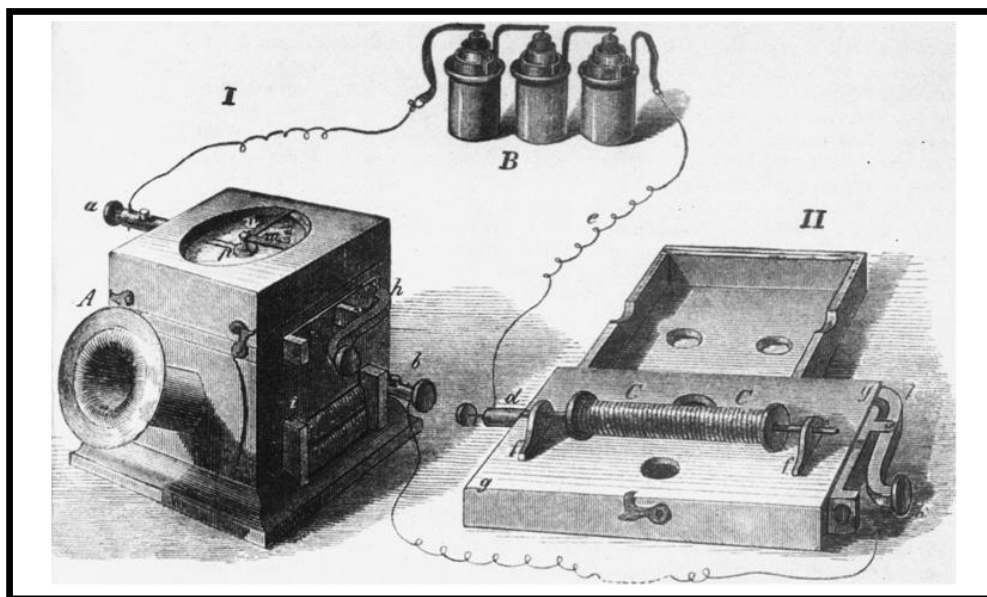


Figura 1-1 Teléfono de Reis

1.2 El desarrollo de la tecnología telefónica

Como acontece siempre con los adelantos tecnológicos, la telefonía continuó desarrollándose. Al principio, para que un abonado se comunicara con otro, este tenía que solicitarle la llamada a una operadora, quien manualmente conectaba los cables

para conmutar un punto con otro. En 1891 se inventó un teléfono “automático” que permitía marcar directamente.

En un principio Bell fue casi exclusivamente la única compañía en explotar la tecnología debido a sus patentes. Sin embargo, cuando estas expiraron nacieron cientos de pequeñas compañías que empezaron a dar servicio, la mayoría en sitios rurales donde Bell aún no llegaba. Poco a poco estas compañías empezaron a crecer y ya a inicios del siglo 20 tenían en su conjunto más abonados que la propia Bell. La sana competencia hizo lo suyo y la tecnología telefónica aceleró su evolución.

En 1947 científicos de Bell inventan el transistor y cambian el curso de la historia de la humanidad. En 1948 ganan el Premio Nobel por su trabajo. En los años 60s se lanzan los primeros satélites de comunicaciones y las comunicaciones entre continentes se facilitan. No está demás decir que esto no hubiera sido posible sin la previa invención del transistor.

1.3 Telefonía VOIP

La voz sobre IP o VOIP consiste en transmitir voz sobre el protocolo de internet (IP- internet Protocol) en forma de paquetes de datos, la Telefonía IP es una aplicación inmediata de esta tecnología, de forma que permita la realización de llamadas telefónicas ordinarias sobre redes IP u otras redes de paquetes utilizando un PC, gateways y teléfonos estándares. En general, servicios de comunicación voz, fax, aplicaciones de mensajes de voz.

1.3.1 Elementos de la VOIP

Los elementos necesarios para la implementación correcta de la telefonía VOIP son los siguientes elementos:

El cliente

El cliente que tiene un número de teléfono asignado a su dirección IP y puede establecer y termina las llamadas de voz, esta información se codifica, se empaqueta y se transmite a través del micrófono (entrada de información) del usuario, de la misma forma la información se decodifica y reproduce a través de los altavoces o audífonos (salida de la información).

Los servidores

Los servidores llevan a cabo funciones relacionadas a la telefonía como crear y gestionar las cuentas, las respuestas de las llamadas automáticas, realizar las funciones de un sistema interactivo de respuesta por voz, almacenar y servir de correo de voz, encaminar los llamados entre la red de la telefonía e internet, de manejar operaciones de base de datos, realizándolas en un tiempo real. Entre estas operaciones se tienen la contabilidad, la recolección, el enrutamiento, la administración, control del servicio y permitir que los usuarios se “registren” almacenando la dirección ip donde deben acceder para realizar la comunicación con este usuario.

Los gateways

Los gateways, es un dispositivo de red que brindan un puente de comunicación entre todos los usuarios, su función principal es convertir las llamadas de voz en tiempo real, entre una red IP y la red telefónica publica conmutada (PSTN) o su centralita digital, además permiten que las llamadas salientes generadas por la centralita digital se conviertan a IP y salgan por la conexión a Internet de banda ancha.

1.3.2 Arquitectura de VOIP

Se definen tres elementos fundamentales para una correcta estructura:

Terminales: es el principal dispositivo utilizado y específicamente diseñado para su uso VOIP y que permite realizar una comunicación utilizando una red IP ya sea mediante red de área local (LAN) o internet, son los sustitutos de los actuales teléfonos, puede ser tradicionalmente un PC equipado con periféricos de audio (parlantes y micrófono) y software conocido como softphone o con hardware denominado teléfono IP.

Gateways: se trata del enlace con la red telefónica tradicional, actuando de forma transparente para el usuario, entre las principales funcionalidades tenemos:

- Integración de una centralita tradicional a una red VOIP, como estrategia de migración a una solución multisede.

- Posibilidad de conexión de una centralita tradicional (analógica o digital) a VOIP para acceso a operadores de bajo coste.
- Conexión de centralitas tradicionales a servicios avanzados: integración con la red Skype, acceso a proveedores de DDI (numeración telefónica) con cobertura mundial a muy bajo costo.
- Acceso a servicios avanzados de CallCenter o contestación telefónica: IVRS, CTI, ACDS, grabación de llamadas, etc.
- Reduce la necesidad de líneas PSTN, lo cual redundará en ahorros de costes.

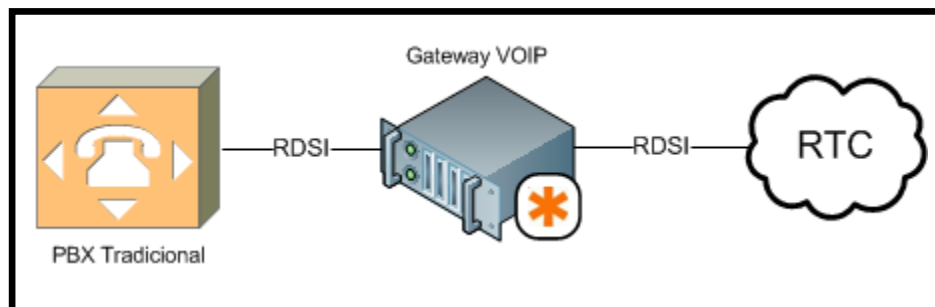


Figura 1-2 Esquema de una Arquitectura

Protocolos de VOIP: son los lenguajes que utilizarán los distintos dispositivos VOIP para su conexión, el cual abarca una serie de transacciones de señalización entre terminales que cargan dos flujos de audio, para cada dirección de la conversación. Esta parte es importante ya que de ella dependerá la eficacia y la complejidad de la comunicación.

Algunos de los protocolos VOIP más importantes y compatibles con Asterisk PBX son: H.323, IAX, SIP, MEGACO (H.248).

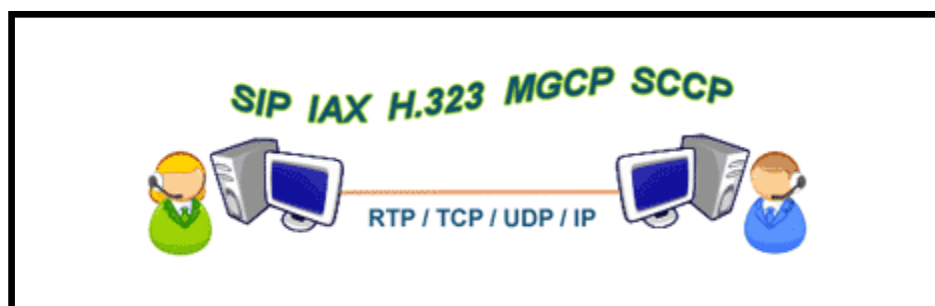


Figura 1-3 Protocolos

H.323: es un estándar de la ITU (International Telecommunications Union) diseñado para permitir transmisiones multimedia en LANS basadas en IP, que describe una familia de protocolos usados para realizar el control de llamadas en una comunicación multimedia a través de redes conmutadas por paquetes. Los protocolos más importantes dentro del H.323 se usan para la configuración, administración y terminación de llamadas (H.225 y H.245); H.225 realiza el control de la llamada y H.245 la administración de la misma.

Sus principales características son:

- No garantiza una calidad de servicio (QoS)
- Es independiente de la topología de la red
- Admite pasarelas
- Permite usar más de un canal (voz, vídeo, datos) al mismo tiempo.
- El estándar permite que las empresas añadan funcionalidades, siempre que implementen las funciones de interoperabilidad necesarias.

Los componentes principales del sistema H.323 son:

- Terminales: Equipamiento que utilizan directamente los usuarios. Se pueden implementar tanto por software (mediante un ordenador) como por hardware (dispositivo físico).
- Guardianes (Gatekeepers): Son el centro de toda organización VOIP y son el equivalente a las centralitas privadas o PBX (Private Branch eXchange). Normalmente se implementan por software.
- Pasarelas (Gateways): Hacen de enlace con la red telefónica conmutada, actuando de forma transparente para el usuario.
- Unidades de Control Multipunto (MCUs): se encargan de gestionar las multi-conferencias.

Los principales protocolos utilizados son:

- RAS (Registro, Admisión, Situación): Se utiliza sólo en zonas que tengan un guardián para la gestión de la zona de control del mismo.
- H.225: Mensajes de establecimiento y finalización de llamada entre terminales o con el guardián.

- H.245: Mensajes de control extremo a extremo. Negociación de las capacidades de ancho de banda (mensajes Termina Capability Set), de la apertura y cierre de los canales lógicos (mensajes Open Logical Channel, Close Logical Channel y End Session Command), de los códecs y mensajes de control de flujo.
- RTP/RTCP (Real-Time Transport Protocol / Real-Time Transport Control Protocol): Transporte punto a punto de datos en tiempo real.

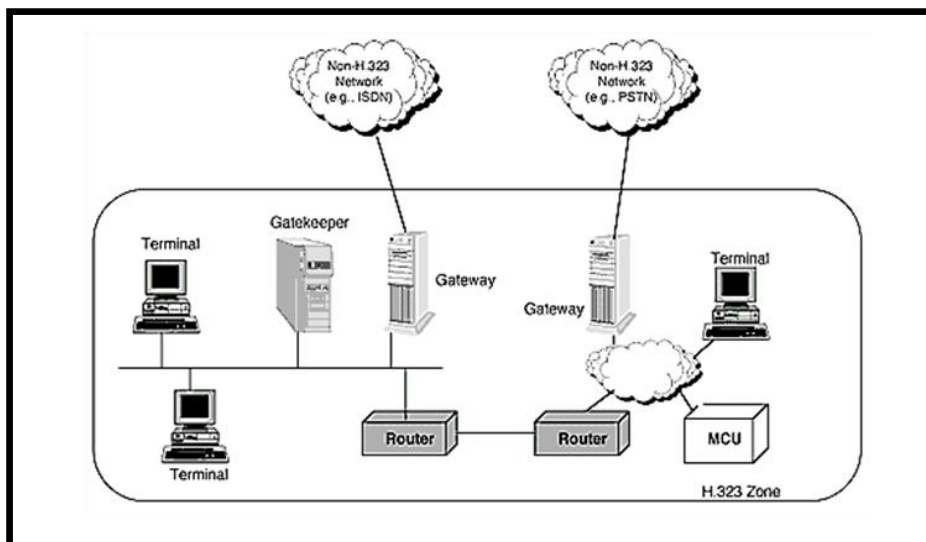


Figura 1-4 Ejemplo de una zona H.323

Megaco (H.428): El protocolo Megaco también conocido como H.428, es un protocolo de control de pasarela con muchas aplicaciones, que tiene su origen en el protocolo MGCP (Media Gateway Control Protocol, Protocolo de control de puerta de enlace al medio), proporciona un control centralizado de las comunicaciones y servicios multimedia a través de redes basadas en IP, este protocolo es el resultado de la cooperación entre la UIT (Unión internacional de telecomunicaciones) y la IETF (Internet Engineering Task Force), permite la conmutación de llamadas de voz, fax y multimedia entre una gran variedad de aplicaciones de pasarela trasladando trenes de información de redes IP RTPC, ATM, y otros sistemas.

Es en esencia un protocolo de control; con topología Maestro/Esclavo, es decir, no pueden funcionar sin un dispositivo de control (softswitch), son protocolos que fueron pensados únicamente para gateways y no para terminales, y el concepto es simplificar lo más posible la inteligencia de estos gateways llevándola al Call Agent

(softswitch). Es importante aclarar que en ambos casos se usa SDP como protocolo de control de señalización de llamada, y RTP para el transporte del audio.

Componentes que intervienen:

- Media Gateways (Pasarela de medios) (MG): Son los elementos funcionales que median entre los puntos finales, es decir, los clientes
- Media Gateway Controller (Controlador de la pasarela de medios) (MGC): Controlaran a los Media Gateways para una buena gestión en el intercambio de información a través del protocolo MGCP. El MGC también se suele llamar Call Agent.

Una de las características fundamentales de este modelo, es que los Media Gateway son capaces (en teoría) de mantener comunicaciones tanto con el H.323 como con el SIP, algo fundamental para la óptima implantación del sistema VOIP.

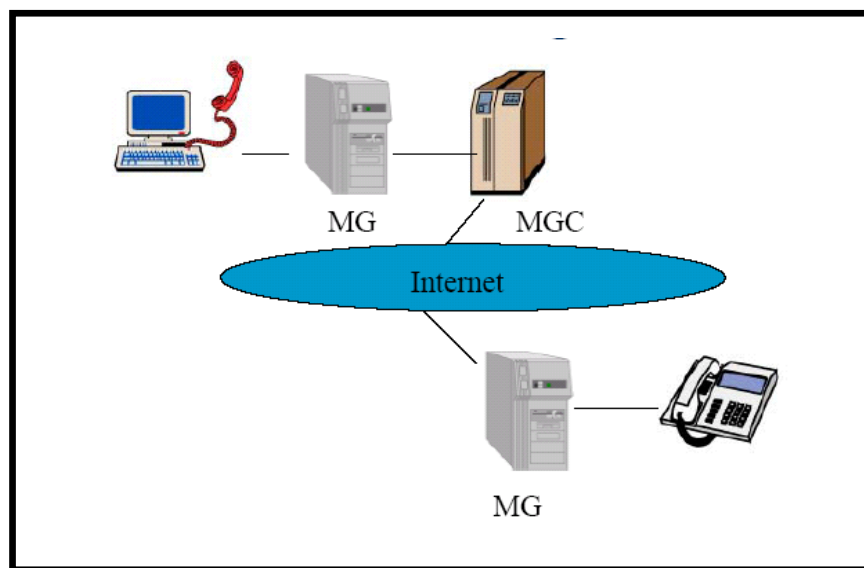


Figura 1-5 Ejemplo de H.428

SIP: El protocolo SIP (Session Initiation Protocol) fue desarrollado por el grupo MMUSIC (Multimedia Session Control) del IETF, definiendo una arquitectura de señalización extremo a extremo que implica que toda la lógica es almacenada en los dispositivos finales (salvo el ruteado de los mensajes SIP) y control para VOIP. Inicialmente fue publicado en febrero del 1996 en la RFC 2543, ahora obsoleta con la publicación de la nueva versión RFC 3261 que se publicó en junio del 2002.

El propósito de SIP es la comunicación entre dispositivos multimedia, implementa funciones típicas de telefonía (llamar a un número, provocar que un teléfono suene al ser llamado, escuchar la señal de tono o de ocupado), permite el establecimiento de sesiones multipunto, permite que un usuario esté registrado en diferentes ubicaciones (pudiendo realizar la búsqueda en paralelo o secuencial entre todas ellas). SIP hace posible esta comunicación gracias a dos protocolos que son RTP/RTCP y SDP, el protocolo RTP se usa para transportar los datos de voz en tiempo real (igual que para el protocolo H.323, mientras que el protocolo SDP se usa para la negociación de las capacidades de los participantes, tipo de codificación, etc.)

Aunque dos terminales SIP puedan comunicarse sin intervención de infraestructuras SIP (razón por la que el protocolo se define como punto-a-punto o entre pares), este enfoque es impracticable para un servicio público. En ese caso requiere de servidores intermediarios (proxy), elementos de registro y servidores de localización (DNS), utilizando un núcleo de red sencillo (y altamente escalable) con inteligencia distribuida en los extremos de la red, incluida en los terminales (ya sea mediante hardware o software).

El protocolo SIP diferencia entre la dirección física (denominada dirección de contacto), que depende de la IP desde la que se conecte el usuario, y la dirección lógica que es invariable para cada usuario. Al igual que en el correo electrónico, las direcciones lógicas de SIP tienen la forma usuario@dominio, gestionando cada dominio una compañía o proveedor de servicios de comunicaciones a través de un servidor (o varios).

Los principales elementos del sistema SIP son:

- Agentes de Usuario (Terminales)
- Servidores de Registro (Registrar)
- Servidores Intermediarios (Proxys)
- Servidores de Redirección (Redirector).

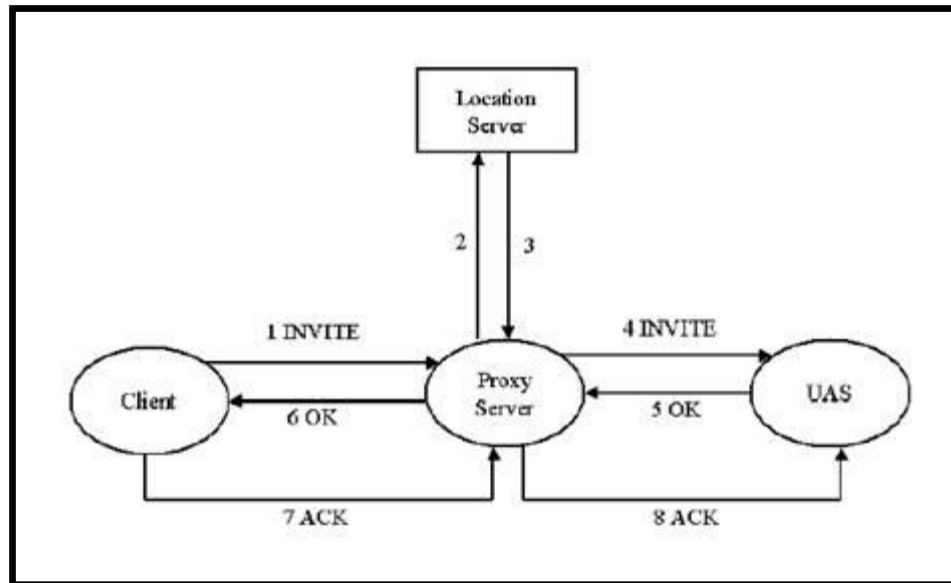


Figura 1-6 Ejemplo de Protocolo SIP

IAX2: El protocolo IAX2 (Inter-Asterisk eXchange v2) es oficialmente el RFC 5456 aprobado por IETF. Este protocolo, que fue diseñado por Mark A. Spencer, Brian Capouch, Ed Guy y Frank Miller, utiliza un único puerto UDP, generalmente el 4569, permite conectar a través de paquetes UDP dos sistemas o terminales hacia un PBX (central telefónica) con Asterisk. Lo relevante es que su publicación como RFC (Request For Comments) permitirá una adopción del mismo a niveles como SIP o H323 en aplicaciones de voz sobre IP; permite gestionar una gran cantidad de codificadores y un gran número de emisiones, lo que significa que puede ser utilizado para transportar virtualmente cualquier tipo de dato. Esta capacidad lo hace muy útil para realizar videoconferencias o realizar presentaciones remotas.

IAX2 soporta Trunking (red), donde un simple enlace permite enviar datos y señalización por múltiples canales. Cuando se realiza Trunking, los datos de múltiples llamadas son manejados en un único conjunto de paquetes, lo que significa que un datagrama IP puede entregar información para más llamadas sin crear latencia adicional. Esto es una gran ventaja para los usuarios de VOIP, donde las cabeceras IP son un gran porcentaje del ancho de banda utilizado.

IAX incluye la capacidad de autenticar en tres formas: en formato de texto, hash MD5, y el intercambio de claves RSA. Esto, por supuesto, no hace nada para cifrar los medios de comunicación o la cabeza es camino entre puntos finales. Muchas de

las soluciones incluyen el uso de una red privada virtual (VPN), aparato o software para cifrar la secuencia en otro nivel de la tecnología, que requiere a los criterios de valoración previa al establecimiento de un método de haber figurado con estos túneles y operacionales. En el futuro, IAX puede ser capaz de cifrar los arroyos entre los puntos finales con el uso de un intercambio de claves RSA, o dinámica de cambio en la palabra clave de configuración, que permite el uso de la clave de renovación automática. Esto sería muy atractivo para crear un vínculo seguro con una institución como su banco. Los diferentes organismos encargados de hacer cumplir la ley, sin embargo, van a querer un cierto nivel de acceso a las conexiones de este tipo.

Elementos del protocolo IAX2

Algunos de los elementos que conforman este protocolo son:

- **Peer** - Es cualquier dispositivo o host que implemente el protocolo IAX2 para comunicarse.
- **Frame** - Es la unidad elemental de comunicación entre dos Peers IAX2. Todos los mensajes del protocolo son enviados en forma de Frames.
- **Elemento de Información (IE)** - Es una unidad discreta de datos que se añade a un frame IAX2 y que contiene datos específicos a una llamada o usuario.
- **Uniform Resource Identifier (URI) de IAX** - Al igual que SIP, IAX utiliza el mecanismo de URI, en su caso en particular, IAX sigue la siguiente sintaxis:

iax:[username@]host[:port][/number[?context]]	
iax	Literalmente IAX
Username	Cadena usada para propósitos de identificación.
Host	Dominio del recurso. Puede ser el FQDN, una dirección IPv4 o [IPv6] encerrada entre corchetes.
Port	Puerto UDP para IAX
Number	El número del peer al que se desea contactar.
Context	El subconjunto de recursos al cual pertenece el usuario llamado dentro del host solicitado.

Tabla 1-1 URI de IAX

Ejemplos:

iax:VOIP.unam.mx/israel

iax:ucol.mx:4569/yocelin

iax:192.0.2.4:4569/israel?gtVOIP

iax:[2001:db8::1]:4569/israel?gtVOIP

iax:ejemplo.com/12022561414

iax:gio@VOIP.unam.mx/12022561414

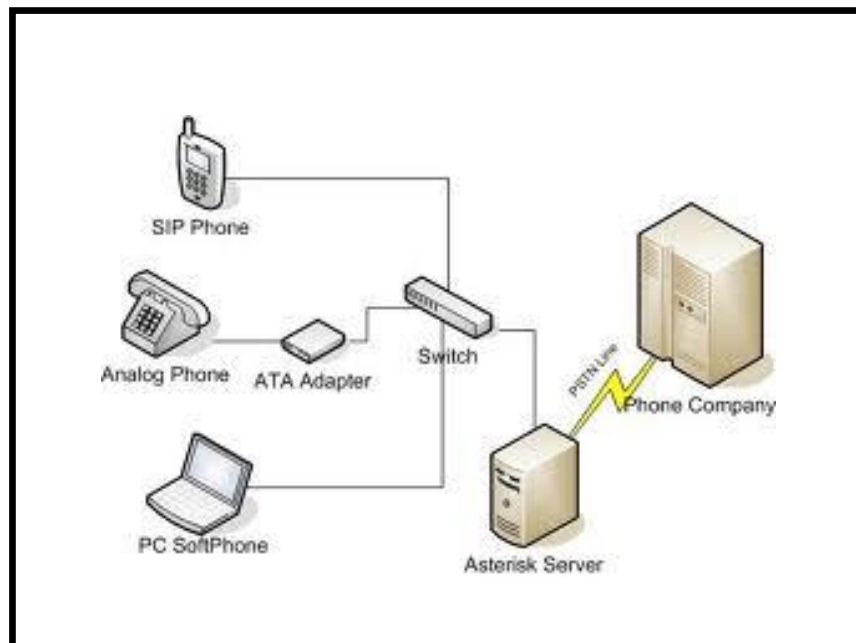


Figura 1-7 Ejemplo de Protocol IAX2

Tabla comparativa de protocolos VOIP más populares

H323		SIP	MGCP/H.248/Megaco	IAX2
Estándar	ITU	IETF	IETF/ITU (H.248)	Digium(c) (IETF RFC5456)
Arquitectura	Distribuida, peer2peer	Distribuida, peer2peer	Centralizada, Cliente - Servidor	peer2peer
Texto/binario	-	Texto	-	Binario
Control de llamada	Gatekeeper	Proxy SIP / Redirect Server / B2BUA	Call agent / Media Control Gateway / Softswitch	-
Multimedia	Si	Si	Si	Si
Transporte medios	RTP	RTP	RTP	En el mismo flujo
Transporte	TCP/UDP	TCP/UDP	TCP/UDP	-

Tabla 1-2 Comparativa de Protocolos VOIP más populares

1.3.3 Parámetros de la VOIP

Este es el principal problema, que presenta hoy en día la penetración tanto de VOIP como de todas las aplicaciones de IP. Garantizar la calidad de servicio sobre Internet, que solo soporta "mejor esfuerzo" (best effort) y puede tener limitaciones de ancho de banda en la ruta, actualmente no es posible; por eso, se presentan diversos problemas en cuanto a garantizar la calidad del servicio.

Códecs

La comunicación de voz es analógica, mientras que la red de datos es digital. El proceso de convertir ondas analógicas a información digital se hace con un codificador-decodificador (el CODEC). Según el Códec utilizado en la transmisión, se utilizará más o menos ancho de banda. La cantidad de ancho de banda utilizada suele ser directamente proporcional a la calidad de los datos transmitidos; además el CODEC comprime la secuencia de datos, y proporciona la cancelación del eco. La compresión de la forma de onda representada puede permitir el ahorro del ancho de banda. Esto es especialmente interesante en los enlaces de poca capacidad y permite tener un mayor número de conexiones de VOIP simultáneamente. Otra manera de ahorrar ancho de banda es el uso de la supresión del silencio, que es el proceso de no enviar los paquetes de la voz entre silencios en conversaciones humanas.

A continuación se muestra una tabla resumen con los códecs más utilizados actualmente:

- El Bit Rate indica la cantidad de información que se manda por segundo.
- El Sampling Rate indica la frecuencia de muestreo de la señal vocal.(cada cuanto se toma una muestra de la señal analógica)
- El Frame size indica cada cuantos milisegundos se envía un paquete con la información sonora.
- El MOS indica la calidad general del códec (valor de 1 a 5)

Nombre	Estandarizado	Descripción	Bit rate (kb/s)	Sampling rate (kHz)	Frame size (ms)	Observaciones	MOS (Mean Opinion Score)
G.711	ITU-T	Pulse Code Modulation (PCM)	64	8	Muestreada	Tiene dos versiones u-law (US, Japan) y a-law (Europa) para muestrear la señal	4.1
G.721	ITU-T	Adaptive differential pulse code modulation (ADPCM)	32	8	Muestreada	Obsoleta. Se ha transformado en la G.726.	
G.722	ITU-T	7 kHz audio-coding within 64 kb/s	64	16	Muestreada	Divide los 16 Khz en dos bandas cada una usando ADPCM	
G.722.1	ITU-T	Codificación a 24 y 32 kb/s para sistemas sin manos con baja perdida de paquetes	24/32	16	20		
G.723	ITU-T	Extensión de la norma G.721 a 24 y 40 kb/s para aplicaciones en circuitos digitales.	24/40	8	Muestreada	Obsoleta por G.726. Es totalmente diferente de G.723.1.	
G.723.1	ITU-T	Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kb/s	5.6/6.3	8	30	Parte de H.324 video conferencing. Codifica la señal usando linear predictive analysis-by-synthesis coding. Para el codificador de high rate utiliza Multipulse	3.8-3.9

						Maximum Likelihood Quantization (MP-MLQ) y para el de low-rate usa Algebraic-Code-Excited Linear-Prediction (ACELP).	
G.726	ITU-T	40, 32, 24, 16 kb/s adaptive differential pulse code modulation (ADPCM)	16/24/32 /40	8	Muestreada	ADPCM; reemplaza a G.721 y G.723.	3.85
G.727	ITU-T	5-, 4-, 3- and 2-bit/sample embedded adaptive differential pulse code modulation (ADPCM)	var.		Muestreada	ADPCM. Relacionada con G.726.	
G.728	ITU-T	Coding of speech at 16 kb/s using low-delay code excited linear prediction	16	8	2.5	CELP.	3.61
G.729	ITU-T	Coding of speech at 8 kb/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)	8	8	10	Bajo retardo (15 ms)	3.92
GSM 06.10	ETSI	RegularPulse Excitation LongTerm Predictor (RPE-LTP)	13	8	22.5	Usado por la tecnología celular GSM	
LPC10	Gobierno de USA	Linear-predictive códec	2.4	8	22.5	10 coeficientes. La voz suena un poco "robótica"	
Speex			8, 16, 32	2.15-24.6 (NB) 4-44.2 (WB)	30 (NB) 34 (WB)		
iLBC			8	13.3	30		

DoD CELP	American Department of Defense (DoD) Gobierno de USA		4.8		30		
EVRC	3GPP2	Enhanced Variable Rate CODEC	9.6/4.8/1. 2	8	20	Se usa en redes CDMA	
DVI	Interactive Multimedia Association (IMA)	DVI4 uses an adaptive delta pulse code modulation (ADPCM)	32	Variable	Muestreada		
L16		Uncompressed audio data samples	128	Variable	Muestreada		

Tabla 1-3 Principales Códecs

Retardo o latencia

A la latencia también se la conoce como retardo. No es un problema específico de las redes no orientadas a conexión y por tanto tampoco de la VOIP. Es un problema general de las redes de telecomunicaciones. Por ejemplo, la latencia en los enlaces vía satélite es muy elevada por las distancias que debe recorrer la información. La latencia se define técnicamente en VOIP como el tiempo que tarda un paquete en llegar desde la fuente al destino. La latencia, junto con el ancho de banda, son determinantes para la velocidad de una red.

Las comunicaciones en tiempo real (como VOIP) y full-dúplex son sensibles a este efecto. Es el problema de "pisarnos". Al igual que el jitter, es un problema frecuente en enlaces lentos o congestionados; la latencia o retardo entre el punto inicial y final de la comunicación debiera ser inferior a 150 ms. El oído humano es capaz de detectar latencias de unos 250 ms, 200 ms en el caso de personas bastante sensibles. Si se supera ese umbral la comunicación se vuelve molesta.

Pérdida de tramas (Frames Lost)

Durante su recorrido por la red IP las tramas se pueden perder como resultado de una congestión de red o corrupción de datos. Además, para tráfico de tiempo real como la voz, la retransmisión de tramas perdidas en la capa de transporte no es práctica por ocasionar retardos adicionales. Por consiguiente, los terminales de voz tienen que retransmitir con muestras de voz perdidas, también llamadas Frame Erasures. El efecto de las tramas perdidas en la calidad de voz depende de cómo los terminales gestionen las Frame Erasures.

En el caso más simple si se pierde una muestra de voz el terminal dejará un intervalo en el flujo de voz. Si muchas tramas se pierden, sonara grietoso con silabas o palabras perdidas. Una posible estrategia de recuperación es reproducir las muestras de voz previas. Esto funciona bien si sólo unas cuantas muestras son perdidas. Para combatir mejor las ráfagas de errores usualmente se emplean sistemas de interpolación. Basándose en muestras de voz previas, el decodificador predecirá las tramas perdidas. Esta técnica es conocida como Packet Loss Concealment (PLC).

Calidad del servicio (QoS)

La calidad de los servicio de VOIP, sistemas de VOIP es a veces un problema con el que nos solemos encontrar. No escuchamos bien, la comunicación se entrecorta, etc.

Los problemas son muchas veces inherentes a la utilización de la red (Internet y su velocidad y ancho de banda); pero si conocemos las causas que pueden producir estos problemas quizás podamos mejorar la calidad, los principales problemas en cuanto a la calidad del servicio (QoS) de una red de VOIP, son la Latencia, el Jitter, la pérdida de paquetes y el Eco.

La calidad de este servicio se está logrando bajo los siguientes criterios:

- La supresión de silencios, otorga más eficiencia a la hora de realizar una transmisión de voz, ya que se aprovecha mejor el ancho de banda al transmitir menos información.
- Compresión de cabeceras aplicando los estándares RTP/RTCP.
- Priorización de los paquetes que requieran menor latencia. Las tendencias actuales son:
 - CQ (Custom Queuing) (Sánchez J.M., VOIP'99): Asigna un porcentaje del ancho de banda disponible.
 - PQ (Priority Queuing) (Sánchez J.M., VOIP'99): Establece prioridad en las colas.
 - WFQ (Weight Fair Queuing) (Sánchez J.M., VOIP'99): Se asigna la prioridad al tráfico de menos carga.
 - DiffServ: Evita tablas de encaminados intermedios y establece decisiones de rutas por paquete.
- La implantación de IPv6, que proporciona mayor espacio de direccionamiento y la posibilidad de tunneling.

Ventajas

- La principal ventaja de este tipo de servicios es que evita los cargos altos de telefonía (principalmente de larga distancia) que son usuales de las compañías de la Red Pública Telefónica Conmutada (PSTN).
- Ahorros en el costo son debidos a utilizar una misma red para llevar voz y datos, especialmente cuando los usuarios tienen sin utilizar toda la capacidad de una red ya existente la cual pueden usar para VOIP sin coste adicional.
- El desarrollo de códecs para VOIP (alaw, G.729, G.723, etc.) ha permitido que la voz se codifique en paquetes de datos cada vez más pequeños. Esto

deriva en que las comunicaciones de voz sobre IP requieran anchos de banda muy reducidos. Junto con el avance permanente de las conexiones ADSL en el mercado residencial, éste tipo de comunicaciones están siendo muy populares para llamadas internacionales.

- Hay dos tipos de servicio de PSTN a VOIP: "Discado Entrante Directo" (Direct Inward Dialling: DID) y "Números de acceso". DID conecta a quien hace la llamada directamente con el usuario VOIP, mientras que los Números de acceso requieren que este introduzca el número de extensión del usuario de VOIP. Los Números de acceso son usualmente cobrados como una llamada local para quien hizo la llamada desde la PSTN y gratis para el usuario de VOIP.
- Las extensiones se pueden reubicar simplemente cambiando los teléfonos IP de sitio y punto de red. Los cambios de configuración se pueden hacer en remoto.
- Música en espera basada en archivos tipo MP3.
- Operadora Automática (IVR) o CTI: Integración con sistemas de gestión (CRM), para llamadas automáticas o identificación datos del cliente por su CLID.
- ACD: Distribución de llamadas entrantes entre agentes comerciales.
- IPCC: Aplicaciones específicas para centros de llamadas (Call Centers) o departamentos comerciales.
- Sistema de correo vocal (Voice Mail) integrado con correo electrónico y directorio.
- Movilidad: posibilidad de acceder a su extensión telefónica (IP) en cualquier punto.
- Terminales Duales SIP-GSM: Teléfonos móviles que disponen de interfaz WIFI con protocolo SIP, pudiendo logarse a la centralita SIP.
- No tienes que incomodarte en una cabina, utiliza la misma tecnología desde tu casa.
- La mayoría de los celulares de última generación son compatibles con esta tecnología.

Desventajas

Las desventajas de VOIP pueden ser molestas, pero son relativamente limitadas para el consumidor. Las quejas principales con respecto a VOIP tienen que ver con el abastecimiento del nivel de la calidad del servicio que acostumbran los clientes con la tecnología regular del teléfono. La razón de esto es multifold. VOIP requiere una cantidad grande de datos para ser comprimido y para ser transmitido, después sin comprimir y para ser entregado, todos en una cantidad de tiempo relativamente pequeña. Los problemas se convierten en conversaciones de VOIP cuando este proceso es demasiado largo y los llamadores experimentan uno de dos problemas; el eco o sobre-habla.

- VOIP requiere de una conexión de banda ancha, con la constante expansión que están sufriendo las conexiones de banda ancha, todavía hay hogares que tienen conexiones por modem, este tipo de conectividad no es suficiente para mantener una conversación fluida con VOIP, puede ser difícil que la compañía sepa exactamente cuánto anchura de banda debe proporcionar - especialmente si el acceso del Internet, la comunicación video u otros servicios de transmisión de datos están utilizando la misma trayectoria.
- VOIP requiere de una conexión eléctrica, en caso de un corte eléctrico a diferencia de los teléfonos VOIP los teléfonos de la telefonía convencional siguen funcionando (excepto que se trate de teléfonos inalámbricos). Esto es así porque el cable telefónico es todo lo que un teléfono convencional necesita para funcionar.
- Llamadas al 911: Estas también son un problema con un sistema de telefonía VOIP. Como se sabe, la telefonía IP utiliza direcciones IP para identificar un número telefónico determinado, el problema es que no existe forma de asociar una dirección IP a un área geográfica, como cada ubicación geográfica tiene un número de emergencias en particular no es posible hacer una relación entre un número telefónico y su correspondiente sección en el 911.
- Dado que VOIP utiliza una conexión de red la calidad del servicio se ve afectado por la calidad de esta línea de datos, esto quiere decir que la calidad de una conexión VOIP se puede ver afectada por problemas como la alta latencia (tiempo de respuesta) o la pérdida de paquetes. Las conversaciones

telefónicas se pueden ver distorsionadas o incluso cortadas por este tipo de problemas.

- VOIP es susceptible a virus, gusanos y hacking,
- En los casos en que se utilice un softphone la calidad de la comunicación VOIP se puede ver afectada por la PC, digamos que estamos realizando una llamada y en un determinado momento se abre un programa que utiliza el 100% de la capacidad de nuestro CPU, en este caso crítico la calidad de la comunicación VOIP se puede ver comprometida porque el procesador se encuentra trabajando a tiempo completo, por eso, es recomendable utilizar un buen equipo junto con su configuración VOIP.
- El inconveniente en este tipo de transmisión son los retrasos ya que las llamadas telefónicas tienen un retraso de 10 milisegundos y la Tecnología VOIP puede llegar a tener un retraso de 400 milisegundos, cortes, y a veces pérdidas de información, debido a que los paquetes, siguen diferentes rutas para llegar a su destino.

CAPITULO 2: ANTECEDENTES

2.1 Asterisk

Asterisk es un software para manejar una central telefónica con capacidad para voz sobre IP (IP-PBX) de licencia libre (GNU). Como cualquier centralita PBX permite interconectar teléfonos y conectar dichos teléfonos a la red telefónica convencional; se trata de una central telefónica con diferentes características, que en tiempos anteriores solo eran accesibles mediante la compra de productos costosos; tales como creación de extensiones, envío de mensajes de voz a e-mail, llamadas en conferencia, menús de voz interactivos(IVR) y distribución automática de llamadas; además se pueden crear nuevas funcionalidades mediante el propio lenguaje de Asterisk o módulos escritos en C o mediante scripts AGI escritos en los diferentes lenguajes.

Para poder utilizar teléfonos convencionales en un servidor Linux corriendo Asterisk o para conectar a una línea de teléfono analógica se necesita hardware especial (no vale con un modem ordinario); pero quizás lo más interesante es que Asterisk soporta numerosos protocolos de Voz sobre IP como SIP, IAX, ZAP y H.323. Asterisk puede operar con muchos teléfonos SIP, actuando como "registrar" o como "Gateway" o entre teléfonos IP y la red telefónica convencional. Los desarrolladores de Asterisk han diseñado un nuevo protocolo llamado IAX para una correcta optimización de las conexiones entre centralitas Asterisk.

Al soportar una mezcla de la telefonía tradicional y los servicios de Voz sobre IP, Asterisk permite a los desarrolladores construir nuevos sistemas telefónicos de forma eficiente o migrar de forma gradual los sistemas existentes a las nuevas tecnologías; algunos sitios usan Asterisk para reemplazar a antiguas centralitas propietarias, otros para proveer funcionalidades adicionales y algunas otras para reducir costos en llamadas a larga distancia utilizando Internet.

2.2 Software

2.2.1 Elastix

Elastix fue creado y actualmente es mantenido por la compañía ecuatoriana Palosanto Solutions. Elastix fue liberado por primera vez en marzo de 2006 pero no se trataba de una distribución sino más bien de una interfaz para mostrar registros de detalles de llamadas para Asterisk, fue recién a finales de diciembre de 2006 cuando

se lo lanzó como una distribución que contenía muchas herramientas interesantes administrables bajo una misma interfaz Web que llamó la atención por su usabilidad.

Elastix es una interfaz libre de Servidor de Comunicaciones Unificadas que integra en un solo paquete: VOIP PBX, fax, mensajería instantánea, correo electrónico, CallCenter; desde entonces hasta la fecha esta distribución no ha parado de crecer en popularidad y actualmente es una de las preferidas del mercado.

2.2.2 Softphone

Para nuestro objetivo de comunicar terminales de servidores IP-PBX diferentes buscamos herramientas de libre acceso para lograrlo, para lo cual optamos por los teléfonos en software o softphones que se han vuelto muy populares por su bajo costo y utilizaremos el softphone ZOIPER.



Figura 2-1 Softphone ZOIPER

2.3 Hardware: Requerimientos del Sistema

La configuración óptima del hardware que necesite nuestro servidor de comunicaciones dependerá básicamente de la cantidad de canales que deseemos que estén operando simultáneamente. Asterisk nos proporciona una guía que permite que el sistema funcione correctamente sin perder la calidad en la comunicación.

Asterisk tiene la necesidad primordial que es el procesador, debido a que lo usa para hacer el procesamiento de los canales de voz. Para construir un Asterisk PBX es necesario:

- Un procesador compatible con Intel mínimo un Pentium 300Mhz.
- Memoria RAM 256 MB.

- Espacio en disco, cerca de 100MB compilados, luego dependiendo de las demás aplicaciones como más código fuente, buzón de voz y grabaciones, deberemos seleccionar la cantidad de disco duro que necesitemos.

2.4 Sintetización de Voz

Sintetización de voz es la tecnología que permite convertir el texto a audio. También se conoce como TTS debido al término Text-To-Speech (texto-a-habla en inglés); se remonta a algunos siglos atrás cuando se intentó sintetizar voz mecánicamente, con aparatos que simulaban el aparato vocal humano. En ese entonces no fueron más que juguetes. Ya en la década de los 30 los laboratorios Bell desarrollaron el famoso VOCODER, que era al mismo tiempo un aparato de reconocimiento de voz que utilizaba electrónica y que podía producir sonido inteligible; su finalidad original era la de reconocer voz para luego poder transmitirla de manera codificada por un medio y sintetizarla en el otro extremo.

El uso de la computación de allí en adelante hizo que fuera posible desarrollar tecnología de Sintetización de voz que fuera práctica y que pudiera estar al alcance de cualquiera que pudiera pagar por ella. Debido a que en un principio muchas de estas tecnologías desarrolladas fueron propietarias y su oferta limitada; sus precios fueron elevados y orientados a grandes empresas o bancos; actualmente ya existen alternativas de Sintetización de voz de relativo bajo costo e inclusive de libre uso como Festival.

2.4.1 Funcionamiento

Veamos un poco de cómo funciona la técnica de Sintetización de voz moderna como la usada por Festival, podemos dividir al proceso de convertir el texto a voz en 3 partes:

2.4.1.1 Análisis de texto

En la primera etapa se normaliza el texto, es decir traducirlo a un formato de palabras estándar y por lo tanto más entendible para ser manipulado en la siguiente etapa. El texto antes de la normalización puede contener palabras mezcladas con números o abreviaciones que no son aptas para la conversión a audio. Por ejemplo, una frase como:

“Cuesta \$100 mil dólares y debes aumentar el 12% por el iva”

Se podría escribir como:

“Cuesta cien mil dólares y debes aumentar el doce por ciento por el iva”

Para fines de sintetizarla a voz.

2.4.1.2 Análisis lingüístico

En esta etapa se convierte el texto normalizado en fonemas. Los fonemas son las unidades fonéticas que componen un lenguaje hablado. Son sonidos únicos y diferentes entre sí; una palabra está compuesta por uno o varios fonemas, para traducir el texto normalizado en fonemas hacen falta bastante trabajo pues se requiere entre otras cosas conocer la entonación de la palabra pues no es lo mismo si se trata de un contexto de pregunta o exclamación.

2.4.1.3 Generación de audio

Una vez que tenemos el texto en una representación de fonemas nos falta traducir esos fonemas a audio. Habrá que unir o concatenar el audio correspondiente a cada fonema e intercalarlo con los silencios que van entre las palabras; sin embargo, no es tan sencillo como suena pues el simple hecho de unir los fonemas audibles puede resultar en un audio de baja calidad que suene robótico o inhumano.

Existen algunas técnicas o algoritmos que se pueden usar para resolver este problema y hacer que la transición entre fonemas sea lo más suave posible. El algoritmo por omisión usado por Festival se llama *Residual Excited Linear Prediction* o simplemente RELP o RE-LPC. (Landivar, pág. 64)



Figura 2-2 Conversión de Texto a Voz con Festival

CAPITULO 3: DEFINICIÓN DEL PROYECTO

3.1 Descripción detallada del proyecto

El proyecto consiste en realizar una centralilla IP-PBX en la cual se encuentra instalado Asterisk, de tal manera que nos permite realizar y recibir llamadas desde las extensiones del servidor local hacia las otras extensiones, poder realizar llamadas fuera de la red interna y poder recibirlas, mediante la utilización de una tarjeta voip, además de poder enviar fax y realizar mensajería instantánea entre los diferentes usuarios.

Para lo cual tenemos un servidor en el que tenemos instalado Asterisk y Elastix; Asterisk lo utilizaremos para realizar las configuraciones de la troncal, extensiones e IVR; además estará implementado el módulo de Sintetización de voz que nos ayudara para realizar la configuración del IVR de dicha centralilla, Elastix es una herramienta que nos proporciona un interfaz grafica vía web la cual nos ayudara a configurar: OpenFire, HylaFax y CallCenter. Además realizaremos un plan de marcado avanzado en el cual pretendemos realizar, que el usuario pueda preguntar a cuanto asciende su deuda ya sea mediante su cedula o el número de factura.

3.2 Definición del plan de Marcado

El plan de marcación nos sirve para llevar un control ordenado de las extensiones que crearemos en nuestra centralita. Esto nos ayuda a generar un número único para cada extensión. De esta manera por ejemplo nuestro número podría estar formado de la siguiente forma: un código de departamento (CD) y el número de nuestra extensión en la centralita, de esta manera nos aseguramos que no habrá ninguna extensión con algún número repetido.

Para nuestro ejemplo realizaremos 5 extensiones:

- 4 extensiones que estarán conectadas a softphones que pertenecerán a los Departamentos de :
 - Contabilidad código de departamento 20 en la cual tendremos la extensión 10 que será para el jefe de departamento
 - Sistemas código de departamento 30 en la cual tendremos la extensión 10 que será para el jefe de departamento

- Facturación código de departamento 40 en la cual tendremos la extensión 10 que será para el jefe de departamento
- Gerencia código de departamento 10 en la cual tendremos la extensión 20 que será para la secretaria.
- Una extensión que estará conectada a un teléfono IP será para la Gerencia código 10 en la cual tendremos la extensión 10 que será para el Gerente.

Con lo cual nuestras extensiones quedarían de la siguiente manera:

1010 para el Gerente

1020 para la Secretaria del Gerente

2010 jefe de departamento de contabilidad

3010 jefe de departamento de sistemas

4010 jefe de departamento de facturación

3.3 Diagrama de Conexiones

Mostraremos el diagrama de Conexiones que se van a realizar:

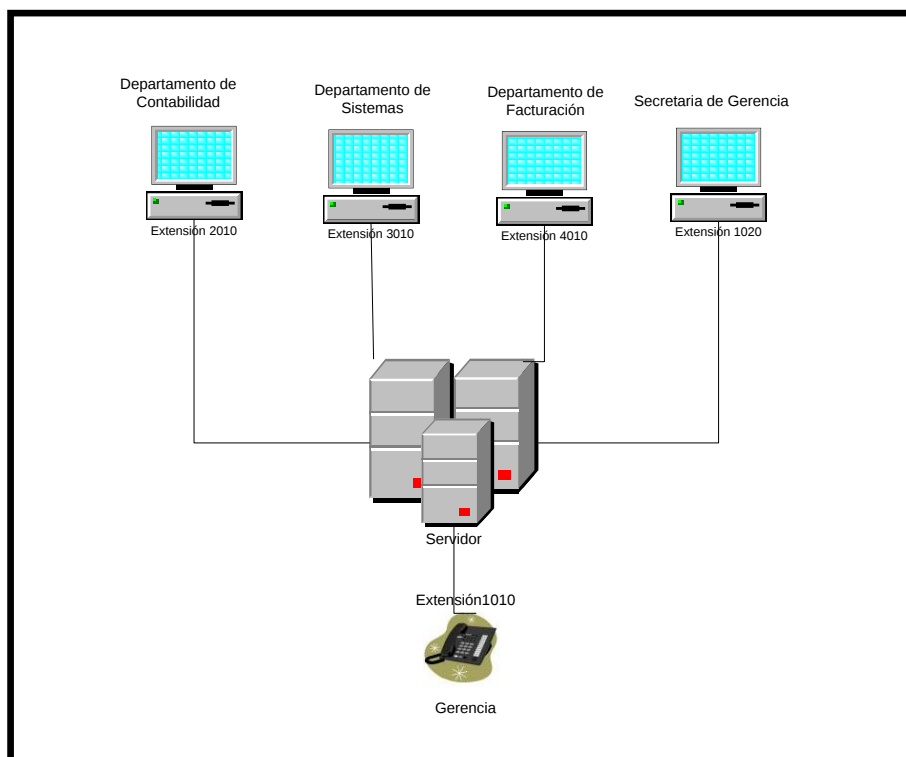


Figura 3-1 Diagrama de Conexiones

CAPITULO 4: CONFIGURACION DEL PROYECTO

En el siguiente capítulo desarrollaremos dos partes en la primera parte realizaremos la configuración de los archivos Asterisk con lo cual configuraremos lo que son las extensiones, IVR y la troncal; además configuraremos el módulo de Festival el cual sirve para realizar la Sintetización de voz. En la segunda parte realizaremos la configuración mediante Elastix de OpenFire que es para mensajería instantánea, el HylaFax que es para realizar envío y recepción de fax y el CallCenter que nos ayudara a realizar campañas (llamadas en donde se pide o se recepta información) de entrada y salida.

4.1 Configuración de Archivos Asterisk

4.1.1 Configuración de Extensiones

Para nuestra proyecto vamos a configurar 2 tipos de extensiones la SIP y la IAX2; la cual utilizaremos para configurar más adelante el fax virtual (HylaFax), la configuración de las extensiones se las realizan modificando los archivos ubicados en el path **/etc/asterisk/**; aquí debemos modificar los archivos *<extensions.conf>*, *<sip_additional.conf>* para las extensiones del tipo SIP y el *<iax_additional.conf>* para las extensiones del tipo IAX2, para lo cual vamos a aplicar los contextos.

En el archivo *<iax_additional.conf>*, nos sirve para configurar las extensiones de tipo IAX2 para lo cual agregamos el contexto con el nombre de la extensión:

```
[1020]
type=friend
setvar=REALCALLERIDNUM=1020
secret=1234
record_out=Adhoc
record_in=Adhoc
qualify=yes
port=4569
nottransfer=yes
mailbox=1020@device
host=dynamic
disallow=
```

```
dial=IAX2/1020
context=from-internal
callerid=device <1020>
allow=ullaw
accountcode=
```

En el archivo `<sip_additional.conf>` contiene parámetros con la configuración SIP de Asterisk y agregamos los contextos con el nombre de la extensión:

```
[1010]
type=friend
secret=1234
record_out=Adhoc
record_in=Adhoc
qualify=yes
port=5060
pickupgroup=
nat=yes
mailbox=1010@device
host=dynamic
dtmfmode=rfc2833
disallow=
dial=SIP/1010
context=from-internal
canreinvite=no
callgroup=
callerid=device <1010>
call-limit=4
allow=ullaw
accountcode=
call-limit=50

[2010]
type=friend
secret=1234
```

record_out=Adhoc
record_in=Adhoc
qualify=yes
port=5060
pickupgroup=
nat=yes
mailbox=2010@device
host=dynamic
dtmfmode=rfc2833
disallow=
dial=SIP/2010
context=from-internal
canreinvite=no
callgroup=
callerid=device <2010>
call-limit=4
allow=ullah
accountcode=
call-limit=50

[3010]
type=friend
secret=1234
record_out=Adhoc
record_in=Adhoc
qualify=yes
port=5060
pickupgroup=
nat=yes
mailbox=3010@device
host=dynamic
dtmfmode=rfc2833
disallow=
dial=SIP/3010

context=from-internal
canreinvite=no
callgroup=
callerid=device <3010>
call-limit=4
allow=ullah
accountcode=
call-limit=50

[4010]
type=friend
secret=1234
record_out=Adhoc
record_in=Adhoc
qualify=yes
port=5060
pickupgroup=
nat=yes
mailbox=4010@device
host=dynamic
dtmfmode=rfc2833
disallow=
dial=SIP/4010
context=from-internal
canreinvite=no
callgroup=
callerid=device <4010>
call-limit=4
allow=ullah
accountcode=
call-limit=50

En el archivo de extensions.conf lo modificamos de la siguiente manera:

[from-local]

Include =>from-internal

[from-internal]

exten => 1010,1,Dial(SIP/1010)

exten => 1010,n,Hangup

exten => 2010,1,Dial(SIP/2010)

exten => 2010,n,Hangup

exten => 3010,1,Dial(SIP/3010)

exten => 3010,n,Hangup

exten => 4010,1,Dial(SIP/4010)

exten => 4010,n,Hangup

exten => 1020,1,Dial(IAX2/1020)

exten => 1020,n,Hangup

Ahora procederemos a configurar nuestro teléfono IP o softphone con la extensión 1010:

1. Agregamos el nombre de la cuenta Sip el cual sería Gerencia:

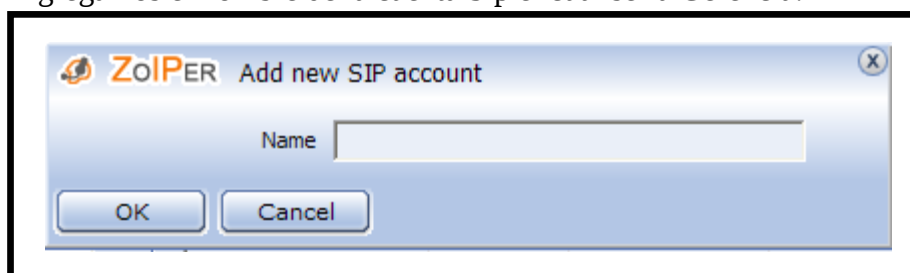


Figura 4-1 Nombre de Cuenta SIP

2. La Configuración de la extensión:

Configuración de Extensión SIP	
Dominio	192.168.1.30
Usuario	1010
Clave	1234
Número Identificador	Gerencia

Tabla 4-1 Configuración de extensión SIP

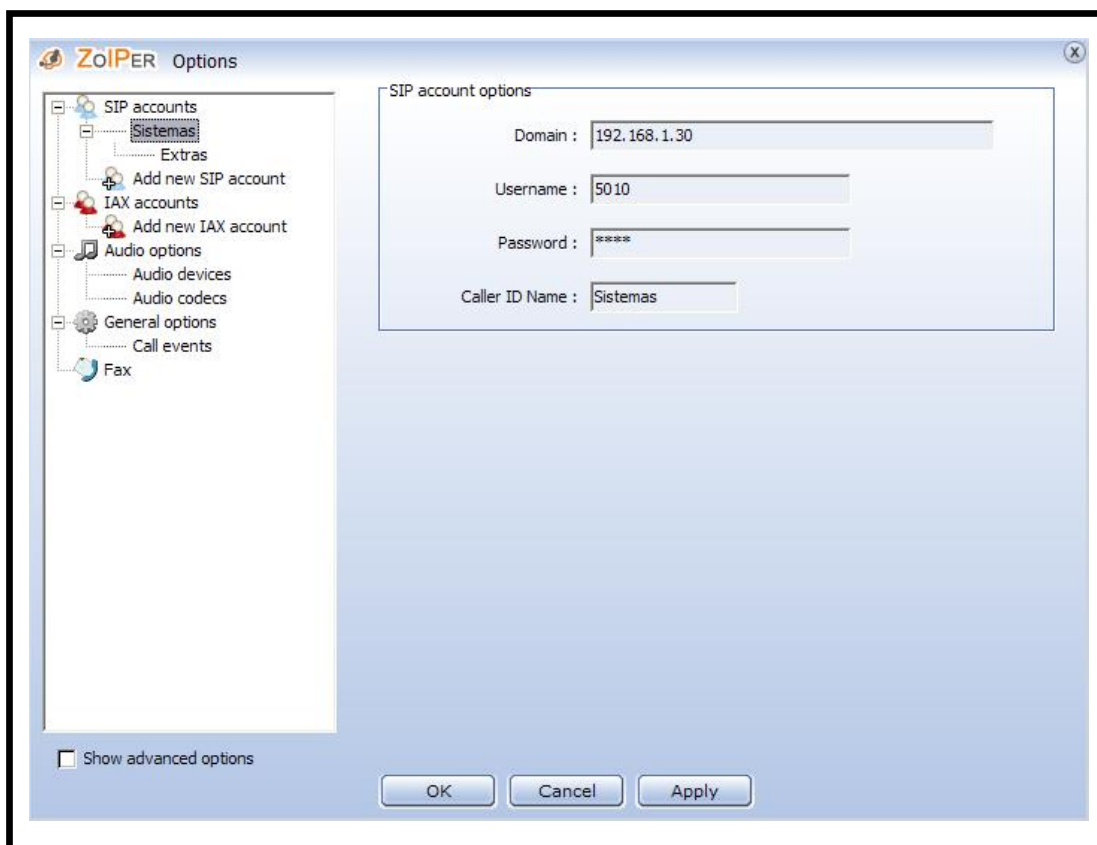


Figura 4-2 Configuración de Cuenta SIP

4.1.2 Configuración de Sintetización de Voz (FESTIVAL)

Al comenzar con la configuración de Festival lo primero que tenemos que realizar es poner la voz en español ya que viene por defecto en ingles para lo cual necesitaremos seguir los siguientes pasos:

1. Necesitamos instalar perl con el comando:
yum -y install perl
2. Necesitamos instalar también una utilidad para crear los paquetes en CentOS con el comando:
yum -y install rpm-build
3. Ahora descargamos Alien en el path **cd /usr/src** con el comando:
wget http://ftp.de.debian.org/debian/pool/main/a/alien/alien_8.78.tar.gz
4. Lo descomprimos con el comando:
tar -xf alien_8.78.tar.gz
5. Entramos en la carpeta y lo instalamos siguiendo la secuencia de estos comando:
cd alien

perl Makefile.PL

make

make install

6. Ingresamos al path `cd /usr/src` y descargamos las voces en español con el comando:

wget http://forja.guadalinux.org/frs/download.php/152/festvox-sflpc16k_1.0.0_all.deb

7. Ahora transformaremos el paquete en `un.rpm` para poder instalar en Centos con el comando:

alien -rv festvox-palpc16k_1.0-1_all.deb

8. Instalamos el paquete con el comando:

rpm -ivh festvox-palpc16k-1.0-2.noarch.rpm

El archivo instalado lo encontraremos en la carpeta `/usr/share/festival/voices/spanish`

9. El siguiente punto es la configuración del archivo de festival vim que se encuentra en el path `/usr/share/festival/festival.scm` y añadimos estas líneas al final:

```
;(language__spanish)
(set! voice_default 'voice_JuntaDeAndalucia_es_sf_diphone)
(define (tts_textasterisk string mode)
  "(tts_textasterisk STRING MODE)
  Apply tts to STRING. This function is specifically designed for
  use in server mode so a single function call may synthesize the string.
  This function name may be added to the server safe functions."
  (let ((wholeutt (utt.synth (eval (list 'Utterance 'Text string)))))
    (utt.wave.resamplewholeutt 8000)
    (utt.wave.rescalewholeutt 5)
    (utt.send.wave.clientwholeutt)))
```

10. Guardamos el archivo y a continuación procedemos a la modificación del archivo de configuración de festival en Asterisk ubicado en el path `/etc/asterisk/festival.conf`; en este archivo hay que modificar unas líneas, para lo cual solo descomentamos las líneas borrando `(;)` y el archivo nos quedaría de la siguiente manera:

`[general]`

```
host=localhost  
port=1314  
festivalcommand=(tts_textasterisk "%s" 'file)  
(quit)\n
```

11. Guardamos el archivo y reiniciamos Asterisk
Amportal restart

En este momento tenemos configurado Festival, para que la sintetización la realice en español, y el comando text2wave, lo utilizaremos para sintetizar la voz y pasarla a un archivo de sonido, que nos transforma el texto en un archivo .WAV, pero como nosotros necesitamos que nos transforme a formato ulaw; tenemos que utilizar el comando de la siguiente manera:

```
text2wave texto.txt -o sonido.ulaw -otypeulaw
```

En donde texto.txt es el archivo de texto que queremos sintetizar, sonido.ulaw es el archivo ya transformado, y utilizamos la opción -otype para que se transforme en el formato ulaw ya que por default lo pone en formato .WAV

4.1.3 Configuración de IVR

Un IVR (Interactive Voice Response) es un menú que se realiza con archivos de sonidos pregrabados, con el cual el usuario puede interactuar mediante pulsaciones DTMF, aquí se reproduce un mensaje dando las diferentes opciones y se espera las pulsaciones del usuario para escoger una de estas opciones; las extensiones “t” e “i”, sirven para controlar si el usuario no pulsa nada en el tiempo fijado o si realiza una pulsación incorrecta respectivamente.

Para la realización de los mensajes utilizados en el IVR podemos utilizar el módulo de festival, al utilizar festival estamos haciendo que el IVR sea de forma dinámica, para lo cual se realiza un script que debe estar ubicado en el path /usr/local/java/Conectores/, con estas instrucciones:

1. Sacamos los idcaller y el número de la extensión de la base de datos y los ponemos en un documento de texto llamado archivo.txt.

```
asterisk -rx "database show"|grep "cidname"|grep -v "default_user"|cut -d/ -f3,4 | tr -d " " > archivo.txt
```

2. Tenemos que leer el archivo línea por línea para ir separando el nombre y el número y ubicamos en un documento de texto, el cual nos servirá para la realización de la Sintetización llamado nombre.txt.
3. Tenemos que transformarlo utilizando el *text2wave* y lo pasamos al formato de ulaw, utilizando el comando.

text2wave nombre.txt número.ulaw -otypeulaw

4. El siguiente paso a realiza es copiar los archivos de sonido creados al directorio /etc/Asterisk/sounds/customs, el script (bsh) quedaría de la siguiente forma:

```
#!/bin/sh
asterisk -rx "database show"|grep "cidname"|grep -v "default_user"|cut -d/ -f3,4 | tr -d " " > archivo.txt
oldIFS=$IFS
```

```
IFS=;;
```

```
while read usuario nombre
```

```
do
```

```
    echo $usuario > usuario.txt
```

```
    echo $nombre > nombre.txt
```

```
    IFS=/;
```

```
    while read número otro
```

```
    do
```

```
        echo $número $nombre > número.txt
```

```
        text2wave número.txt -o $nombre.ulaw -otype ulaw
```

```
        mv $nombre.ulaw /var/lib/asterisk/sounds/custom/
```

```
    done < usuario.txt
```

```
IFS=;;
```

```
done < archivo.txt
```

```
IFS=$old_IFS
```

5. Para que el script se realice automáticamente todos los días ubicamos modificamos el archivo que se encuentra en el path **/etc/rc.local**, nos quedaría de la siguiente manera:

```
#!/bin/sh

# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

/usr/local/sbin/motd.sh > /etc/motd

/usr/sbin/xfxotune -s

/usr/sbin/amportal start_fop

/usr/bin/festival -server

/usr/local/java/Conectores/sonido.sh

/var/lib/asterisk/agi-bin/arrancar.sh
```

6. A continuación realizamos un archivo llamado IVR.java en el cual vamos a proceder a reproducir todos los sonidos antes realizados y a redireccionar a la extensión pedida; si el usuario escoge la opción 4010 nos dará a otro menú en donde tenemos la opciones:

1. Conectarnos con la extensión del departamento de facturación.
2. Pedir el saldo que el cliente debe por su número de cedula.
3. Pedir el saldo que debe por el número de factura.

```
import net.sf.asterisk.fastagi.AGIRequest;
import net.sf.asterisk.fastagi.AGICChannel;
import net.sf.asterisk.fastagi.AGIException;
import net.sf.asterisk.fastagi.AbstractAGIScript;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.nio.channels.GatheringByteChannel;
import java.util.prefs.BackingStoreException;

public class IVR extends AbstractAGIScript {
```

```

    public void service(AGIRequest peticion,AGIChannel canal) throws
    AGIException {
        answer(canal);
        int codigo = 0;
        try {
            File f = new File( "/etc/cron.daily/archivo.txt" );
            BufferedReader entrada;
            String opcion = "";
            String facturacion = "";
            String cidname = "";
            try {
                entrada = new BufferedReader( new FileReader( f ) );
                String linea;
                while (entrada.ready()) {
                    linea = entrada.readLine();
                    String[] lineas= linea.split(":");
                    opcion = getData(canal, "custom/"+lineas[1],3000);
                    if
(opcion.toUpperCase().compareTo("(TIMEOUT)")>0                                &&
Integer.parseInt(opcion)>0) {
                        break;
                    }
                }
            }

            //En el caso de no escoger una opcion se espera un determinado tiempo a esperar
            opcion.

            //En el caso que la opcion sea el departamentodefatura se da otras opciones
                if (opcion.compareTo("4010") == 0) {
            //Este comando ejecuta el agi que puede estar local o remoto simplemente se cambia
            en localhost por una ip en
            //donde se encuentre escuchando el servidor de AGI
                opcion=getData(canal,"custom/facturacion");
                if (opcion.compareTo("1") == 0)
                    codigo = execCommand(canal, "Dial", "SIP/4010");
                else

```

```

        if(opcion.compareTo("2") == 0)
            codigo = execCommand(canal, "AGI",
"agi://localhost/credito.agi");
        else
            if (opcion.compareTo("3") == 0)
                codigo = execCommand(canal, "AGI",
"agi://localhost/credito1.agi");
            } else {
//Caso contrario se llama a la extensión seleccionada anteriormente
                codigo = execCommand(canal, "Dial", "SIP/" + opcion);
            }
        } catch (IOException e) {
//en el caso de escoger una opción inválida posiblemente caiga en este nivel
            e.printStackTrace();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    hangup(canal);
}
}

```

7. Creamos un archivo llamado *<extensión s_custom1.conf>* en el path */etc/asterisk* y dentro del archivo creamos el contexto [IVR], en el cual recibimos la llamada, ejecutamos el archivo *ivr.agi* y esperamos a que el usuario ingrese una extensión y la redireccionamos a la dirección pedida.

```

[IVR]
exten => 185,1,Wait(1)
exten => 185,n,Background(custom/bienvenida)
exten => 185,n,WaitExten(,)
exten => 1,1,goto(IVR1,1,1)
exten => 1,n,Hangup()
exten => i,1,goto(IVR,185,1)
exten => t,1,goto(IVR,185,1)

```

```
exten => h,1,Hangup
```

```
[IVR1]
```

```
exten => 1,1,Set(TIMEOUT(digit)=6)
```

```
exten => 1,n,AGI(agi://localhost/ivr.agi)
```

En el archivo `<extensión s_custom.conf>` incluimos el contexto `[from-pstn-custom]` en el cual agregamos las siguientes líneas y nos quedaría de la siguiente manera:

```
[from-pstn-custom]
```

```
exten => s,1,Answer()
```

```
exten => s,n,goto(IVR,185,1)
```

```
#include extensión s_custom1.conf
```

Otra forma de hacerlo es de solo crear el contexto; donde solo tenemos un archivo de sonido el cual que nos da todas las opciones; lo podemos crear ya sea grabando el texto utilizando a una persona o utilizando un sintetizador de voz (loquendo) y nos quedaría de esta forma:

```
[IVR2]
```

```
exten => 2,1,Set(TIMEOUT(digit)=6)
```

```
exten => 2,n,Background(custom/extensiones)
```

```
exten => 2,n,WaitExten(5)
```

```
exten => 1010,1,Dial(SIP/1010)
```

```
exten => 1010,n,Hangup()
```

```
exten => 2010,1,Dial(SIP/2010)
```

```
exten => 2010,n,Hangup()
```

```
exten => 3010,1,Dial(SIP/3010)
```

```
exten => 3010,n,Hangup()
```

```
exten => 4010,1,Background(custom/facturacion)
```

```
exten => 4010,n,WaitExten(5)
```

```
exten => 1,1,Dial(SIP/4010)
```

```
exten => 1,1,Hangup()
```

```
exten => 2,1,AGI(agi://localhost/credito.agi)
```

```
exten => 3,1,AGI(agi://localhost/credito1.agi)
```

```
exten => 4010,n,Hangup()
```

exten => 1020,1,Dial(IAX2/1020)

exten => 1020,n,Hangup()

exten => i,1,goto(IVR2,2,1)

exten => t,1,goto(IVR2,2,1)

exten => h,1,Hangup()

Modificamos el contexto [IVR] para poder utilizar cualquiera de los 2 IVR:

[IVR]

exten => 185,1,Wait(1)

exten => 185,n,Background(custom/bienvenida)

exten => 185,n,WaitExten(,)

;exten => 185,n,Hangup()

exten => 1,1,goto(IVR1,1,1)

exten => 1,n,Hangup()

exten => 2,1,goto(IVR2,2,1)

exten => 2,n,Hangup()

exten => i,1,goto(IVR,185,1)

exten => t,1,goto(IVR,185,1)

exten => h,1,Hangup()

4.1.4 Configuración de Troncal

Las troncales (trunk) son lo que se utiliza para llevar una ó varias llamadas a un VSP (VOICE SERVICE PROVIDER) ó a algún dispositivo que responda al número que se marcó (por ejemplo, otro Asterisk o la PSTN). Existen cuatro tipos de troncales:

- **SIP** (Session Initiation Protocol)
- **ZAP o DAHDI** (líneas analógicas)
- **IAX2** (Inter Asterisk)
- **Personalizada** (Custom trunk)

Todas las troncales son configuradas de la misma forma.

4.1.4.1 Configuraciones generales de troncales.

Outbound Caller ID

La identificación de llamada saliente al configurar y activar esta opción deshabilitará la identificación de llamadas de todos los clientes a los que se les hagan llamadas desde esta troncal. El formato es “callername” <#####>

Canales máximos

Esto limita el número máximo de canales (llamadas simultáneas) que pueden ser usados por esta troncal, llamadas entrantes y salientes. Déjelo en blanco para no especificar el máximo.

Reglas de marcado saliente

Las reglas de marcado son muy importantes, y simples de aprender. Le indica al servidor cómo las llamadas van a ser marcadas en esta troncal. Puede ser utilizado para agregar ó quitar prefijos. Los números que no tengan una equivalencia con ningún patrón definido aquí serán marcados como estén. Obsérvese que un patrón sin un + ó | (para agregar ó remover un prefijo) es inútil.

Reglas:

X Equivale a cualquier dígito de 0 a 9

Z Equivale a cualquier dígito de 1 a 9

N Equivale a cualquier dígito de 2 a 9

[1237-9] Equivale a cualquier dígito entre corchetes (en este ejemplo 1, 2, 3, 7, 8, 9)

. Equivale a uno ó más caracteres (no permitido antes de un | ó +)

Quita un prefijo de discado del número (por ejemplo: 613|NXXXXXX equivaldrá cuando alguien marque 61335551234, pero sólo ingresará en la troncal 5551234)

Suma un prefijo de discado al número marcado (por ejemplo: + 1613+NXXXXXX equivaldrá cuando alguien marca 5551234 e ingresará a la troncal como 16135551234)

Si usted está configurando una troncal SIP que le pide usar el prefijo 011 para las llamadas internacionales y ninguno para las llamadas a Estados Unidos y Canadá (001XXXXXXXXXX), debería configurar estas dos reglas:

011+N

La primera, agrega el prefijo 011 a todos los número pasados por una “Ruta saliente” (que debe quitar el “00” antes de pasar el número a la troncal) que inicie con un número del 2 al 9.

Prefijo de marcado saliente.

Es usado para añadir un prefijo a la cadena de marcado de todas las llamadas salientes.

Configuración de troncales ZAP o DAHDI

Se debe configurar el hardware que se va a utilizar y los canales ZAP o DAHDI son identificados ya sea por un número de grupo o por un número de canal, el cual está definido en el archivo `/etc/asterisk/dahdi-channels.conf`.

Configuración para troncales SIP e IAX2

Antes de configurar la troncal explicaremos los puntos que hay que configurar:

Parámetros de Configuración de una Troncal	
Nombre de la troncal	Aquí se coloca un nombre (que debe ser único) para identificar la troncal. Es recomendable utilizar el nombre del VSP.
Detalles del troncal de salida	Aquí se colocan los parámetros que permiten autenticar a la central contra el VSP.
Allow	Códecs aceptados en orden de preferencia.
Auth	Nombre de usuario para autenticarse contra el VSP.
Context	Si “type” está fijado en “user”, es el contexto para las llamadas entrantes. Si está fijado en “peer”, es el contexto para las llamadas salientes. Si “type=friend”, es el contexto para llamadas entrantes y salientes.
Host	Es cómo la central busca al servidor del VSP contratado. Los valores posibles son dynamic, hostname o dirección ip.
Insecure	Especifica cómo manejar la conexión con el otro punto. Los valores posibles son: very, yes, no, invite, port.
Port	Puerto de comunicaciones del cliente.
Secret	Si Asterisk® está actuando como un cliente de un servidor SIP remoto, en secret colocamos la contraseña que usará para autenticarse.
Type	Relación de la central con el servidor o cliente remoto. Los valores posibles son: user, peer o friend.
Username	Nombre de usuario para autenticarse con el servidor remoto.

Tabla 4-2 Parámetros de Configuración de una troncal

Cadena de registro (Register string)

La mayoría de los VSP (VOICE SERVER PROVIDER) requieren que la central se registre con el servidor. Los parámetros y forma de los registros son suministrados por el proveedor. En general, suelen tener un formato similar a este:

nombre_de_usuario:contraseña@servidor

Contexto del troncal de entrada

Es normalmente el nombre de la cuenta o el número asignado por el VSP.

Ejemplo de configuración de una troncal del tipo SIP para esto configuraremos el archivo <sip_additional.conf>, el cual nos quedara de la siguiente manera:

```
[trunk-sip]
context = from-pstn
host = dynamic
port = 5080 (Diferente del 5060 para poder usar el FXS en ese puerto)
username = sip
secret = 1234
qualify = yes
nat = no
allow = all
type = friend
```

Nosotros necesitamos configurar un troncal de tipo ZAP para lo cual necesitaremos configurar los archivos <dahdi-channels.conf> y el archivo <chan_dahdi.conf>

Chan_dahdi.conf

```
[trunkgroups]
[channels]
context=from-pstn
signalling=fxs_ks
rxwink=300 ; Atlas seems to use long (250ms) winks
usecallerid=yes
hidecallerid=no
callwaiting=yes
```

```

usecallingpres=yes
callwaitingcallerid=yes
threewaycalling=yes
transfer=yes
canpark=yes
cancallforward=yes
callreturn=yes
echocancel=yes
echocancelwhenbridged=no
faxdetect=incoming
echotraining=800
rxgain=3.0
txgain=3.0
callgroup=1
pickupgroup=1
;busydetect=yes
;busycount=3
immediate=no
context=from-pstn
#include dahdi-channels.conf
;#include chan_dahdi_additional.conf
Dahdi_channels.conf
signalling=fxs_ks
callerid=asreceived
group=1
context=from-pstn
channel => 2

```

4.2 Configuración de Elastix

4.2.1 Configuración de Mensajería Instantánea (OPENFIRE)

EL módulo de mensajería instantánea (OpenFire) por defecto se encuentra desactivado entonces nuestro primer paso es activarlo:

1. Damos clic en la pestaña de IM nos saldrá un mensaje indicando si queremos activar el sistema damos clic en el link [clic here](#)



Figura 4-3 Habilitar Openfire

2. Procedemos a escoger el idioma con el cual vamos a realizar el resto de las configuraciones de la mensajería instantánea y seguidamente le damos en continuar.

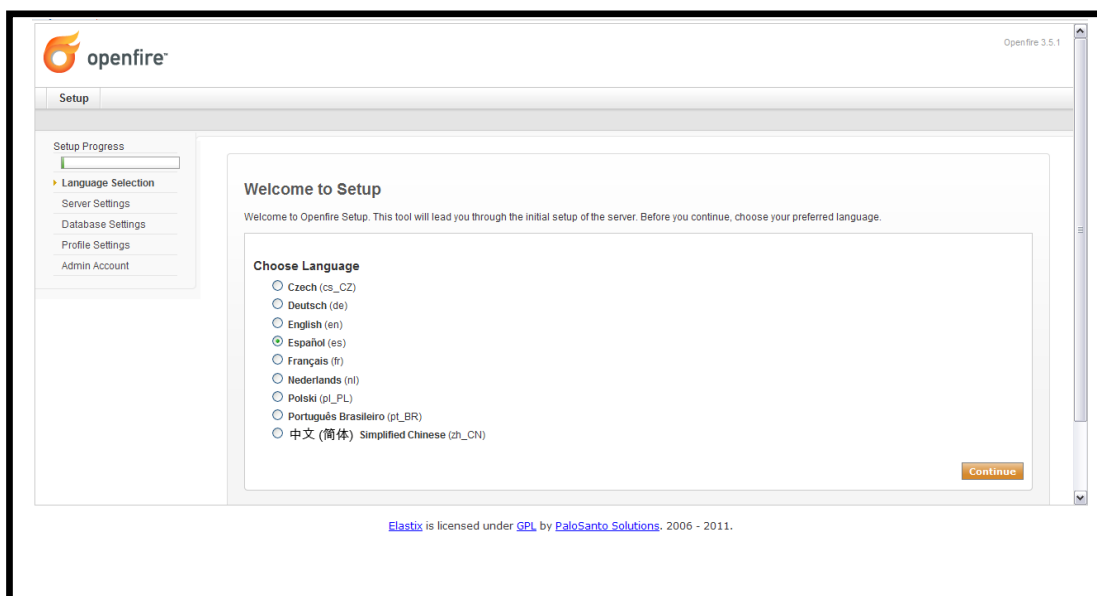


Figura 4-4 Configuración de Idioma

3. En el siguiente paso tendremos que especificar el nombre del dominio del servidor esto deberá ser los mismos que se configuro en los parámetros de red.

- Primero nos pide que ingresemos el dominio aquí tenemos 2 alternativas:
 - ingresar el nombre del dominio
 - la dirección IP del servidor.
- Los otros 2 aspectos son el puerto de consola de administración y el puerto de consola de administración segura se los deja por default los que están.
- Procedemos a dar clic en continuar.

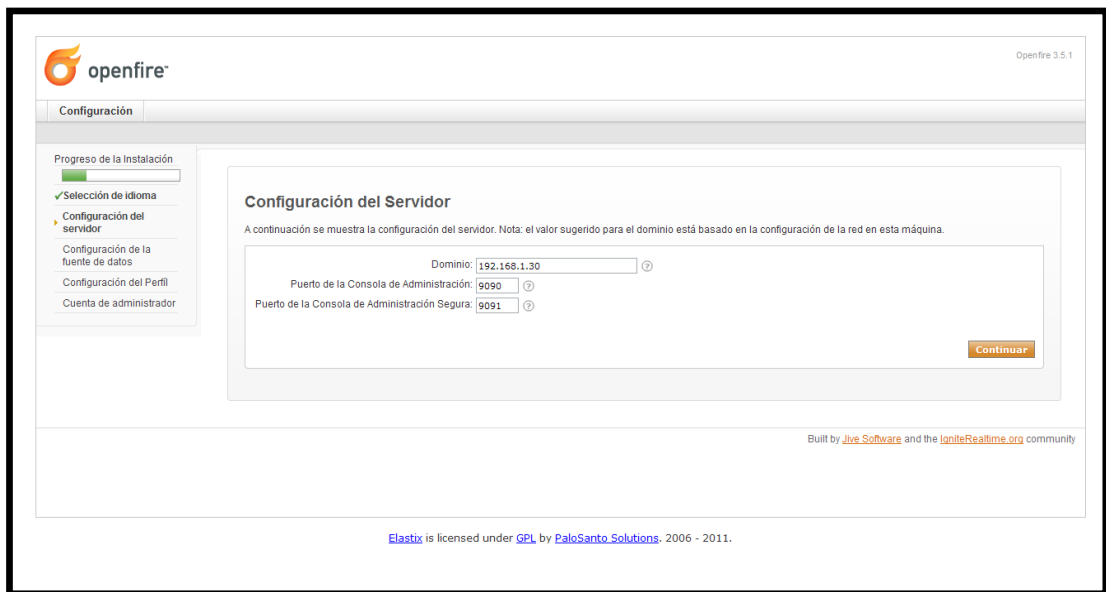


Figura 4-5 Configuración del Servidor

4. La siguiente configuración es de la fuente de datos, que no es nada más que la base de datos donde OPENFIRE almacenara la configuración y la información que debe ir almacenando conforme su almacenamiento.
 - Aquí podemos usar un motor de base de datos estándar (Ej. MySQL) o podemos usar una base de datos embebida que trae OOPENFIRE y que nos facilita las tareas de configuración si es que no vemos ninguna ventaja en usar un motor estándar.
 - En la mayoría de los casos es conveniente elegir la base de datos embebida.
 - Procedemos a dar clic en continúe.

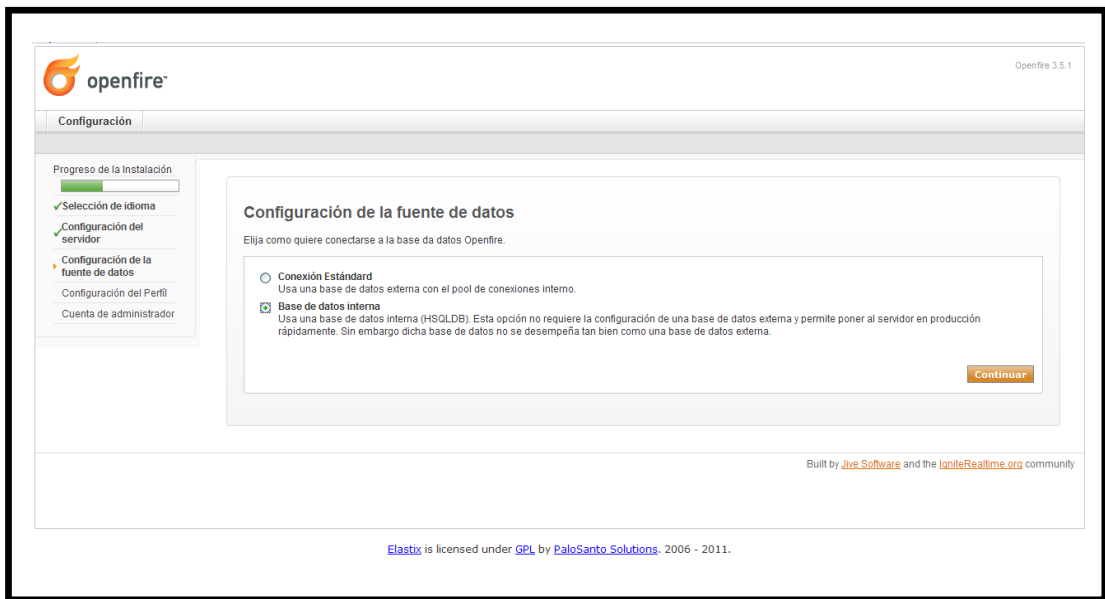


Figura 4-6 Configuración de la Fuente de Datos

En la pantalla que nos aparece después de haber elegido conexión estándar es para configurar la base de datos que vamos a utilizar, el driver a utilizar, el url de la base de datos.

5. El siguiente punto a configurar es el seteo del perfil, esto se refiere a donde se almacenaran los usuarios del OPENFIRE y tenemos 3 opciones:

- **por defecto:-** almacena en la base del OPENFIRE
- **Servidor de Directorio:-** integración con Active Directory o OpenLAD utilizando el protocolo LPDA
- **Integración con Clearspace:-** Integrar con una instalación existente de Clearspace.

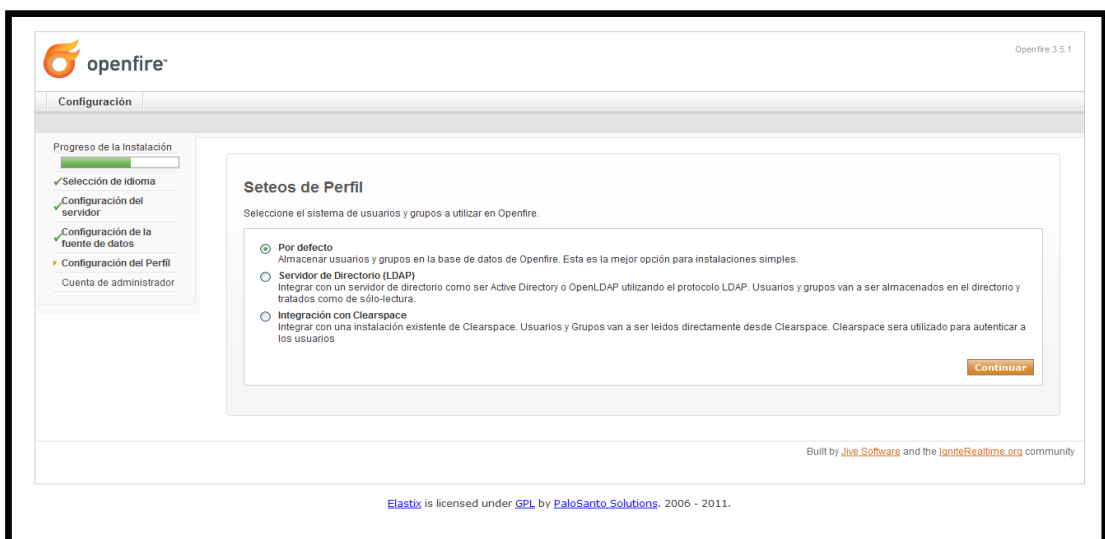


Figura 4-7 Seteo de Perfil

6. El siguiente punto a configurar es la cuenta del Administrador del Openfire en donde introducimos una nueva contraseña y el email del administrador.

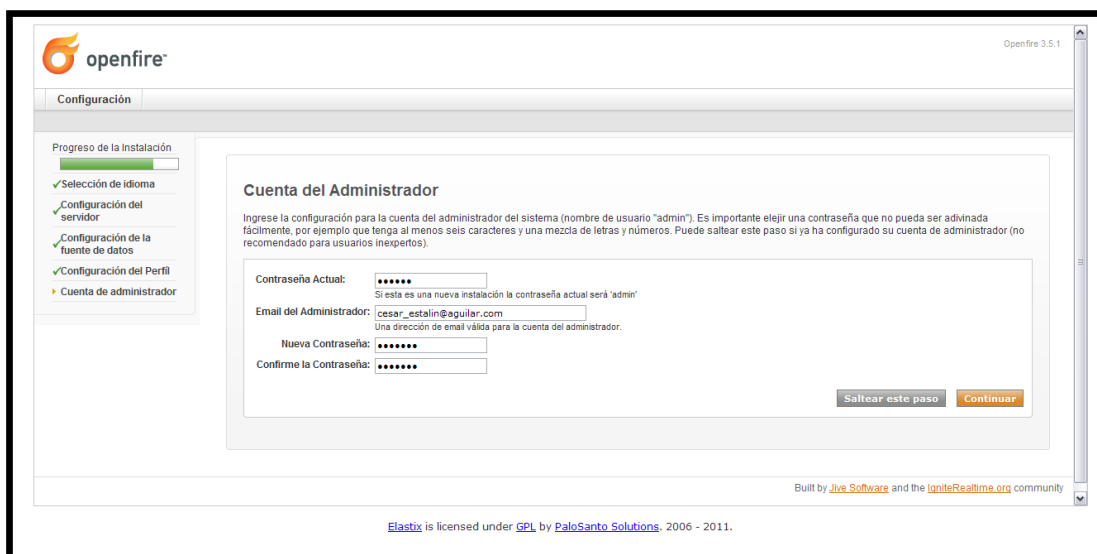


Figura 4-8 Configuración de la Cuenta del Administrador

A continuación nos saldrá el mensaje de que la configuración esta completa y nos da un link para conectarnos a la consola de administración, procedemos a dar clic sobre el link Conéctese a la consola de administración

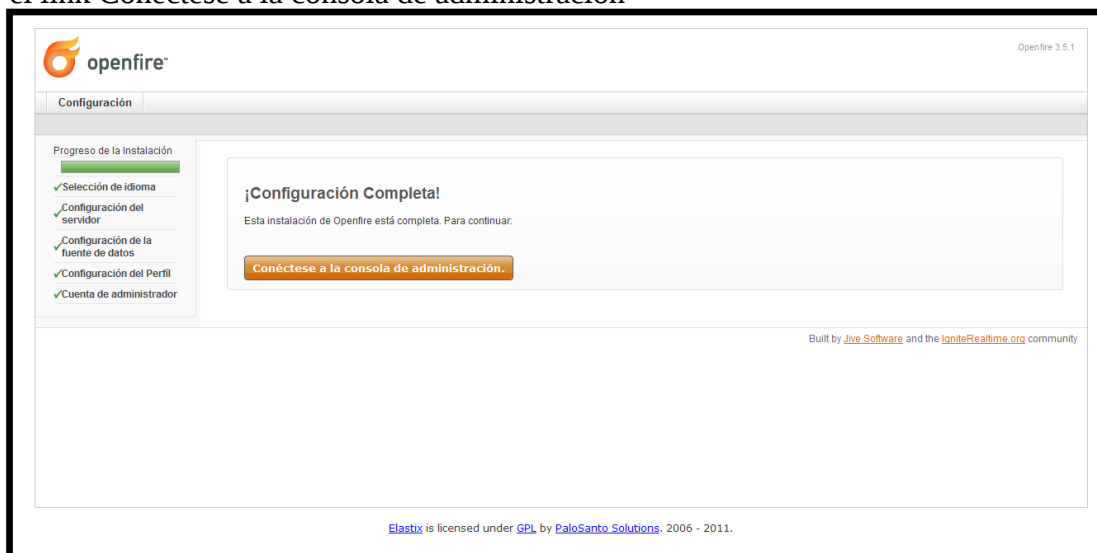
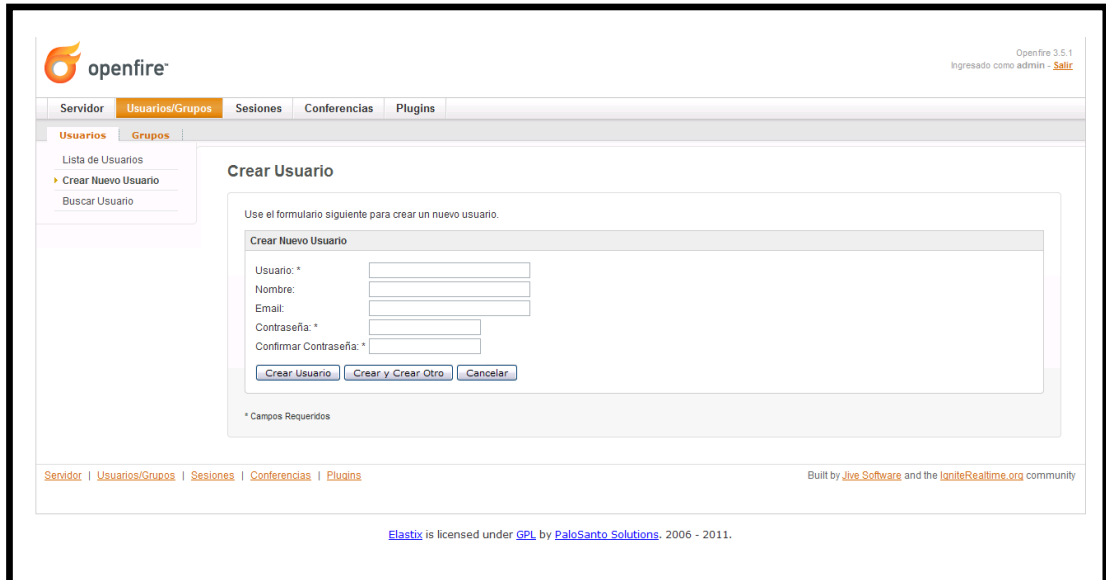


Figura 4-9 Configuración Finalizada

7. A continuación crearemos los usuarios que tendrán este servicio para lo cual nos ubicamos en el menú Usuario
8. o/Grupos y después seleccionamos la opción de crear nuevo usuario que se encuentra en la parte izquierda; los datos que necesitaremos ingresar son:

- **Usuario:** donde va un Nick cualquiera
- **Nombre:** va el nombre del usuario
- **Email:** el mail de usuario
- **Contraseña:** contraseña



The screenshot shows the OpenFire 3.5.1 web interface. The top navigation bar includes 'Servidor', 'Usuarios/Grupos', 'Sesiones', 'Conferencias', and 'Plugins'. The 'Usuarios/Grupos' section is active, showing a sidebar with 'Lista de Usuarios', 'Crear Nuevo Usuario', and 'Buscar Usuario'. The main content area is titled 'Crear Usuario' and contains a form with the following fields: 'Usuario: *', 'Nombre:', 'Email:', 'Contraseña: *', and 'Confirmar Contraseña: *'. Below the form are three buttons: 'Crear Usuario', 'Crear y Crear Otro', and 'Cancelar'. A note at the bottom of the form states '* Campos Requeridos'.

Figura 4-10 Configuración de Usuarios

Ya tenemos creado los usuarios y configurado el OpenFire, lo siguiente es instalar spark instant messenger el cual nos servirá para poder hacer uso de nuestro sistema de mensajería instantánea; para la instalación de este software debemos seguir estos pasos:

1. Damos doble clic sobre el icono de software spark

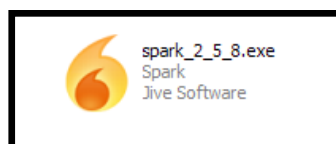


Figura 4-11 Icono de Software Spark

2. En la pantalla que nos sale ponemos ejecutar

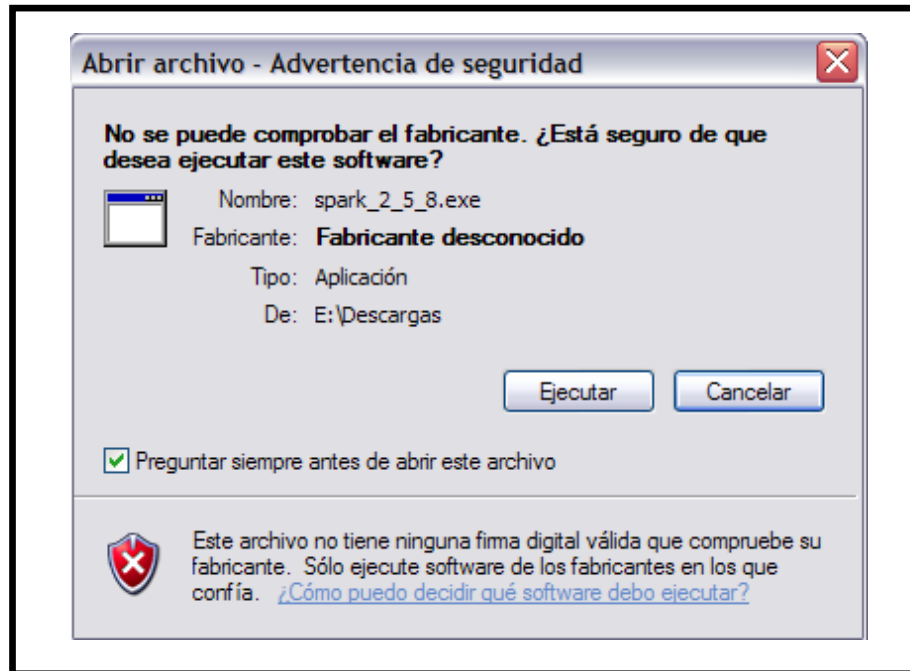


Figura 4-12 Ejecutar el programa

3. Damos clic en Next

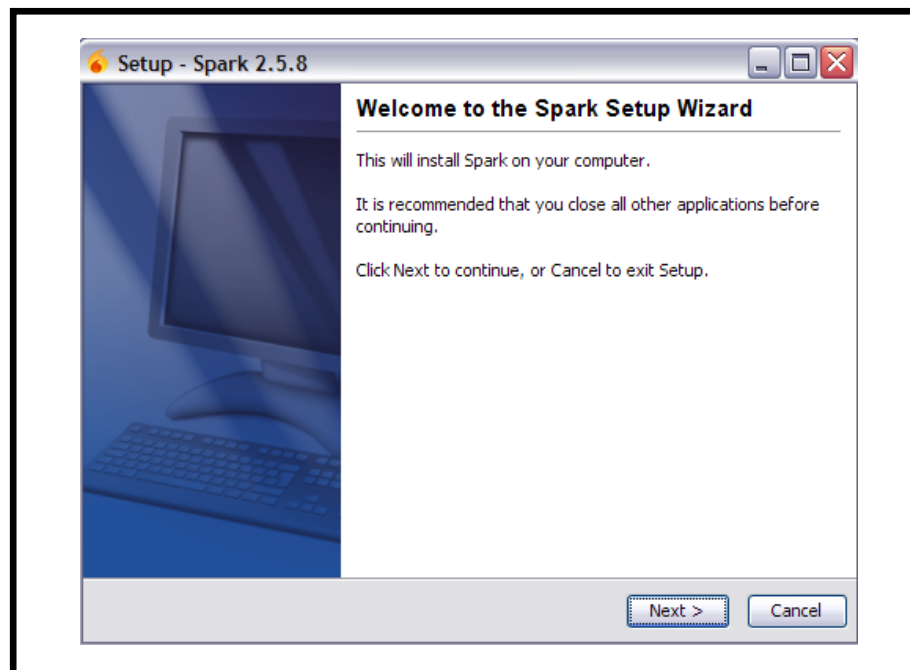


Figura 4-13 Configuración de Instalación de Spark 2.5.8

4. Seleccionamos el Directorio donde queremos que se guarde el programa y damos clic en Next

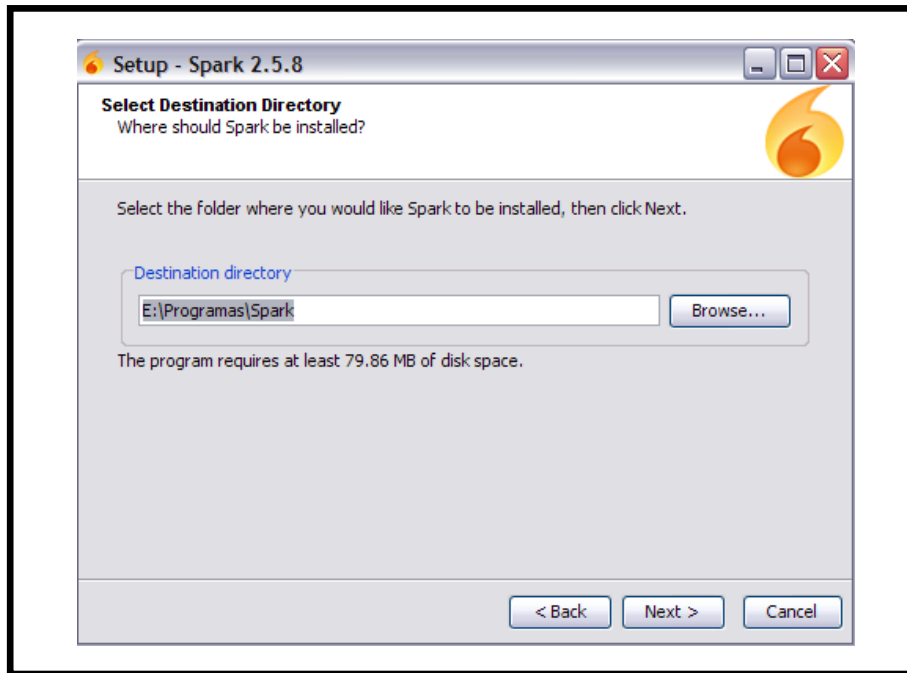


Figura 4-14 Configuración del Directorio para la instalación

5. Selección del nombre que tendrá la carpeta dentro del Menú principal:
para lo cual le dejamos como esta y damos clic en Next

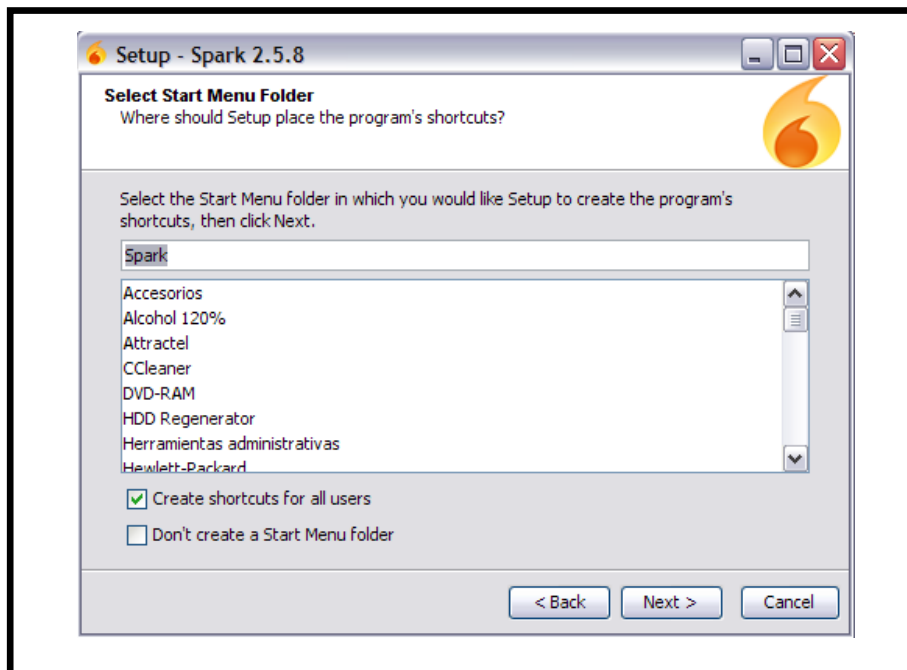


Figura 4-15 Configuración del Nombre en el Menú Principal

6. Aquí nos pide si queremos crear un acceso directo en el escritorio y si deseamos crear un acceso directo en la barra de acceso rápido, marcamos la que deseemos, y damos clic en Next

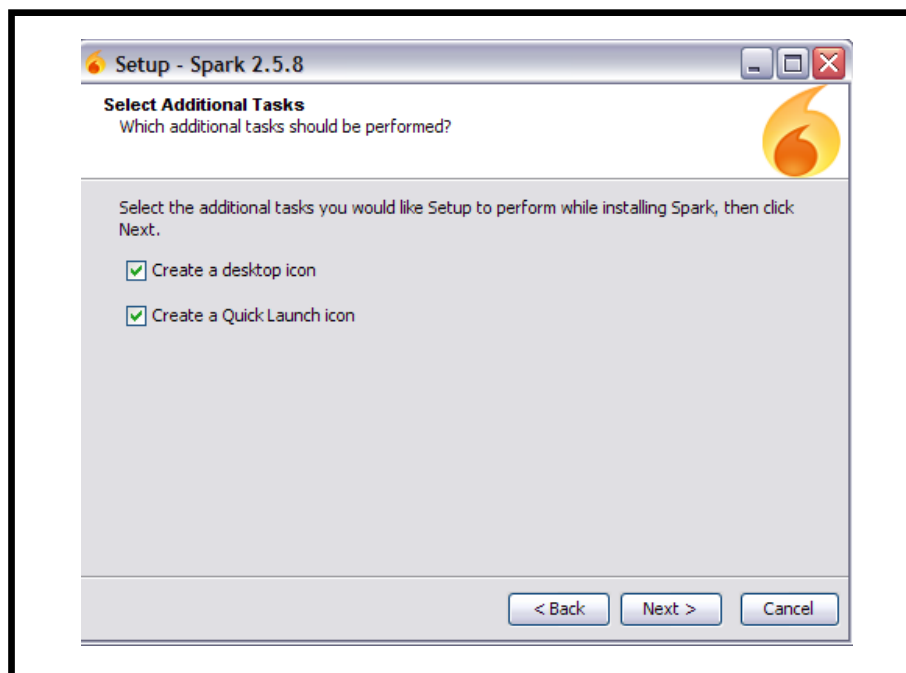


Figura 4-16 Configuración de Creación de iconos

7. Esperemos que se instale, y damos clic en finalizar

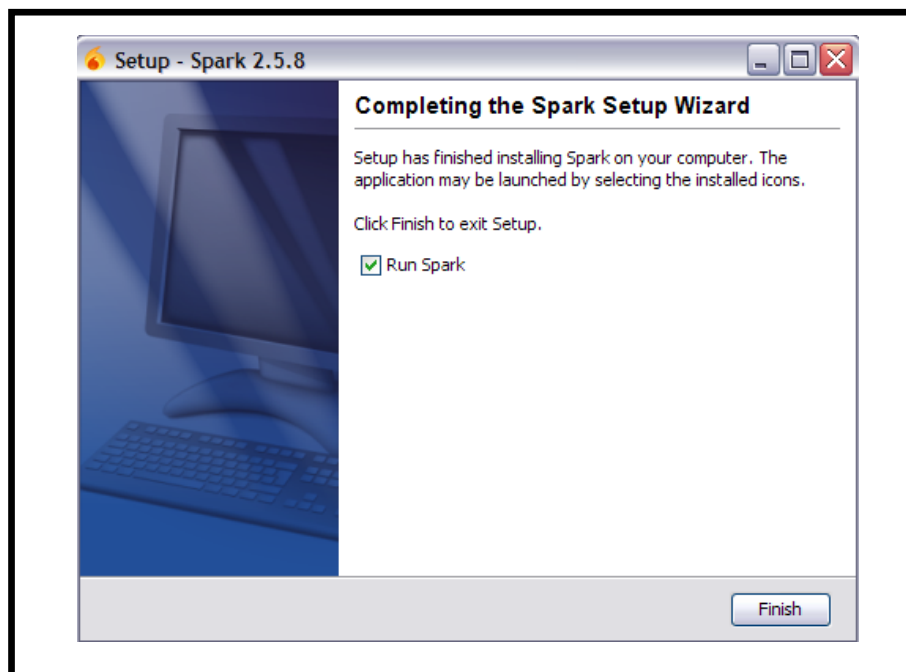


Figura 4-17 Finalización de la Instalación de Spark 2.5.8

Se debe configurar para que pueda acceder a nuestro servidor OpenFire, para lo cual necesitamos llenar los campos donde dice nombre de usuario, cualquiera de los usuarios antes creados, la contraseña del usuario y en donde dice servidor tenemos

que poner la dirección IP de nuestro servidor y podemos enviar mensajes instantáneos a todos nuestros contactos.

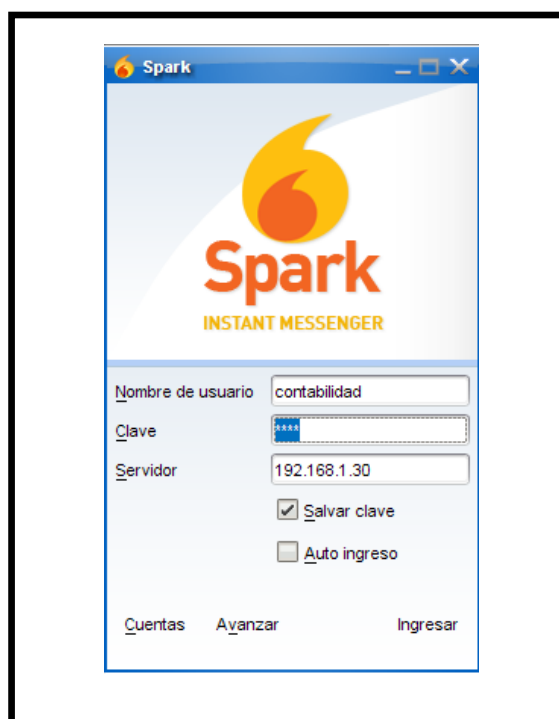


Figura 4-18 Spark 2.5.8

4.2.2 Configuración de FAX (HYLAFAX)

Lo primero que debemos realizar es la configuración de la extensión IAX2, eso yo lo realizamos en la sección anterior de la configuración de extensiones, a continuación describiremos los pasos a seguir para configurar el fax virtual:

Nos vamos a la pestaña FAX y escogemos la opción de Nuevo Fax Virtual y tenemos que configurar con los siguientes datos:

Configuración de Fax Virtual	
Nombre de Fax Virtual	Un nombre para identificar
Email Destino	Dirección de Correo donde llegar el Fax
Nombre Caller Id	Nombre para el identificador de llamadas
Número Caller Id	Número para el identificador de llamadas
Extensión de Fax (IAX)	El mismo número creado en la extensión IAX
Secreto(IAX)	El mismo secret con el que creamos la extensión IAX

Tabla 4-3 Configuración de Fax Virtual

La configuración quedaría de esta forma:

Figura 4-19 Configuración En Elastix del Fax Virtual

Procederemos a configurar Winprint Hylafax para Windows el cual nos va a servir para poder enviar los fax, el archivo para la instalación viene envasado en forma de un instalador de Windows, que instala los sistemas necesarios de los archivos .DLL y registra el monitor con el sistema. Este incluye soporte para un nuevo tipo de puerto en su sistema, llamado "WinPrintHylaFAX puerto". Una vez que haya instalado correctamente los archivos, para una correcta configuración tenemos que seguir estos pasos:

1. Necesitamos crear una nueva impresora para estos nos vamos a Inicio ->panel de Control -> Impresoras y Faxes y escogemos la opción "Agregar impresora".

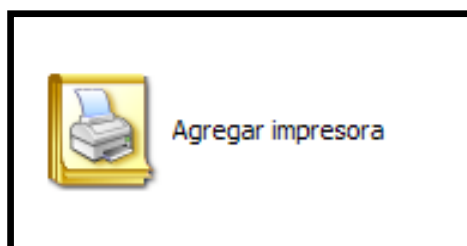


Figura 4-20 Icono Agregar Impresora

2. En la pantalla que tenemos en la Figura 4-21 solo hacemos clic en siguiente para poder comenzar a la configuración de nuestra Impresora:

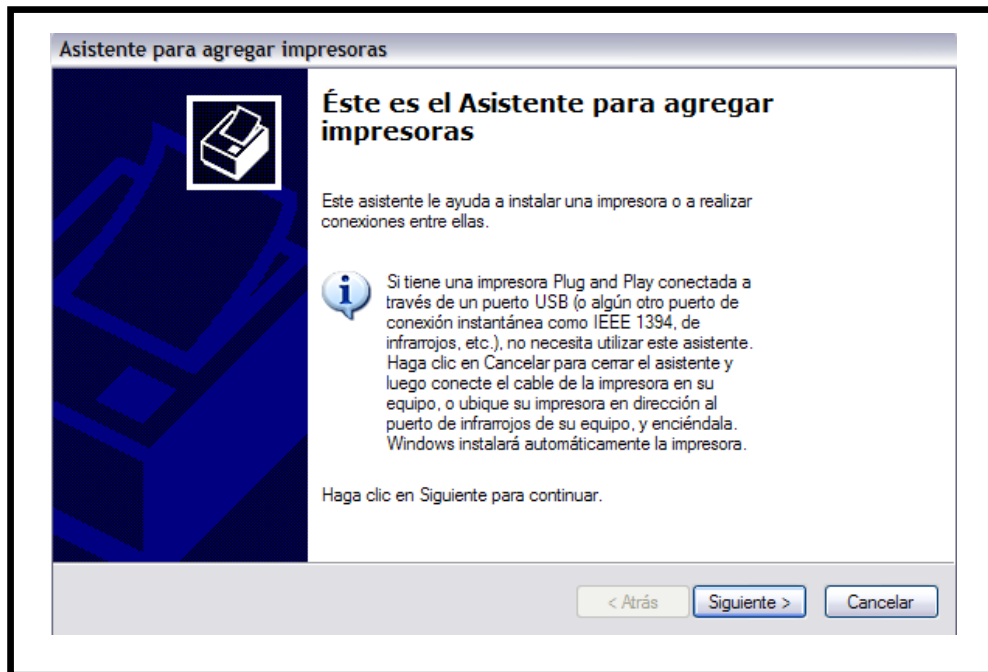


Figura 4-21 Asistente para Agregar Impresora

3. Procedemos a seleccionar una "Impresora local conectada a este equipo", ya que la impresora que estamos configurando no es una impresora física, desmarcamos la opción "Detectar e instalar mi impresora Plug and Play automáticamente" y le damos clic en siguiente.

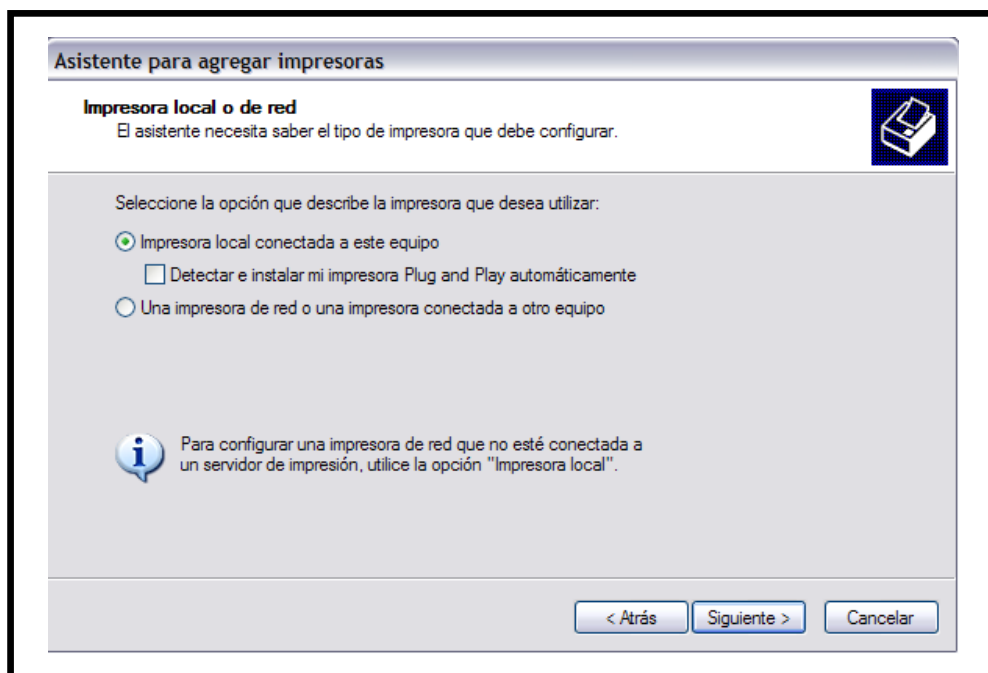


Figura 4-22 Descripción de la Impresora a Utilizar

4. A continuación, se pedirá qué puerto se va a utilizar. Seleccionamos la opción de "crear un nuevo tipo de puerto." Si la instalación se ha realizado correctamente, debería aparecer "WinprintHylafax" como una de las opciones. (Si no aparece en la lista, algo salió mal con la instalación.)

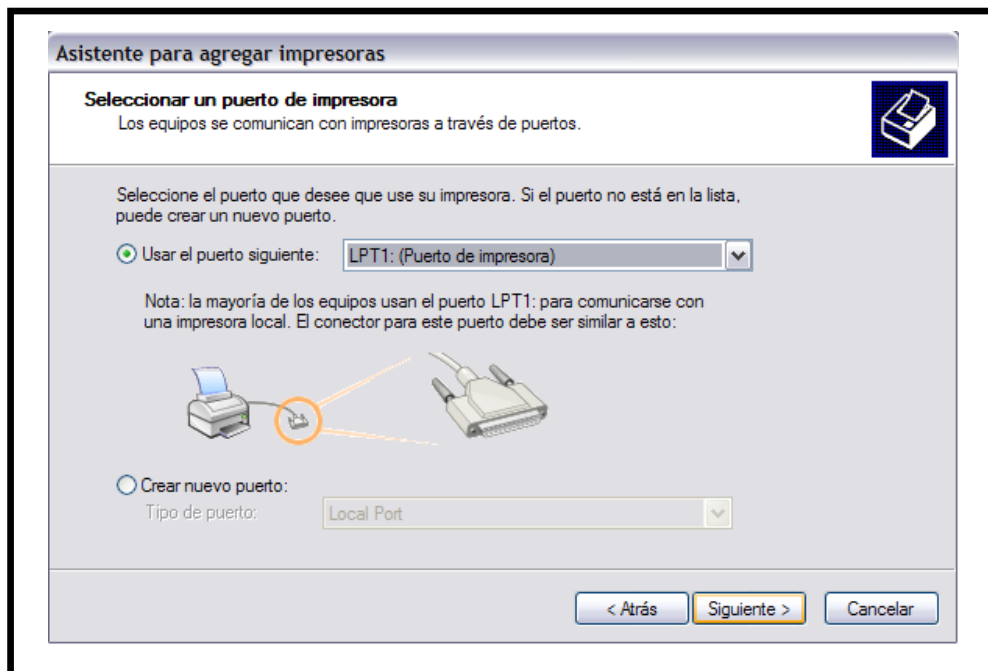


Figura 4-23 Configuración de Puerto de la Impresora

5. A continuación, un cuadro emergente para que le demos el nombre del puerto. El nombre en realidad no importa, pero dándole el nombre HFAX1: le ayudará a identificarlo.

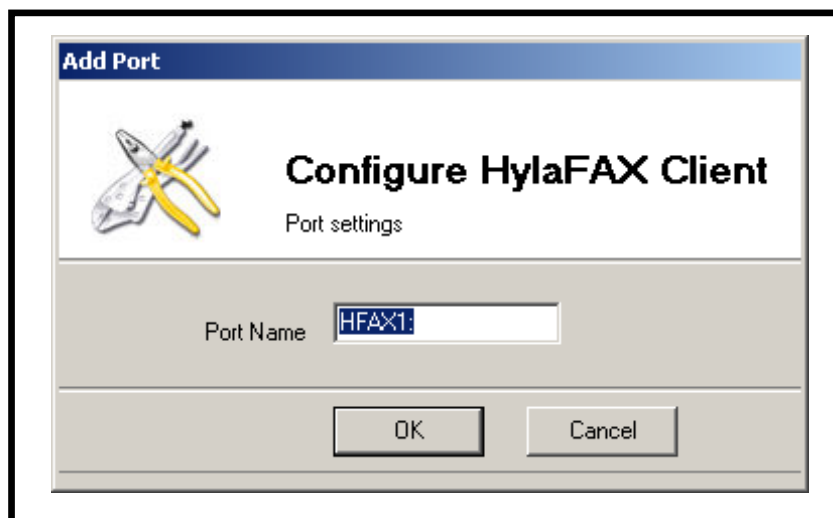


Figura 4-24 Nombre del Puerto

6. A continuación, se debe de seleccionar el tipo de impresora. La mejor opción es la Apple LaserWriter 12/640 PS; y le damos clic en siguiente.

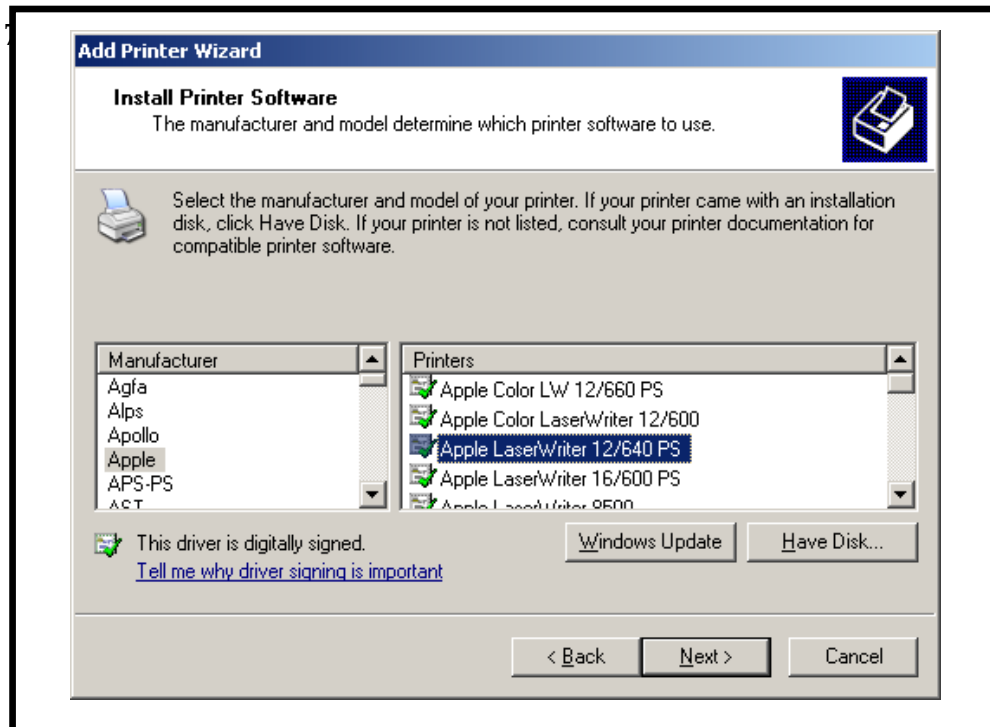


Figura 4-25 Configuración del Modelo de la impresora

8. A continuación tenemos que dar un nombre que lo identifique, ponemos en “NO” en la parte para que esta no sea nuestra impresora por defecto y le damos clic en siguiente:

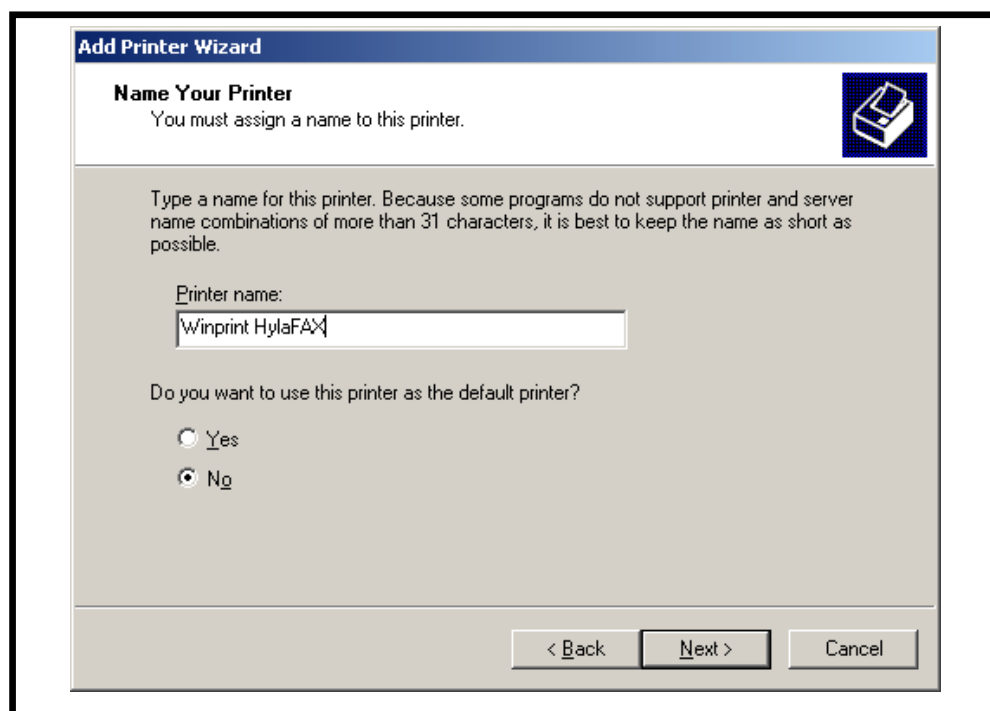


Figura 4-26 Nombre de la impresora

9. Nos va a pedir si queremos imprimir una página de prueba ponemos “NO”.
10. Haga clic en la impresora para llamar a sus propiedades, y seleccione la pestaña "Puertos". Localice el puerto que acaba de crear.

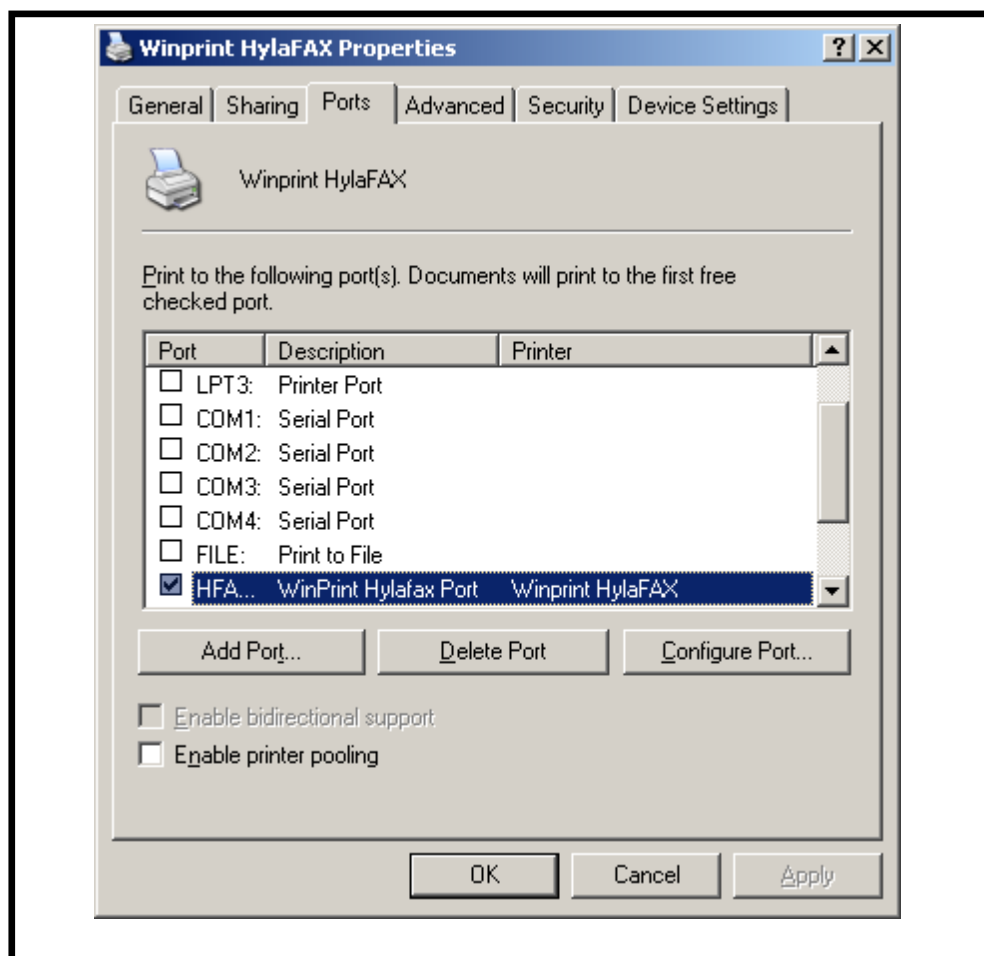


Figura 4-27 Configuración de Puerto de Impresora

10. Pulse el botón "Configurar puerto" para configurar la conexión a su servidor HylaFax.

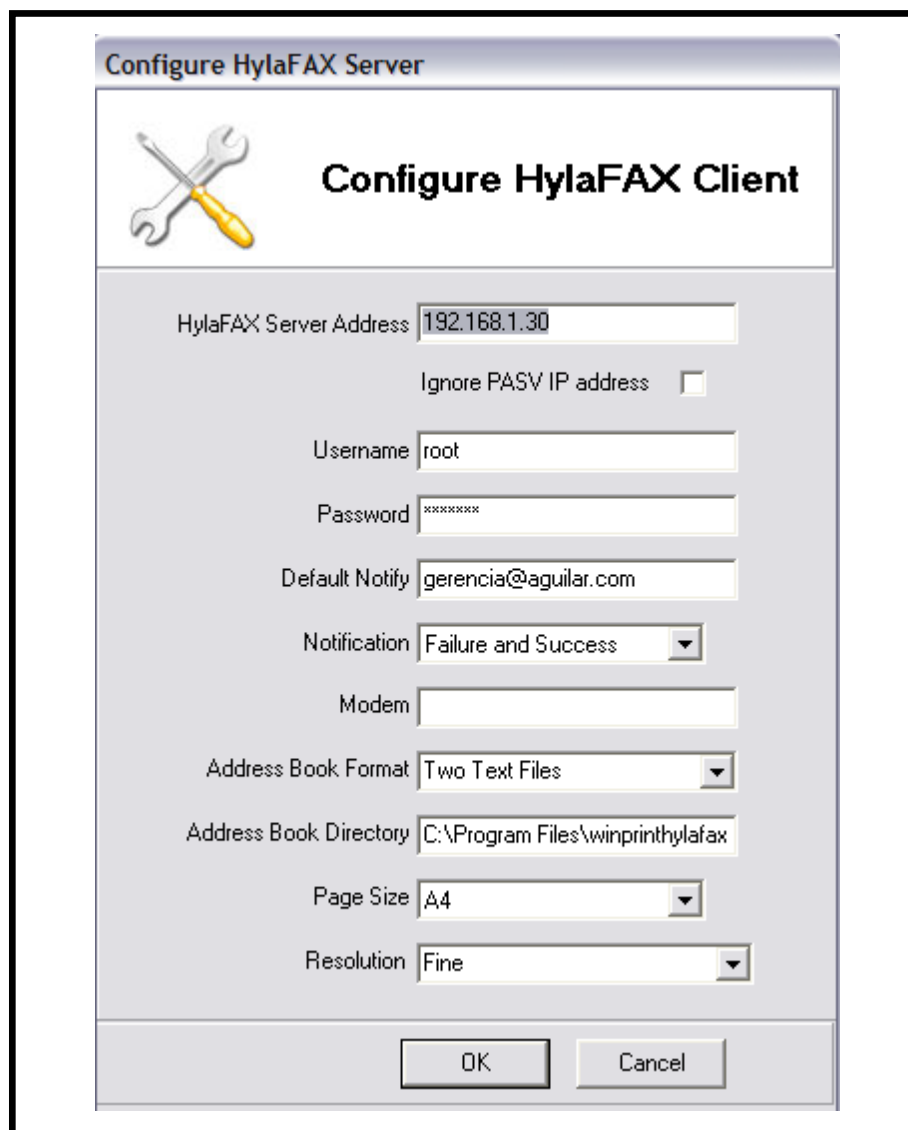


Figura 4-28 Configuración de Cliente HylaFax

La mayoría de las personas que tienen problemas son debido a que su servidor HylaFax no les permite conectarse. Hay que modificar el servidor HylaFax, y asegúrese de que ha configurado los permisos correctos, que puede ser tan simple como poner la dirección IP de su cliente en el archivo `<hosts.hfaxd>` ubicado en el path `/var/spool/hylafax/etc`. Puede que no necesite el nombre de usuario o contraseña, en cuyo caso, puede dejar contraseña en blanco, pero el nombre de usuario es siempre necesaria, incluso si no está autenticado; el archivo `<hosts.hfaxd>` nos quedara de la siguiente manera:

localhost

127.0.0.1

192.168.1.30; dirección IP del servidor Elastix

192.168.1.15; dirección IP de la PC1

Al momento de proceder a enviar un fax se lo realizara de la siguiente manera:

Desde cualquier editor de texto, enviamos a imprimir el archivo y seleccionamos la impresora que hemos creado “WinPrintHylafax” y nos saldrá la Figura 4-29 en donde colocamos el número de fax a enviar:

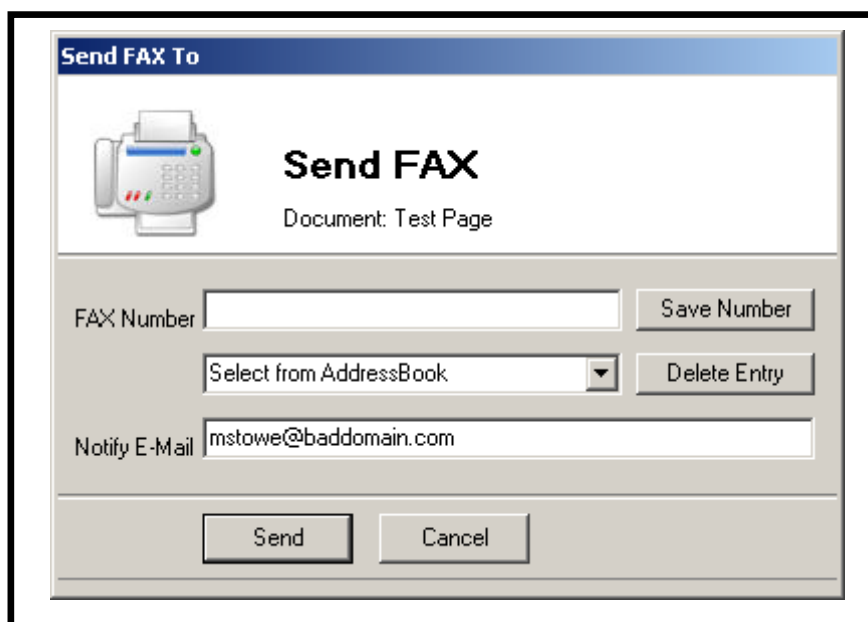


Figura 4-29 Enviar Fax

El número de fax es totalmente inédito y no verificado. Se basa en reglas de marcado HylaFax. Como era de esperar, el servidor HylaFax enviará una entrega de correo electrónico de confirmación (o no hacerlo) a la dirección de correo electrónico especificada.

4.2.3 Configuración de CallCenter

Para una mejor comprensión del módulo del CallCenter se explicarán los conceptos básicos de este modulo, como son las colas, agentes, formularios y lo que son las campañas entrantes y salientes.

Colas: Se define como un grupo de usuarios para los cuales se define ciertas políticas; permite que un ilimitado número de llamantes puedan permanecer en espera hasta que un representante o recurso esté disponible para dar asistencia.

Agentes: Son los usuarios dinámicos conectados a la cola, es decir que pueden ingresar a la cola desde cualquier extensión

Formularios: son creados para poder recolectar información al momento de ejecutar una campaña y realizar llamadas desde la consola de agente.

Campañas Entrantes: Cuando el usuario realiza llamadas a la central telefónica, se lo envía a una cola hasta que un recurso este habilitado.

Campañas Salientes: es información de cuándo y cómo se va a generar una serie de llamadas de manera automática a números telefónicos que se deben subir en un archivo CSV; esta información también permite asociar una llamada con una respuesta a un formulario asociado a la campaña.

Los pasos a seguir para poder configurar el CallCenter son:

1. Crear una troncal
2. Crear ruta de salida
3. Configurar extensiones
4. Crear cola
5. Crear agente
6. Crear formulario
7. Configurar módulo de CallCenter
8. Crear campañas
9. Ingresar al Agent Console

La troncal ya la tenemos configurada en la sección anterior al igual que las extensiones y la ruta de salida lo siguiente a configurar son las colas (queue) para lo cual seguimos estos pasos:

1. Ir a la pestaña PBX
2. En el menú izquierdo escogemos “Colas” y procedemos a crear una cola con los datos de la tabla 4-3:

Cola (Queue)	
Número de Cola	1234
Nombre de Cola	COLAOUTPUT
Agentes fijos	A4001,0
MaxCallers	1
Ringstrategy	Sonar todos
Tiempo de espera de Agente	15
Reintentar	5

Tabla 4-3 Configuración de Cola

Nos quedaría de la siguiente manera:

Añadir cola

Añadir cola

Número de cola:

Nombre de la cola:

Contraseña de la cola:

Prefijo del nombre del CID:

Prefijo de tiempo de espera:

Información de alerta:

Agentes fijos:

A4010,0

Captura rápida de extensión

Figura 4-30 Configuración de Cola

Opciones de la cola

Anuncio de agente:

Anuncio de entrada:

Clase de música en espera:

Hacer sonar en lugar de música en espera: ☐

Tiempo máximo de espera:

Llamantes máximos:

Entrar si vacía:

Salir cuando vacía:

Ring Strategy:

Tiempo de espera de agente:

Reintentar:

Wrap-Up-Time:

Grabación de llamadas:

Event When Called:

Estado:

Skip Busy Agents:

Queue Weight:

Autofill: ☐

Agent Regex Filter:

Report Hold Time:

Service Level:

Figura 4-31 Configuración de Opciones de Cola

Guardamos y aplicamos los cambios.

El siguiente punto a configurar son los agentes y tenemos que seguir los pasos:

1. Ir a la pestaña “CallCenter”.
2. Luego al menú llamado “Agents”.
3. Después damos clic en el botón “Nuevo Agente” y procedemos a crear con la configuración de la tabla 4-4:

Agente (Agent)	
Número	4010
Nombre	A4010
Contraseña	1234
Repetir Contraseña	1234

Tabla 4-4 Configuración del Agente

4. Nos quedaría de la siguiente manera:

Figura 4-32 Configuración de Agente

5. Guardamos y Aplicamos los cambios

El siguiente punto a configurar es el formulario que utilizaremos para realizarle la encuesta al cliente, para lo cual tenemos seguir estos pasos:

1. Ir a la pestaña “Call Center”.
2. Luego al menú llamado “Formularios”.
3. Procedemos a dar clic en el botón nuevo formulario y crearemos un formulario básico de recolección de datos con las preguntas de la tabla 4-5:

Formulario		
Nombre del Campo	Tipo	Orden
Nombre del Cliente	Typetext	1
Dirección del Cliente	Typetext	2
Producto Interesado	Typetext	3

Tabla 4-5 Configuración del Formulario

El diseño quedaría de la siguiente forma en modo de diseño, luego daremos clic en el botón aplicar cambios:

Figura 4-33 Crear Formulario

En modo de visualización seria de esta forma:

Figura 4-34 Visualización del Formulario

El siguiente punto a configurar es el módulo de CallCenter, la configuración del CallCenter se la realiza tanto en la interfaz web como en los archivos de configuración de Elastix, lo cual se debe realizar antes de crear las campañas, seguiremos los pasos siguientes para la configuración:

1. Ir a la pestaña “CallCenter”.
2. Luego al menú llamado “Configuración”.
3. Procedemos a configurar con los datos de la tabla 4-6:

Call Center	
Login de Asterisk	Root
Password de Asterisk	Gogoman
Password de Asterisk(confirmación)	Gogoman
Activar debug de marcador	Marcar para activar

Tabla 4-6 Configuración de Call Center

Nos quedaría de la siguiente forma:

Figura 4-35 Visualización de Configuración CallCenter

Luego damos clic en el botón guardar, así mismo en estado del dialer hacemos clic en iniciar para activar el servicio del marcador progresivo.

Ahora tenemos que configurar la cola llamante que usaremos para las campañas, para lo cual configuraremos los archivos de Asterisk, en el archivo de *<extensión _custom.conf>*, que se encuentra en el path **/etc/asterisk** de nuestro servidor Elastix.

En el Shell de nuestro servidor Elastix con un editor de texto ingresamos el siguiente comando:

```
vi /etc/asterisk/extensión s_custom.conf
```

Vamos a encontrar esto :

```
[from-internal-custom]
```

```
exten => 1234,1,Playback(demo-congrats) ; extensión s can dial 1234
```

```
exten => 1234,2,Hangup()
```

```
exten => h,1,Hangup()
```

```
include => agentlogin
```

```
include => conferences
```

```
include => calendar-event
```

```
include => weather-wakeup
```

Tendremos que cambiarlo, y debería quedar así

```
[from-internal-custom]
```

```
;exten => 1234,1,Playback(demo-congrats) ; extensión s can dial 1234
```

```
exten => 1234,1,Queue(1234) : siendo 1234 el nombre de la cola saliente
```

```
exten => 1234,2,Hangup()
```

```
exten => h,1,Hangup()
```

include =>agentlogin
include =>conferences
include => calendar-event
include =>weather-wakeup

Guardamos y salimos del archivo, lo que hemos hecho es comentar la linea donde hace un playback del archivo demo-congrats, por la línea donde ingresa a la cola llamada 1234, previamente creada.

La siguiente Configuración es crear la campaña que vamos a utilizar para lo cual primeramente debemos de preparar nuestro archivo de teléfonos, el marcador progresivo necesita un archivo de llamadas en formato cvs, por ejemplo vamos a usar un archivo llamado "base-prueba.csv" con el siguiente contenido

Número, Nombre, Dirección

909231671,"Cliente1","Rimac"
94832730,"cliente2","Callao"

En este caso estamos usando el prefijo 1+número, ya que en la troncal, se encuentra el prefijo "9"

Ir a la pestaña "CallCenter" y luego al menú llamado "Llamadas salientes" , procederemos a hacer clic en el botón "Crear nueva campaña" ,procederemos a crear una campana básica :

Nombre : CP1

Range Date : 16/04/2009 Start 16/04/2009 End

Schedule per Day : 01:00 Start time 23:00 End Time

Formulario : FORMULARIO1 (previamente creador en el paso 6)

Troncal : SIP/JUSVOIP (Sacamos todas las llamadas por nuestra cuenta VOIP)

Intentos : 2 (número de intentos que Elastix intentara que un determinado número le responda)

Cola : 1234 COLAOUTPUT (Cola de agentes previamente creada)

Archivo de llamas : cargar el archivo(nuestro archivo de llamadas en formato cvs)

Script : Bienvenido al la campana de prueba (un pequeño dialogo el cual deseamos que los agentes digan cuando hablan con un cliente)

Quedaría de la siguiente forma:

Figura 4-36 Creación de Campaña

Luego damos clic en guardar y la campaña pasara al estado activo.

Nombre	Fecha Inicio	Hora Inicio	Fecha Fin	Hora Fin	Intervalo	Temática	Código	Límites computados	Presencia	Estado	Operaciones
CPI	2006-04-16	01:00:00	2006-04-16	20:00:00	5	SP-Integ	008			Activo	Ver Datos

Figura 4-37 Listado de Campañas

CAPITULO 5: PLAN DE MARCADO AVANZADO

5.1 Concepto

El plan de marcado avanzado lo podemos ligar con lo que es el Asterisk Gateway Interface (AGI) que sirve para la integración con programas externos en diferentes lenguajes de programación como:

C, C#, Bourne Shell, PHP, Ruby, Python, Perl, Java

El programa debe cumplir:

- Ser ejecutable (mediante un intérprete es válido).
- Localizado por defecto en /var/lib/asterisk/agi-bin
- Asociado con alguna extensión en el Dial Plan:

exten => 200,1,AGI(mi_programa.php|argumentos)

Comunicación con Asterisk:

- El programa escribe en la salida estándar para enviar un comando a Asterisk.
- El programa lee de entrada estándar para obtener una respuesta de Asterisk.

5.2 Commandos AGI

ANSWER

AUTOHANGUP <time>

CHANNEL STATUS [<channelname>]

EXEC <application><options>

GET DATA <filename> [<timeout>] [<max digits>]

GET VARIABLE <variablename>

HANGUP [<channelname>]

RECEIVE CHAR <timeout>

RECORD FILE <filename><format><escape digits>

<timeout> [BEEP]

SAY DIGITS <digit string><escape digits>

SAY NUMBER <number><escape digits>

SEND IMAGE <image>

SEND TEXT "<text to send>"

SET CALLERID <number>

SET CONTEXT <desired context>

SET EXTENSIÓN <new extensión >

SET PRIORITY <new priority number>
SET VARIABLE <variablename><value>
STREAM FILE <filename><escape digits>
TDD MODE <on|off>
VERBOSE <level>
WAIT FOR DIGIT <timeout>

5.3 Configuración

Para poder utilizar lo que es el plan de marcado avanzado en java necesitaremos realizar las siguientes configuraciones:

- a. Necesitamos crear el archivo arrancar.sh y lo ubicamos en el path /var/lib/Asterisk/agi-bin que nos quedaría de la siguiente manera:

```
#!/bin/bash
java -cp Asterisk-java.jar:. net.sf.asterisk.fastagi.DefaultAGIServer
```
- b. Necesitamos insertar la librería AGI (Asterisk-java.jar), en el path /var/lib/asterisk/agi-bin/.
- c. Instalar java, lo cual vamos a proceder a copiar la carpeta jdk1.6 donde nosotros queramos, yo procederé a ponerle en el path/usr/local/java/conectores/
- d. Cambiar el archivo profile que se encuentra en el path /etc/profile, aquí tenemos que agregar las variables de entorno para que funcione java, al final agregamos:

```
JAVA_HOME=/usr/local/java/Conectores/jdk1.6.0
PATH=$JAVA_HOME/bin/:$PATH
CLASSPATH=/usr/local/java/Conectores/Asterisk-
java.jar:/usr/local/java/Conectores/servlet.jar:/usr/local/java/Co
nectores/mysql-connector-java-5.1.12.jar
export JAVA_HOME PATH CLASSPATH
```

Debemos reiniciar el server para que cambie el entorno de variables

- d. Instalar phpMyAdmin con el comando yum -y install phpMyAdmin

- e. Tenemos que dar permisos en el archivo phpMyAdmin.conf que se encuentra en el path /etc/httpd/conf.d/phpMyAdmin.conf el comando a utilizar es:

```
vi /etc/httpd/conf.d/phpMyAdmin.conf
```

primero allow from que está en 127.0.0.1 le cambiamos por allow from all

```
<Directory /usr/share/phpMyAdmin/>
```

```
Order deny, allow
```

```
deny from all
```

```
allow from all
```

```
allow from ::1
```

```
</Directory>
```

Después cambiamos allow from none por allow from all

```
<Directory /usr/share/phpMyAdmin/libraries>
```

```
Order Deny, Allow
```

```
Deny from All
```

```
Allow from All
```

```
</Directory>
```

Ya tenemos configurado y nos podremos conectar de forma remota a la base de datos.

- f. Procedemos a escribir y compilar el programa que vamos a utilizar.
- g. Tenemos que agregar un fichero de propiedades llamado FastAGI-mapping.properties, que debe estar en la ruta cd /var/lib/asterisk/agi-bin, en donde pondremos el nombre de el archivo java y también colocaremos el mismo archivo pero con extensión agi nos quedaría de la siguiente manera.

Ejemplo.agi=Ejemplo

- h. Cambiar permisos y usuarios del archivo java y de FastAgi-mapping.properties

Chmod 775 nombre del archivo.java FastAgi-mapping.properties

Chown Asterisk:Asterisk nombre del archivo.java FastAgi-mapping.properties

- i. Para que corra tiene que estar dentro de algún contexto con la siguiente instrucción AGI(agi:localhost://el nombre del archivo.agi)

5.4 Ejemplo

El archivo agi que utilizaremos para conectarnos a la base de datos y extraer los valores de cuanto adeuda el cliente quedaría de la siguiente manera:

```
//IMPORTAR LAS CLASES QUE VAMOS A UTILIZAR
import net.sf.asterisk.fastagi.AGIRequest;
import net.sf.asterisk.fastagi.AGIChannel;
import net.sf.asterisk.fastagi.AGIException;
import net.sf.asterisk.fastagi.AbstractAGIScript;
import java.io.*;
import java.sql.*;
import java.text.*;

//CREACION DE LA CLASE PRINCIPAL
public class Credito extends AbstractAGIScript
{
    public void service(AGIRequest petition, AGIChannel canal)
        throws AGIException
    {
        Connection conexion;
        Statement instruccion;
        answer(canal);

// DAR EL MENSAJE DE INSERTAR CEDULA
        String cedula = getData(canal, "custom/Hola");
        sayDigits(canal, cedula);
        double cuota = 0.0;
        double saldo = 0.0;
        double pago = 0.0;
        java.util.Date fecha = new java.util.Date();
        int cuotas = 0;
        try
```

```

{
    Class.forName("com.mysql.jdbc.Driver");
    //COONECTAR A LA BASE DE DATOS
    conexion=DriverManager.getConnection("jdbc:mysql://localhost:3306/gym"
    ,"root","gogoman");

    instruccion=conexion.createStatement();
    // SQL QUE SACARA LOS DATOS
    Stringcadenasql="Selectcredito.fecha_pago          fecha,
Count(credito.total)  cuotas,  SUM(credito.total)  saldo,  credito.total  pago
fromcredito,factura_cabecerawherefactura_cabecera.num_fac          =
credito.codigo_factura  and  credito.pagado=0  and  factura_cabecera.cedula
="+cedula + "groupbycredito.total";

    ResultSet tabla=instruccion.executeQuery(cadenasql);
    tabla.next();

    cuotas =tabla.getInt("cuota");
    fecha = tabla.getDate("fecha");
    pago = tabla.getDouble("pago");
    saldo = tabla.getDouble("saldo");
}catch(Exception e){
    System.out.println(e);
}

// DAR MENSAJE DE EL VALOR DE LA CUOTA
String pago2=String.valueOf(pago);
streamFile(canal,"custom/saldo");
sayNumber(canal,pago2);
streamFile(canal,"dollar");

// DAR MENSAJE DE EL VALOR DE LA FECHA DE PAGO
String fecha2=String.valueOf(fecha);
// SEPARAR FECHA
streamFile(canal,"custom/saldo");
sayNumber(canal,fecha2);

```

```

        // DAR MENSAJE DE EL VALOR DE CUOTAS Q DEBE
        String cuotas2=String.valueOf(cuotas);
        streamFile(canal,"custom/saldo");
        sayNumber(canal,cuotas2);

        // DAR MENSAJE DE EL TOTAL Q DEBE
        String saldo2=String.valueOf(saldo);
        streamFile(canal,"custom/saldo");
        sayNumber(canal,saldo2);
        streamFile(canal,"dollar");
        hangup(canal);
    }
}

```

EL FastAGI-mapping.properties nos quedaría de la siguiente manera:

Credito.java=Credito.agi

La instrucción para poder correr el agi que lo debemos incluir en el ivr seria el siguiente:

exten =>2030,n,AGIi(agi:/192.168.1.30/credito.agi)

CAPITULO 6: PRUEBAS

6.1 Prueba de llamada entre extensiones

En esta sección verificaremos que se puede efectuar llamadas entre las diferentes extensiones. Además visualizaremos que ocurre en la línea de comando de Asterisk de cada PBX al momento de realizar estas llamadas. Para verificar que todas las conexiones estén bien lo primero que debemos hacer es entrar a la consola de Asterisk con el comando `asterisk -r` y seguidamente ingresamos el comando `sip show peers`, el cual nos va a dar todas las extensiones SIP que están conectadas.

```
asterisk*CLI> sip show peers
Name/username      Host                Dyn Nat ACL Port    Status
1010/1010          168.0.0.41         D  N  A  5060    OK (1 ms)
2010/2010          168.0.0.116        D  N  A  5060    OK (1 ms)
3010/3010          168.0.0.207        D  N  A  5060    OK (1 ms)
4010/4010          168.0.1.182        D  N  A  5060    OK (3 ms)
4 sip peers [Monitored: 4 online, 0 offline Unmonitored: 0 online, 0 offline]
```

Figura 6-1 muestra las extensiones SIP conectadas

A continuación vamos a realizar una llamada entre la extensión 3010 que corresponde al departamento de sistemas y la extensión 1010 que pertenece al departamento de gerencia y procederemos a ver lo que ocurre en la consola de Asterisk.



Figura 6-2 extensión 3010 realizando llamada a extensión 1010



Figura 6-3 Extensión 1010 Recibiendo llamada de Extensión 3010

```
-- Executing [1010@from-internal:1] Macro("SIP/3010-0000001d", "exten-vm,1010,1010") in new stack
-- Executing [s@macro-exten-vm:1] Macro("SIP/3010-0000001d", "user-callerid,") in new stack
-- Executing [s@macro-user-callerid:1] Set("SIP/3010-0000001d", "AMPUSER=3010") in new stack
-- Executing [s@macro-user-callerid:2] GotoIf("SIP/3010-0000001d", "0?report") in new stack
-- Executing [s@macro-user-callerid:3] ExecIf("SIP/3010-0000001d", "1?Set(REALCALLERIDNUM=3010)") in new stack
-- Executing [s@macro-user-callerid:4] Set("SIP/3010-0000001d", "AMPUSER=3010") in new stack
-- Executing [s@macro-user-callerid:5] Set("SIP/3010-0000001d", "AMPUSERCIDNAME=Departamento de Sistemas") in new stack
-- Executing [s@macro-user-callerid:6] GotoIf("SIP/3010-0000001d", "0?report") in new stack
-- Executing [s@macro-user-callerid:7] Set("SIP/3010-0000001d", "AMPUSERCID=3010") in new stack
-- Executing [s@macro-user-callerid:8] Set("SIP/3010-0000001d", "CALLERID(all)=Departamento de Sistemas <3010>") in new stack
-- Executing [s@macro-user-callerid:9] ExecIf("SIP/3010-0000001d", "1?Set(CHANNEL(language)=es)") in new stack
-- Executing [s@macro-user-callerid:10] GotoIf("SIP/3010-0000001d", "0?continue") in new stack
-- Executing [s@macro-user-callerid:11] Set("SIP/3010-0000001d", "_PTL=64") in new stack
-- Executing [s@macro-user-callerid:12] GotoIf("SIP/3010-0000001d", "1?continue") in new stack
-- Goto (macro-user-callerid,s,19)
-- Executing [s@macro-user-callerid:19] NoOp("SIP/3010-0000001d", "Using CallerID 'Departamento de Sistemas' <3010>") in new stack
-- Executing [s@macro-exten-vm:2] Set("SIP/3010-0000001d", "RingGroupMethod=none") in new stack
-- Executing [s@macro-exten-vm:3] Set("SIP/3010-0000001d", "VMBOX=1010") in new stack
-- Executing [s@macro-exten-vm:4] Set("SIP/3010-0000001d", "EXTTOCALL=1010") in new stack
-- Executing [s@macro-exten-vm:5] Set("SIP/3010-0000001d", "CFURST") in new stack
-- Executing [s@macro-exten-vm:6] Set("SIP/3010-0000001d", "CFBKXT") in new stack
-- Executing [s@macro-exten-vm:7] Set("SIP/3010-0000001d", "RT=15") in new stack
-- Executing [s@macro-exten-vm:8] Macro("SIP/3010-0000001d", "record-enable,1010,1010") in new stack
-- Executing [s@macro-record-enable:1] GotoIf("SIP/3010-0000001d", "1?check") in new stack
-- Goto (macro-record-enable,s,4)
-- Executing [s@macro-record-enable:4] ExecIf("SIP/3010-0000001d", "0?MacroExit()") in new stack
-- Executing [s@macro-record-enable:5] GotoIf("SIP/3010-0000001d", "0?Group:OUT") in new stack
-- Goto (macro-record-enable,s,15)
-- Executing [s@macro-record-enable:15] GotoIf("SIP/3010-0000001d", "1?IN") in new stack
-- Goto (macro-record-enable,s,20)
-- Executing [s@macro-record-enable:20] ExecIf("SIP/3010-0000001d", "1?MacroExit()") in new stack
-- Executing [s@macro-exten-vm:9] Macro("SIP/3010-0000001d", "dial,15,tr,1010") in new stack
-- Executing [s@macro-dial:1] GotoIf("SIP/3010-0000001d", "1?dial") in new stack
-- Goto (macro-dial,s,3)
-- Executing [s@macro-dial:3] AGI("SIP/3010-0000001d", "dialparties.agi") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/dialparties.agi
dialparties.agi: Starting New Dialparties.agi
dialparties.agi: Caller ID name is 'Departamento de Sistemas' number is '3010'
dialparties.agi: Methodology of ring is 'none'
-- dialparties.agi: Added extension 1010 to extension map
-- dialparties.agi: Extension 1010 is disabled
-- dialparties.agi: Extension 1010 do not disturb is disabled

-- <SIP/3010-0000001d>AGI Script dialparties.agi completed, returning 0
-- Executing [s@macro-dial:7] Dial("SIP/3010-0000001d", "SIP/1010,15,tr") in new stack
== Using SIP RTP TOS bits 184
== Using SIP RTP CoS mark 5
-- Called 1010
-- SIP/1010-0000001d is ringing
-- SIP/1010-0000001d answered SIP/3010-0000001d
-- Executing [h@macro-dial:1] Macro("SIP/3010-0000001d", "hangupcall") in new stack
-- Executing [s@macro-hangupcall:1] GotoIf("SIP/3010-0000001d", "1?noautomon") in new stack
-- Goto (macro-hangupcall,s,3)
-- Executing [s@macro-hangupcall:3] NoOp("SIP/3010-0000001d", "TOUCH_EXITCH_TRIGGER") in new stack
-- Executing [s@macro-hangupcall:4] GotoIf("SIP/3010-0000001d", "1?skipprg") in new stack
-- Goto (macro-hangupcall,s,7)
-- Executing [s@macro-hangupcall:7] GotoIf("SIP/3010-0000001d", "1?skipblkvm") in new stack
-- Goto (macro-hangupcall,s,10)
-- Executing [s@macro-hangupcall:10] GotoIf("SIP/3010-0000001d", "1?theend") in new stack
-- Goto (macro-hangupcall,s,12)
-- Executing [s@macro-hangupcall:12] Hangup("SIP/3010-0000001d", "") in new stack
== Spawn extension (macro-hangupcall, s, 12) exited non-zero on 'SIP/3010-0000001d' in macro 'hangupcall'
== Spawn extension (macro-dial, h, 1) exited non-zero on 'SIP/3010-0000001d'
== Spawn extension (macro-exten-vm, s, 9) exited non-zero on 'SIP/3010-0000001d' in macro 'exten-vm'
== Spawn extension (from-internal, 1010, 1) exited non-zero on 'SIP/3010-0000001d'
-- Executing [h@from-internal:1] Macro("SIP/3010-0000001d", "hangupcall") in new stack
-- Executing [s@macro-hangupcall:1] GotoIf("SIP/3010-0000001d", "1?noautomon") in new stack
-- Goto (macro-hangupcall,s,3)
-- Executing [s@macro-hangupcall:3] NoOp("SIP/3010-0000001d", "TOUCH_EXITCH_TRIGGER") in new stack
-- Executing [s@macro-hangupcall:4] GotoIf("SIP/3010-0000001d", "1?skipprg") in new stack
-- Goto (macro-hangupcall,s,7)
-- Executing [s@macro-hangupcall:7] GotoIf("SIP/3010-0000001d", "1?skipblkvm") in new stack
-- Goto (macro-hangupcall,s,10)
-- Executing [s@macro-hangupcall:10] GotoIf("SIP/3010-0000001d", "1?theend") in new stack
-- Goto (macro-hangupcall,s,12)
-- Executing [s@macro-hangupcall:12] Hangup("SIP/3010-0000001d", "") in new stack
== Spawn extension (macro-hangupcall, s, 12) exited non-zero on 'SIP/3010-0000001d' in macro 'hangupcall'
== Spawn extension (from-internal, h, 1) exited non-zero on 'SIP/3010-0000001d'
-- Registered IAX2 '1020' (AUTHENTICATED) at 127.0.0.1:40000
[Apr 27 17:52:36] NOTICE[2831]: : Restricting registration for peer '1020' to 60 seconds (requested 300)
-- Registered IAX2 '1020' (AUTHENTICATED) at 168.0.0.175:4569
-- Registered IAX2 '1020' (AUTHENTICATED) at 127.0.0.1:40000
[Apr 27 17:53:31] NOTICE[2825]: : Restricting registration for peer '1020' to 60 seconds (requested 300)
```

Figura 6-4 Consola Asterisk muestra ruteo de llamada

6.2 Prueba de llamadas entrantes y salientes

En este punto lo que nos interesa es comprobar que las llamadas se realicen exitosamente tanto las que entran como las que salen para lo que visualizaremos que ocurre en la línea de comando de Asterisk al momento de realizar las llamadas.

Llamadas Entrantes

Al marcar el número de nuestra base celular, ingresa a la central telefónica y lo primero que nos accede a nuestro IVR dando el mensaje de bienvenida para poder escoger la opción que deseemos:

```
-- Executing [s@from-pstn:1] Answer("DAHDI/2-1", "") in new stack
-- Executing [s@from-pstn:2] Goto("DAHDI/2-1", "IVR,185,1") in new stack
-- Goto (IVR,185,1)
-- Executing [185@IVR:1] Wait("DAHDI/2-1", "1") in new stack
-- Remote UNIX connection
-- Remote UNIX connection disconnected
-- Executing [185@IVR:2] Background("DAHDI/2-1", "custom/bienvenida") in new stack
-- <DAHDI/2-1> Playing 'custom/bienvenida.slin' (language 'es')
-- Executing [185@IVR:3] WaitExten("DAHDI/2-1", ",") in new stack
```

Figura 6- 5 Prueba de llamada Entrante

Llamadas Salientes

Estamos realizando llamada al número 092231671 agregamos el prefijo 9 para decirle que salga por un canal definido en la troncal, en el softphone ZOIPER quedaría de la siguiente manera:



Figura 6-6 Llamada desde zoiper

En las líneas de comando nos muestra el ruteo de la llamada, siendo esta un éxito:

```

-- Executing [9092231671@from-internal:1] Macro("SIP/4010-0000000c", "user-callerid.SKIPPTL") in new stack
-- Executing [s@macro-user-callerid:1] Set("SIP/4010-0000000c", "AMPUSER=4010") in new stack
-- Executing [s@macro-user-callerid:2] GotoIf("SIP/4010-0000000c", "0?report") in new stack
-- Executing [s@macro-user-callerid:3] ExecIf("SIP/4010-0000000c", "1?Set(REALCALLERIDNUM=4010)") in new stack
-- Executing [s@macro-user-callerid:4] Set("SIP/4010-0000000c", "AMPUSER=4010") in new stack
-- Executing [s@macro-user-callerid:5] Set("SIP/4010-0000000c", "AMPUSERCIDNAME=Departamento de Facturacion") in new stack
-- Executing [s@macro-user-callerid:6] GotoIf("SIP/4010-0000000c", "0?report") in new stack
-- Executing [s@macro-user-callerid:7] Set("SIP/4010-0000000c", "AMPUSERCID=4010") in new stack
-- Executing [s@macro-user-callerid:8] Set("SIP/4010-0000000c", "CALLERID(all)=Departamento de Facturacion <4010>") in new stack
-- Executing [s@macro-user-callerid:9] ExecIf("SIP/4010-0000000c", "0?Set(CHANNEL(language)=)") in new stack
-- Executing [s@macro-user-callerid:10] GotoIf("SIP/4010-0000000c", "1?continue") in new stack
-- Goto (macro-user-callerid,s,19)
-- Executing [s@macro-user-callerid:19] NoOp("SIP/4010-0000000c", "Using CallerID 'Departamento de Facturacion' <4010>") in new stack
[Apr 27 15:49:16] WARNING[4348]: : : Can't find trailing parenthesis for function 'LEN('?
[Apr 27 15:49:16] WARNING[4348]: : : Can't find trailing parenthesis for function 'LEN('?
-- Executing [9092231671@from-internal:2] ExecIf("SIP/4010-0000000c", "1?Set(TRUNKCIDOVERRIDE=082725877)") in new stack
-- Executing [9092231671@from-internal:3] Set("SIP/4010-0000000c", "XDEST=") in new stack
-- Executing [9092231671@from-internal:4] Macro("SIP/4010-0000000c", "record-enable,4010,OUT,") in new stack
-- Executing [s@macro-record-enable:1] GotoIf("SIP/4010-0000000c", "1?check") in new stack
-- Goto (macro-record-enable,s,4)
-- Executing [s@macro-record-enable:4] ExecIf("SIP/4010-0000000c", "0?MacroExit()") in new stack
-- Executing [s@macro-record-enable:5] GotoIf("SIP/4010-0000000c", "0?Group:OUT") in new stack
-- Goto (macro-record-enable,s,15)
-- Executing [s@macro-record-enable:15] GotoIf("SIP/4010-0000000c", "0?IN") in new stack
-- Executing [s@macro-record-enable:16] ExecIf("SIP/4010-0000000c", "1?MacroExit()") in new stack
-- Executing [9092231671@from-internal:5] Macro("SIP/4010-0000000c", "dialout-trunk,1,092231671,") in new stack
-- Executing [s@macro-dialout-trunk:1] Set("SIP/4010-0000000c", "DIAL_TRUNK=1") in new stack
-- Executing [s@macro-dialout-trunk:2] GotoIf("SIP/4010-0000000c", "0?sub-pincheck,s,1") in new stack
-- Executing [s@macro-dialout-trunk:3] GotoIf("SIP/4010-0000000c", "0?disabletrunk,1") in new stack
-- Executing [s@macro-dialout-trunk:4] Set("SIP/4010-0000000c", "DIAL_NUMBER=092231671") in new stack
-- Executing [s@macro-dialout-trunk:5] Set("SIP/4010-0000000c", "DIAL_TRUNK_OPTIONS=tr") in new stack
-- Executing [s@macro-dialout-trunk:6] Set("SIP/4010-0000000c", "OUTBOUND_GROUP=OUT_1") in new stack
-- Executing [s@macro-dialout-trunk:7] GotoIf("SIP/4010-0000000c", "0?nomax") in new stack
-- Executing [s@macro-dialout-trunk:8] GotoIf("SIP/4010-0000000c", "0?chanfull") in new stack
-- Executing [s@macro-dialout-trunk:9] GotoIf("SIP/4010-0000000c", "0?skipoutcid") in new stack
-- Executing [s@macro-dialout-trunk:10] Set("SIP/4010-0000000c", "DIAL_TRUNK_OPTIONS=") in new stack
-- Executing [s@macro-dialout-trunk:11] Macro("SIP/4010-0000000c", "outbound-callerid,1") in new stack
-- Executing [s@macro-outbound-callerid:1] ExecIf("SIP/4010-0000000c", "0?Set(CALLERPRES(i)=)") in new stack
-- Executing [s@macro-outbound-callerid:2] ExecIf("SIP/4010-0000000c", "0?Set(REALCALLERIDNUM=4010)") in new stack
-- Executing [s@macro-outbound-callerid:3] GotoIf("SIP/4010-0000000c", "1?normcid") in new stack
-- Goto (macro-outbound-callerid,s,6)
-- Executing [s@macro-outbound-callerid:6] Set("SIP/4010-0000000c", "USEROUTCID=") in new stack
-- Executing [s@macro-outbound-callerid:7] Set("SIP/4010-0000000c", "EMERGENCYCID=") in new stack
-- Executing [s@macro-outbound-callerid:8] Set("SIP/4010-0000000c", "TRUNKOUTCID=082725877") in new stack
-- Executing [s@macro-outbound-callerid:9] GotoIf("SIP/4010-0000000c", "1?trunkcid") in new stack
-- Goto (macro-outbound-callerid,s,12)
-- Executing [s@macro-outbound-callerid:12] ExecIf("SIP/4010-0000000c", "1?Set(CALLERID(all)=082725877)") in new stack
-- Executing [s@macro-outbound-callerid:13] ExecIf("SIP/4010-0000000c", "0?Set(CALLERID(all)=)") in new stack
-- Executing [s@macro-outbound-callerid:14] ExecIf("SIP/4010-0000000c", "1?Set(CALLERID(all)=082725877)") in new stack
-- Executing [s@macro-outbound-callerid:15] ExecIf("SIP/4010-0000000c", "0?Set(CALLERPRES(i)=prohib_passed_screen)") in new stack
-- Executing [s@macro-dialout-trunk:12] ExecIf("SIP/4010-0000000c", "0?AGI(fixlocalprefix)") in new stack
-- Executing [s@macro-dialout-trunk:13] Set("SIP/4010-0000000c", "OUTNUM=092231671") in new stack
-- Executing [s@macro-dialout-trunk:14] Set("SIP/4010-0000000c", "custom=DAHDI/2") in new stack
-- Executing [s@macro-dialout-trunk:15] ExecIf("SIP/4010-0000000c", "0?Set(DIAL_TRUNK_OPTIONS=M(setmusic^))") in new stack
-- Executing [s@macro-dialout-trunk:16] Macro("SIP/4010-0000000c", "dialout-trunk-predial-hook,") in new stack
-- Executing [s@macro-dialout-trunk-predial-hook:1] MacroExit("SIP/4010-0000000c", "") in new stack
-- Executing [s@macro-dialout-trunk:17] GotoIf("SIP/4010-0000000c", "0?bypass,1") in new stack
-- Executing [s@macro-dialout-trunk:18] GotoIf("SIP/4010-0000000c", "0?customtrunk") in new stack
-- Executing [s@macro-dialout-trunk:19] Dial("SIP/4010-0000000c", "DAHDI/2/092231671,300,") in new stack
-- Called 2/092231671
-- DAHDI/2-1 answered SIP/4010-0000000c
-- Executing [h@macro-dialout-trunk:1] Macro("SIP/4010-0000000c", "hangupcall,") in new stack
-- Executing [s@macro-hangupcall:1] GotoIf("SIP/4010-0000000c", "1?noautomon") in new stack
-- Goto (macro-hangupcall,s,3)
-- Executing [s@macro-hangupcall:3] NoOp("SIP/4010-0000000c", "TOUCH_MONITOR_OUTPUT=") in new stack
-- Executing [s@macro-hangupcall:4] GotoIf("SIP/4010-0000000c", "1?skipprq") in new stack
-- Goto (macro-hangupcall,s,7)
-- Executing [s@macro-hangupcall:7] GotoIf("SIP/4010-0000000c", "1?skipblkvm") in new stack
-- Goto (macro-hangupcall,s,10)
-- Executing [s@macro-hangupcall:10] GotoIf("SIP/4010-0000000c", "1?theend") in new stack
-- Goto (macro-hangupcall,s,12)
-- Executing [s@macro-hangupcall:12] Hangup("SIP/4010-0000000c", "") in new stack
== Spawn extension (macro-hangupcall, s, 12) exited non-zero on 'SIP/4010-0000000c' in macro 'hangupcall'
== Spawn extension (macro-dialout-trunk, h, 1) exited non-zero on 'SIP/4010-0000000c'
-- Hungup 'DAHDI/2-1'
== Spawn extension (macro-dialout-trunk, s, 19) exited non-zero on 'SIP/4010-0000000c' in macro 'dialout-trunk'
== Spawn extension (from-internal, 9092231671, 5) exited non-zero on 'SIP/4010-0000000c'
-- Executing [h@from-internal:1] Macro("SIP/4010-0000000c", "hangupcall") in new stack
-- Executing [s@macro-hangupcall:1] GotoIf("SIP/4010-0000000c", "1?noautomon") in new stack
-- Goto (macro-hangupcall,s,3)
-- Executing [s@macro-hangupcall:3] NoOp("SIP/4010-0000000c", "TOUCH_MONITOR_OUTPUT=") in new stack
-- Executing [s@macro-hangupcall:4] GotoIf("SIP/4010-0000000c", "1?skipprq") in new stack
-- Goto (macro-hangupcall,s,7)
-- Executing [s@macro-hangupcall:7] GotoIf("SIP/4010-0000000c", "1?skipblkvm") in new stack
-- Goto (macro-hangupcall,s,10)
-- Executing [s@macro-hangupcall:10] GotoIf("SIP/4010-0000000c", "1?theend") in new stack
-- Goto (macro-hangupcall,s,12)
-- Executing [s@macro-hangupcall:12] Hangup("SIP/4010-0000000c", "") in new stack
== Spawn extension (macro-hangupcall, s, 12) exited non-zero on 'SIP/4010-0000000c' in macro 'hangupcall'
== Spawn extension (from-internal, h, 1) exited non-zero on 'SIP/4010-0000000c'

```

Figura 6-7 Prueba de Llamadas salientes

6.3 Prueba de OpenFire

Aquí lo que nos interesa es saber que los mensajes se están enviando y recibiendo correctamente para lo cual necesitaremos que 2 computadoras estén conectadas al OpenFire con el spark, y enviar mensajes entre sí; en la

computadora número uno, estamos conectados a spark con la cuenta de contabilidad y en la maquina numero dos estamos conectados con la cuenta de gerencia; Al enviar un mensaje desde la máquina de contabilidad vemos que si le está llegando a la máquina de gerencia.

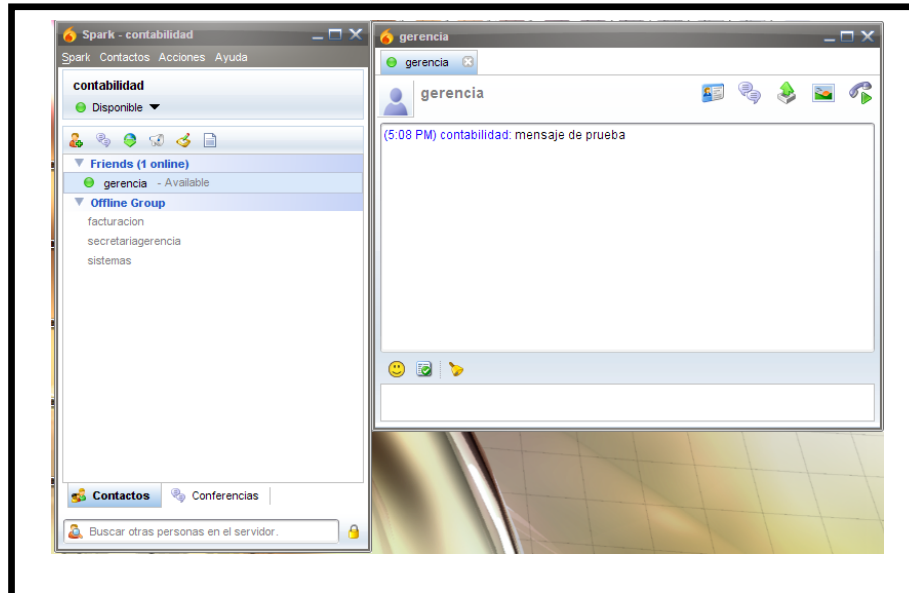


Figura 6-8 Computadora #1 conectada con Spark

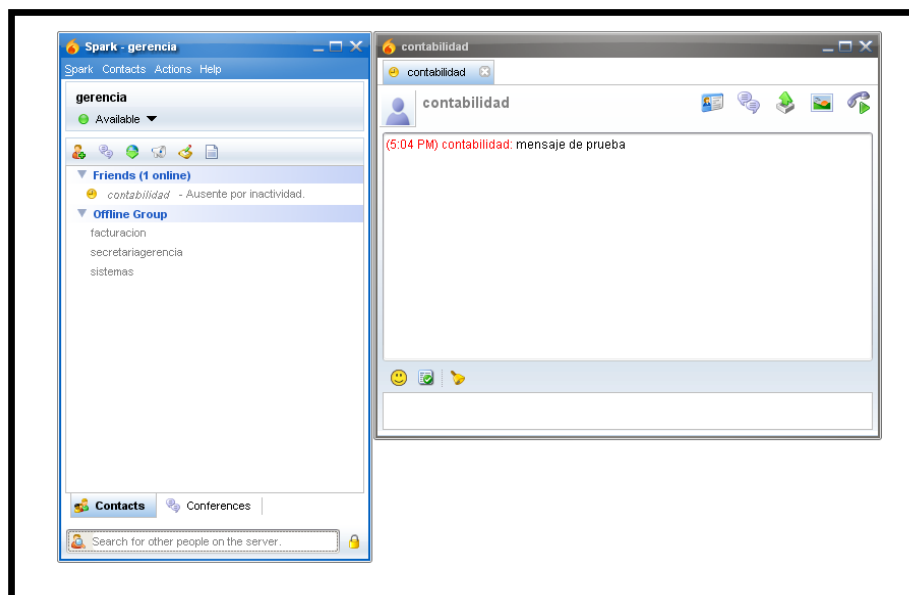


Figura 6-9 computadora # conectada con spark

6.4 Prueba de Hylafax

En esta sección de las pruebas debemos probar que el fax virtual sirve para lo cual enviaremos un fax, para lo cual ingresaremos a Word y crearemos un archivo con el texto “Documento de prueba” y procederemos a enviar

a la extensión 1020 que pertenece a la secretaria de gerencia y el fax se enviara al correo de la secretaria.

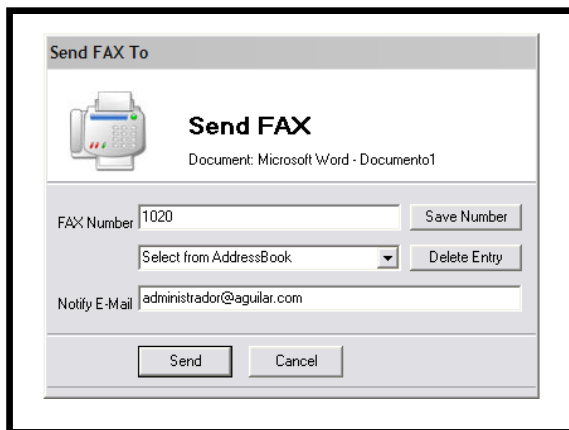


Figura 6-10 Enviar Fax

6.5 Prueba de Sintetización de Voz

La prueba de sintetización la podemos realizar de 2 maneras:

- Ingresamos a festival con el comando festival, y ponemos el comando (SayText “Prueba”), y nos devolverá el sonido.
- La segunda forma es reproducir los archivos que anteriormente transformamos de texto a formato ulaw con el comando text2 wave.

6.6 Prueba de CallCenter

Después procedemos a Ingresar a **la pestaña “CallCenter”** y luego al menú llamado “Agent Console”, con los siguientes datos:

Número de Agente: 4010

Extensión: 4010

De la siguiente forma:



Figura 6-11 Conectarse al Agente

Luego pulsamos el botón "Ingresar", en ese momento nuestro anexo 4010 (en mi caso un softphone) comenzara a timbrar, y nos pedirá con un mensaje en ingles, que ingresemos nuestra contraseña de agente seguido de la tecla #, digitamos "1234#", para este caso y nos permitirá ingresar a la consola del agente, tendría que ver esto:



Figura 6-12 Conectando Elastix con Zoiper

Luego de algunos segundos, en cuanto el marcador progresivo detecte que una llamada ha sido establecida debería enlazarnos con esa llamada cambiando el estado de la consola del agente de la siguiente forma:

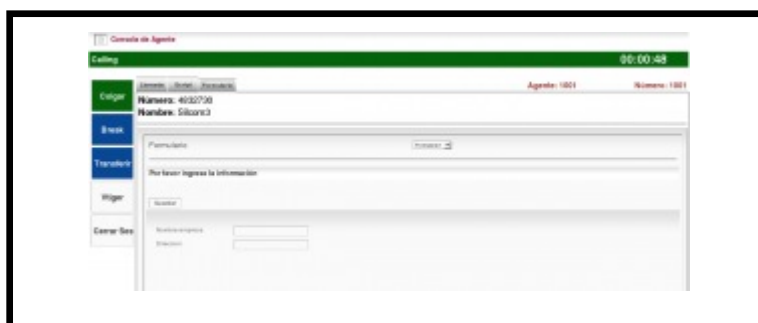


Figura 6-13 Consola del Agente

Luego de haber conversado haber terminado la conversación con el cliente tendríamos que hacer clic en el botón "Colgar" para que se libere el agente y nos enlace con la siguiente llamada.

CONCLUSIONES

- Asterisk soluciona la gestión de comunicaciones telefónicas tanto como llamadas salientes, entrantes o re direccionamiento dentro de la misma red utilizando proveedores de telefonía celular.
- Se mostro la correcta integración de Asterisk con Agi para la elaboración de diferentes programas, la integración de estas 2 herramientas es muy útil ya que se puede realizar mejoras en una empresa.
- Se comprobó que Asterisk permite manejar un fluido registro telefónico, se pudieron realizar validaciones de una forma más sencilla a través de AGI.
- No solo se pueden realizar operaciones desde una interfaz web de forma sencilla sino que también por medio de un sistema telefónico que brinde a los usuarios instrucciones de cómo realizarlo.

RECOMENDACIONES

- Al momento de implementar el AGI hay que tener en cuenta que debemos tener todos los archivos dentro de la carpeta agi-bin que se encuentra en el path /var/lib/Asterisk/agi-bin
- Al momento de correr el Agi debemos tener en cuenta que se debe levantar el servicio.

GLOSARIO

Dialplan: Un plan de marcado establece el número esperado y el patrón de dígitos de un número de teléfono. Esto incluye los códigos de país, códigos de acceso, códigos de área y todas las combinaciones de dígitos marcados.

Gateway: Es un dispositivo, con frecuencia un ordenador, que permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación. Su propósito es traducir la información del protocolo utilizado en una red al protocolo usado en la red de destino.

Open Source: Código abierto (en inglés open source) es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre.

PBX: Un PBX es cualquier central telefónica conectada directamente a la red pública de teléfono por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y/o salientes con autonomía sobre cualquier otra central telefónica.

MAC: Media Access Control (Control de acceso al medio). Identificador hexadecimal de 48 bits que corresponde de manera única a cualquier interfaz o dispositivo de red (routers, switch, tarjetas de red)

MACRO: es una serie de instrucciones que se almacenan para que se puedan ejecutar de forma secuencial mediante una sola llamada u orden de ejecución.

SOFTPHONE: Un Softphone (en inglés combinación de Software y de Telephone) es un software que hace una simulación de teléfono convencional por computadora.

TRONCAL: En lenguaje técnico de telefonía, una línea troncal es un enlace que interconecta las llamadas externas de una central telefónica, concentrando y unificando varias comunicaciones simultáneas en una sola señal para un transporte y transmisión a distancia más eficiente (generalmente digital) y poder establecer comunicaciones con otra central o una red entera de ellas.

VOIP: Voz sobre Protocolo de Internet, también llamado Voz sobre IP, VozIP, VOIP (por sus siglas en inglés), es un grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP (Internet Protocol).

BIBLIOGRAFIA

- Landivar, E. (2009 de Marzo de 29). *Volumen uno Manual de Elastix*. Recuperado el 30 de 12 de 2010, de Elastix:
http://ufpr.dl.sourceforge.net/project/elastix/Tutorials_Docs_Manuals/Comunicaciones%20Unificadas%20con%20Elastix/Comunicaciones_Unificadas_con_Elastix_Volumen_1_29Mar2009.pdf
- Volumen Dos Manual de Elastix, Disponible en la Web, 1 de diciembre del 2010
http://ufpr.dl.sourceforge.net/project/elastix/Tutorials_Docs_Manuals/Comunicaciones%20Unificadas%20con%20Elastix/Comunicaciones_Unificadas_con_Elastix_Volumen_2_29Mar2009.pdf
- David Gomillion, Construyendo Sistemas telefónicos con Asterisk, Packt Publishing, Abril 10, 2007
- Tutorial de red IP, Disponible en la Web, 6 de febrero del 2011
<http://neo.lcc.uma.es/evirtual/cdd/tutorial/red/ip.html>
- Voz Ip, Disponible en la Web, 15 de febrero del 2010
<http://es.kioskea.net/contents/internet/ip.php3>
- Protocolo IAX2, Disponible en la web 1 de marzo del 2011
<http://www.alcancelibre.org/article.php/protocolo-iax2-es-oficialmente-rfc5456>
- Ventajas y Desventajas de la telefonía VOIP, Disponible en la Web, el 6 de marzo del 2011
<http://www.informatica-hoy.com.ar/voz-ip-VOIP/Ventajas-y-desventajas-de-la-tecnologia-VOIP.php>
- Protocolo IAX2, Disponible en la web, el 10 de marzo del 2011
http://www.asteriskguide.com/mediawiki/index.php/El_Protocolo_IAX
- Redes de Comunicaciones, Disponible en la Web, 10 de marzo del 2011
http://guimi.net/monograficos/G-Redes_de_comunicaciones/G-RCnode67.html
- Protocolo IAX, Disponible en la Web, 19 de marzo del 2011
http://www.VOIP.unam.mx/mediawiki/index.php/Protocolo_IAX