

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación



DISEÑO E IMPLEMENTACIÓN DE UNA AGENDA TELEFÓNICA EN LÍNEA QUE PERMITA AL USUARIO LLAMAR A SUS CONTACTOS DESDE UNA PÁGINA WEB

INFORME DE MATERIA DE GRADUACIÓN

Previa a la obtención del Título de:

INGENIERO EN TELEMÁTICA

Presentado por:

SORAYA MARIELA MINGA BASTIDAS

JUAN LEÓN MERA MEJÍA

GUAYAQUIL - ECUADOR

2012

AGRADECIMIENTO

Agradezco a Dios y a mi familia, por darme la oportunidad y los recursos necesarios para culminar con éxito mi formación profesional. Además a mis amigos que con su apoyo moral y conocimientos, promovieron a la culminación de mi carrera.

Soraya M. Minga Bastidas

Agradezco a Dios que es mi fortaleza, a mi familia que es mi inspiración y mi apoyo y a mis amigos que siempre han estado en los momentos buenos y malos para dar una mano.

Juan León Mera Mejía

DEDICATORIA

Dedico este trabajo a mis padres y profesores, quienes demostraron gran paciencia y apoyo incondicional en este largo camino.

Soraya M. Minga Bastidas

Dedico el esfuerzo hecho a mi padre, a mi madre y a mi hermana, cada uno contribuyo de diferente forma para que pueda cumplir este objetivo.

Juan León Mera Mejía

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Trabajo de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela superior Politécnica del Litoral".

(Reglamento de Graduación de la ESPOL)



Soraya Mariela Minga Bastidas



Juan León Mera Mejía

TRIBUNAL DE SUSTENTACIÓN

Ing. Gabriel Astudillo

PROFESOR DE LA MATERIA DE GRADUACIÓN



Ing. Patricia Chávez

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

RESUMEN

Este proyecto consistió en la elaboración de una Agenda Telefónica Virtual sobre una plataforma web, la cual podrá ser utilizada desde cualquier computadora con acceso a Internet y un teléfono IP, sea este real o virtual. Los usuarios deben registrarse para hacer uso del sistema y contarán con tres números telefónicos personales. Antes de realizar una llamada se deberá definir el número al cual desea ser contactado, para así hacer uso de la retro-llamada en caso de ser necesario. De esta manera podrán comunicarse fácilmente con los contactos de su agenda, evitando llamadas exageradamente costosas y la necesidad de memorizar información innecesaria.

La página web fue desarrollada con lenguaje PHP y montada en un servidor Apache para poder ser visualizada desde cualquier navegador. Para el manejo de las llamadas se hizo uso de una central telefónica basada en Asterisk; mientras que la fusión de ambas partes se llevo a cabo mediante el uso de una base de datos MySQL, la misma que contendrá información necesaria tanto para la central, como para el sitio web. Todo esto sobre una distribución de Linux, como es Fedora 16.

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
DECLARACIÓN EXPRESA	IV
TRIBUNAL DE SUSTENTACIÓN	V
RESUMEN	VI
ÍNDICE GENERAL	VII
ABREVIATURAS	XI
ÍNDICE DE FIGURAS	XII
ÍNDICE DE TABLAS	XV
INTRODUCCIÓN	XVI
CAPITULO 1 ANTECEDENTES Y	
JUSTIFICACIÓN	1
1.1. Antecedentes	1
1.2. Justificación	2
1.3. Descripción del Proyecto	2
1.3.1. Objetivos Generales	3
1.3.2. Objetivos Específicos	3
1.4. Metodología	5
CAPITULO 2	7
MARCO TEÓRICO	7
2.1. Asterisk	7

2.2. SIP	9
2.3. AMI	10
2.4. AGI.....	11
2.5. ARA	12
CAPITULO 3	14
IMPLEMENTACIÓN	14
3.1. Introducción	14
3.2. Hardware	15
3.3. Software.....	16
3.3.1. Central Telefónica.....	16
3.3.2. Sitio Web	17
3.3.3. Softphone	17
3.4. Instalación.....	18
3.5. Configuración.....	19
3.5.1. Seguridad	19
3.5.2. Servicios Adicionales	20
3.5.3. Archivos de Asterisk:	21
3.5.4. Base de Datos.	26
3.5.5. Scripts.....	37
CAPITULO 4	40
FUNCIONAMIENTO Y PRUEBAS	40
4.1. Iniciando servicios	40

4.2. Guía del Usuario	41
4.2.1. Registro del Cliente	42
4.2.2. Agregar Contactos	45
4.2.3. Configuración de Softphone	47
4.2.4. Compra de Saldo	51
4.2.5. Realización de la llamada	52
4.2.6. Configuraciones opcionales	53
4.3. Funcionamiento	57
4.3.1. Llamada	57
4.3.2. Menú: Buzón de Mensajes.	61
4.3.3. Menú: Retrollamada	62
4.3.4. Llamada al Buzón de Voz	64
CONCLUSIONES Y RECOMENDACIONES	65
TRABAJO A FUTURO	68
APÉNDICE	69
GLOSARIO DE TERMINOS	70
ANEXOS	72
ANEXO A: CÓDIGO	73
I. Fichero “res_config_mysql.conf”	73
II. Fichero “extconfig.conf”	74
III. Fichero “sip.conf”	75
IV. Fichero “extensions.conf”	76

ANEXO B: FIGURAS	78
ANEXO C: TABLAS	80
BIBLIOGRAFÍA	83

ABREVIATURAS

AGI	Interfaz de Puerta de Enlace de Asterisk
ARA	Arquitectura de Tiempo-Real de Asterisk
AMI	Interfaz de Administración de Asterisk
PBX	Central Privada de Intercambio
BSD	Distribución de Software Berkeley
SIP	Protocolo de Inicio de Sesión
IAX	Protocolo de Intercambio Entre-Asterisk Versión 2.
VoIP	Voz sobre Protocolo de Internet
RTP	Protocolo de Transporte de Tiempo Real
SDP	Protocolo de Descripción de Sesión
TCP	Protocolo de Control de Transmisión
UDP	Protocolo de Datagrama de Usuario
SQL	Lenguaje de Consulta Estructurado
VSP	Proveedor de Servicio de VoIP
SELinux	Seguridad-Mejorada de Linux
CDR	Historial de Detalle de Llamadas
PHP	Pre-Procesador de HiperTexto

ÍNDICE DE FIGURAS

Figura 3.1. Extensiones por cliente registrado	29
Figura 3.2. Macro – Extensions	30
Figura 3.3. Opción: Mensaje de Voz.....	31
Figura 3.4. Opción: Retrollamada.....	32
Figura 3.5. Opción: Inválida o Fuera de Tiempo	32
Figura 3.6. Llamada al Buzón de Voz.....	32
Figura 3.7. Macro - Retrollamada.....	33
Figura 3.8. Contexto: extensions - inverso.....	34
Figura 3.9. Script consultarTiempo.....	37
Figura 3.10. Script setearSaldo.....	38
Figura 4.1. Página Principal.....	42
Figura 4.2. Enlace de Registro.....	43
Figura 4.3. Página de Registro	44
Figura 4.4. Agenda del Usuario	45
Figura 4.5. Búsqueda de Contactos.....	45
Figura 4.6. Resultado de la búsqueda de usuarios	46
Figura 4.7. Contacto agregado en la agenda	46
Figura 4.8. Perfil del Usuario - Info. Teléfono.....	47
Figura 4.9. Configuración X-Lite v4.0.....	48
Figura 4.10. Configuración Twinkle	49

Figura 4.11. Twinkle - Establecer cuenta por defecto	50
Figura 4.12. Twinkle - Correctamente registrado.....	50
Figura 4.13. Página de Saldo.....	51
Figura 4.14. Elegir ubicación del llamante	52
Figura 4.15. Contacto	52
Figura 4.16. Página para editar Perfil	54
Figura 4.17. Elegir foto del Perfil.....	55
Figura 4.18. Desactivar cuenta	55
Figura 4.19. Botón del Buzón de Voz	56
Figura 4.20. Eliminar Contactos.....	56
Figura 4.21. Elección de la extensión: casa	57
Figura 4.22. Softphone del Llamante	58
Figura 4.23. Llamada en el Softphone del destinatario.....	58
Figura 4.24. Llamado al contacto	59
Figura 4.25. Realización de la llamada en la central.....	59
Figura 4.26. Llamada finalizada	60
Figura 4.27. Llamada finalizada en la central	60
Figura 4.28. Llamada no contestada.....	61
Figura 4.29. Reproducción de menú	61
Figura 4.30. Dejar mensaje de voz.....	62
Figura 4.31. Ejecución macro-retrollamada	63
Figura 4.32. Ejecución contexto: extensions-inverso	63

Figura 4.33. Retrollamada contestada.....	64
Figura 4.34. Llamada al buzón de voz en la central.....	64
Figura B-I. Modelo lógico de la Base de Datos: asterisk	78
Figura B-II. Script realizarLlamada	79

ÍNDICE DE TABLAS

Tabla I. Características del Equipo Anfitrión.....	15
Tabla II. Características de la Máquina Virtual.....	16
Tabla III. Software utilizado en la PBX.....	16
Tabla IV. Software utilizado en el sitio web.....	17
Tabla V. Software utilizado como Softphones.....	17
Tabla VI. Descripción de la Tabla: users.....	27
Tabla VII. Descripción de la Tabla: agenda.....	27
Tabla VIII. Descripción de la Tabla: sip_buddies.....	28
Tabla IX. Descripción de la Tabla: extensions.....	29
Tabla X. Descripción de la Tabla: voicemail.....	35
Tabla XI. Descripción de la Tabla: cdr.....	36

INTRODUCCIÓN

En la actualidad se requiere una constante comunicación con nuestra red de contactos, para así poder desarrollar mejores relaciones tanto a nivel laboral como personal. El principal problema que se nos presenta es la disponibilidad de la información, que puede estar distribuida en varios dispositivos, lo que nos lleva a tener que recordar en cual de todos ellos se encuentra lo que necesitamos, o por otro lado al centralizarla nos vemos obligados a depender de un solo equipo, esto resulta un gran inconveniente si el mismo no se encuentra accesible en el momento preciso.

Aquello que siempre se encuentra al alcance es Internet, con una gran capacidad para albergar información y una amplia versatilidad al momento de comunicar a las personas, brindándonos el entorno perfecto para almacenar la red de contactos y hacer uso de ella de forma más eficaz y eficiente. Es por tanto que se plantea la creación de un sistema usando la plataforma L.A.M.P.A. (Linux. Apache. MySQL. PHP. Asterisk.), que permita la realización de llamadas telefónicas prepagadas a través de VoIP mediante el acceso a un sitio web, el mismo que contiene la información de nuestros contactos con los cuales podremos comunicarnos de manera rápida y directa con un solo click.

CAPITULO 1

ANTECEDENTES Y JUSTIFICACIÓN

1.1. Antecedentes

Actualmente la mayoría de los ambientes empresariales o domésticos cuentan con una Intranet, ya sea para poder acceder a los recursos de la misma o sencillamente para conectarse a Internet. Teniendo en cuenta que este sistema ya se encuentra implementado en nuestros hogares u oficinas podemos obtener aún más provecho de él.

Una red convergente no solo permite transportar datos, existe transmisión de voz por igual. Aprovechar este recurso implica un ahorro significativo en la implementación de centrales telefónicas, que no solamente servirían para la comunicación interna sino que tendrían una cobertura mucho más amplia gracias a Internet y el equipamiento/configuración adecuada.

Al contar con una red de telefonía es necesario almacenar los números telefónicos de los diferentes usuarios, esto se puede escribir en una agenda o dispositivo electrónico. El problema de los artículos antes mencionados, es que pueden ser robados, olvidados o

extraviados. Lo ideal sería almacenar esta información en un lugar de alta disponibilidad para el usuario, el cual es la Web, para así hacer uso de esos datos directamente, sin necesidad de moverlos de una localidad a otra para realizar una llamada. Esta situación lleva al desarrollo de la Agenda Telefónica Virtual, TuxFono.

1.2. Justificación

La elaboración de una Agenda en línea permite acceso total a la información de la red de contactos; evita la dependencia de un dispositivo único, para su almacenamiento, e instalación de un software dedicado, el cual no podría usarse con otro servicio.

Se busca ofrecer opciones similares a una central telefónica convencional, como son: el buzón de mensajes y retollamada, todo por una tarifa única sin importar la ubicación del destinatario. Además permite la posibilidad de tener hasta tres líneas telefónicas independientes por usuario sin ningún costo adicional.

1.3. Descripción del Proyecto

El desarrollo de TuxFono pretende alcanzar los siguientes objetivos.

1.3.1. Objetivos Generales

Implementar un sistema de comunicaciones basado en VoIP, el cual incluye: una agenda telefónica en línea, una central telefónica exclusiva para este fin y un sistema de cobranzas. De esta forma se podrá establecer una comunicación confiable y eficaz entre los usuarios registrados en el mismo.

1.3.2. Objetivos Específicos

- Simplificar la conexión entre los llamantes, de tal forma que se elimina la necesidad de marcado, de recordar el número del destinatario y la dependencia de un dispositivo de almacenamiento.
- Proveer de gran movilidad a los usuarios, pues se podrá realizar llamadas con su agenda, solamente con acceder a Internet y tener un teléfono IP disponible.
- Integrar diversas herramientas, como son: Apache, como servidor para el sitio web desarrollado en PHP; junto con MySQL, como motor de base de datos y Asterisk.

El proyecto consiste en la creación de una agenda telefónica virtual, la cual permitirá a los usuarios registrados poder llamar a los contactos que tengan en su agenda con solo un click. Además cuenta con la

capacidad de ejecutar una retrollamada, en caso de que el destinatario se encuentre ocupado o no disponible para contestar el teléfono.

El sistema se denomina: **TuxFono**.

El usuario se registra ingresando sus datos personales e información de una tarjeta de crédito, finalizada la operación debe dirigirse a su perfil donde se le mostrará los usuarios y contraseñas que deberá configurar en los dispositivos que conectará al sistema. Posteriormente debe buscar entre los demás clientes, a aquellos que desee agregar a su agenda y añadirlos. Luego se procede con la obtención de saldo, esto se hace mediante la compra de las tarjetas virtuales que se encuentran en el sitio web, las mismas que serán cargadas a la tarjeta de crédito. En lo referente al costo de las llamadas y retrollamadas, se estableció una tarifa de 0,03 ctvs el minuto.

Previo a la realización de la llamada se debe elegir la ubicación del llamante, para este propósito se cuenta con tres opciones: casa (por defecto), trabajo y otros. Se inicia la llamada al destinatario dando click sobre una de las tres extensiones disponibles en el contacto. Asterisk hará timbrar primero el teléfono del llamante, una vez que se conteste la llamada, la central hará sonar el teléfono del destinatario, para

finalmente conectarlas directamente; en caso de que el destinatario no conteste, se escuchará un menú el cual indicará si desea dejar un mensaje de voz o utilizar la función de retrollamada, la misma que devolverá la llamada apenas se encuentre libre la línea.

TuxFono cuenta además con la posibilidad de poder revisar su correo de voz con solo un click, la llamada a su casillero de voz se hace automáticamente desde la página web, incluye una alerta por correo electrónico con el mensaje que ha dejado el llamante.

1.4. Metodología

Se inicia con la instalación y configuración de las herramientas necesarias en el equipo que funcionará como central telefónica y servidor web.

Para la centralita se utiliza Asterisk y el motor de base de datos, MySQL, para poder integrar ARA y evitar así la recarga manual de los archivos de configuración. Mientras que en lo referente al diseño y manejo del sitio web, se ha hecho uso de Apache y PHP; incluyendo de igual forma el motor de base de datos para poder gestionar los perfiles de los clientes registrados.

Además incluye la creación de scripts AGI y AMI, para poder enviar ordenes a la central y manejar el plan de marcado con una lógica más avanzada. Todo esto es sobre una máquina virtual usando VirtualBox que alberga Fedora 16 como sistema operativo.

Posteriormente se procederá a hacer las pruebas correspondientes en una red local, en la cual se ha configurado previamente el dominio de www.tuxfono.com en los equipos involucrados, para facilidad de las pruebas.

CAPITULO 2

MARCO TEÓRICO

2.1. Asterisk

Es un sistema basado en software libre que realiza las funciones de una centralita telefónica (PBX) y opera bajo un sistema operativo basado en Linux. Aunque puede trabajar de igual forma sobre otras plataformas como son: BSD, Windows, Open Solaris y Mac OS X. No se trata únicamente de un programa sino de un demonio también, es decir que se ejecuta en segundo plano. Su configuración normalmente se almacena en varios archivos de texto editables de forma tradicional.

Asterisk permite la integración de teléfonos virtuales o Softphones, teléfonos IP, equipos analógicos para poder realizar llamadas internas o externas a la red. Incluso nos brinda la oportunidad de conectarnos a líneas de VoIP, líneas fijas análogas, troncales SIP, líneas móviles GSM, entre otras.

Además de contar con características propias de una centralita clásica se encuentra dotado con algunas características propias de centrales más avanzadas, como son: programación de contestadora automática

con opciones, buzón de voz, teleconferencias, colas de llamadas, registro de las mismas y transferencia automática de llamadas. Su única limitante es el equipo físico, debido a que demandará mayor cantidad de memoria, espacio en disco, al igual que mejor capacidad de procesamiento a medida que se incremente la cantidad de usuarios.

A continuación se enumeran algunas de las funcionalidades más destacadas de esta PBX. [10]

- Transferencia de llamadas, internas y externas.
- Hasta 150 extensiones por cada sistema.
- Operadora Digital (menús interactivos y guiados).
- Gestor de listas negras (números bloqueados).
- Acciones a realizar según horarios y fechas.
- Registro de llamadas entrantes y salientes.
- Enrutamiento de llamadas salientes por parámetros.
- Desvío de llamadas si está ocupado o no contesta.
- Opción No molestar.
- Llamada en espera.
- Incorporar extensiones remotas (teletrabajadores o delegaciones)
- Identificador de llamante (CallerID).
- Grabación para llamadas entrantes y salientes.

- Soporta videoconferencia con protocolos SIP e IAX2.
- Entorno gráfico de configuración y administración vía web.
- Salas de conferencia (dos o más terminales simultáneamente).
- Monitorización de llamadas en curso.
- Buzones de voz (general, individual, por grupos) protegidos.

2.2. SIP

SIP es un protocolo de señalización para VoIP, cuyo principal propósito es establecer, modificar y terminar las sesiones. Se complementa con otros protocolos como RTP y SDP; el primero usado para la transmisión de información como audio y video, y el segundo se encarga de entablar las negociaciones entre los participantes de la sesión describiendo y codificando sus características y capacidades. Esto se traduce en que cada comunicación hace uso de dos canales.

[11]

Toda la lógica de SIP reside en sus terminales, incluyendo el estado de la llamada (excepto el enrutamiento de los mensajes SIP). Esto quiere decir que maneja la teoría de la “nube tonta”.

SIP posee muchas cualidades destacables como su simplicidad al ser usado en el desarrollo de software, extensibilidad al soportar futuras características, escalabilidad al permitir grandes volúmenes de tráfico,

integración al poder ser añadido en otras aplicaciones y trabajar con otros protocolos y compatibilidad al funcionar sobre TCP y UDP. Por defecto los clientes SIP usarán el puerto UDP 5060, si no funciona se intentará en el puerto TCP 5060; aunque puede ser configurado de manera diferente.

2.3. AMI

Como su nombre lo indica es una interfaz que permite a Asterisk interactuar con otros programas, los cuales pueden darle instrucciones a la PBX o recibir notificaciones de algún evento a través de paquetes TCP/IP. Para esta comunicación en especial se usa un protocolo de línea-base, que usa un conjunto de líneas con un formato “clave:valor” que se denomina paquete. El fin de cada línea está determinado por un CR/LF (salto de línea y un retorno de carro), y el final de un paquete es un doble CR/LF.

Antes de poder ejecutar algún comando en Asterisk es necesario que exista una autenticación por parte del programa cliente, luego de la cual existe libre paso de los paquetes, los mismos que deben llevar la etiqueta “Action” en su primera línea si es enviado del programa cliente a Asterisk y “Response” o “Event” si es en la dirección contraria. [12]

A continuación se muestra descripción de cada uno de los paquetes que maneja el AMI para poder mantener comunicación entre el programa cliente y la central telefónica.

- **Action:** Paquete que envía el cliente a Asterisk, para hacer algún requerimiento. Existen algunas opciones dependiendo de los módulos que ha sido cargados en Asterisk. Solo se puede trabajar con una acción a la vez. Este paquete contiene el nombre de la operación a ser realizada, así como sus parámetros.
- **Response:** Es la respuesta de Asterisk hacia el cliente después de cumplir con la acción solicitada.
- **Event:** Los datos relativos a un evento generado desde el interior del núcleo de Asterisk o un módulo de extensión.

2.4. AGI

Esta interfaz permite a otros programas tomar control del plan de marcado de Asterisk por medio de la creación de scripts. El objetivo de los mismos es proporcionar una lógica avanzada, comunicación con bases de datos y acceder a otros recursos externos; de esta forma se puede realizar acciones que de otro modo serían demasiado complicadas o imposibles. [1]

Los scripts AGI pueden relacionarse con Asterisk por medio de los canales de comunicación:

- **STDIN:** su función es la de obtener información desde Asterisk.
- **STDOUT:** se usa para enviar información a Asterisk.
- **STDERR:** se lo utiliza para enviar información de debug hacia la consola de Asterisk; y su uso es opcional. [1]

Inicialmente Asterisk pasa al script un conjunto de variables con sus respectivos valores, este traslado de información culmina cuando la central envía una línea en blanco. Posteriormente AGI toma el control del plan de marcado enviando comandos a Asterisk por medio del STDOUT, cada uno de estas sentencias recibe una respuesta; este proceso durará durante toda la ejecución del script.

2.5. ARA

La configuración de Asterisk se basa en la modificación de ciertos archivos. A pesar de la facilidad que esto proporciona, presenta algunas desventajas; por ejemplo al encontrarse trabajando en un entorno dedicado a las comunicaciones no se puede detener el servicio, para actualizar algunos de los archivos. Esto representaría pérdidas considerables e inconformidad por parte de los usuarios.

Para evitar este problema se cuenta con herramientas como: ARA, la cual cuenta con funciones y drivers implementados actualmente dentro de Asterisk.

ARA logra la integración transparente de la estructura de configuración interna de Asterisk con el lenguaje de consulta estructurado (SQL) usado en las base de datos, permitiendo el mapeo estático de los archivos de configuración o de cualquier entrada especial en tiempo de ejecución que cree dinámicamente objetos, peers, etc. sin la necesidad de recargar algún módulo o servicio. Esto quiere decir que todo va a ser almacenado en una base de datos y puede ser cargado sobre la marcha, permitiendo así añadir nuevos usuarios y modificar los planes de marcado sin detener el servicio.

CAPITULO 3

IMPLEMENTACIÓN

3.1. Introducción

La evolución en el campo tecnológico ha demostrado que las herramientas basadas en software libre llevan una gran ventaja ante las privadas. Esto se debe a su gran desempeño, bajo costo, versatilidad y seguridad, permitiendo así la creación de sistemas de una enorme capacidad y bajo consumo de recursos.

Es por esto que TuxFono ha sido implementado usando como base la distribución *Fedora 16*. Las funciones de PBX las realiza *Asterisk* por su facilidad de instalación y configuración, al presentar sus archivos en directorios bien definidos y contar con una amplia documentación. Mientras que el servidor web es *Apache*, donde se almacena el sitio web de la agenda telefónica en línea.

El hardware que se debe usar dependerá en gran medida de la cantidad de llamadas que deba procesar la central telefónica, para este caso se usó una máquina virtual sobre VirtualBox puesto que se trataba de demostraciones a una escala muy pequeña y la capacidad de procesamiento requerida no tenía grandes demandas.

3.2. Hardware

El equipo sobre el cual se trabajó no es una estación física sino virtual, puesto que durante el desarrollo de TuxFono surgió la necesidad de guardar los diferentes progresos del servidor; ya que en muchas ocasiones se requería instalación de software que podría afectar el desempeño de la central en formas no deseadas. Así se garantizaba el retorno a un punto específico del trabajo de manera rápida y segura para continuar con las pruebas.

Sin duda otro de los beneficios de usar una máquina virtual, es la capacidad de poder transportarla a otro equipo que puede tener mejores recursos de hardware, para así incrementar el desempeño que puede llegar a tener TuxFono. En la Tabla I y II se describen las características del equipo anfitrión y máquina virtual respectivamente, usado en la realización del proyecto.

Marca - Modelo:	HP Compaq 6710b
Memoria RAM:	4GB
Procesador:	Intel Core 2 Duo 2,40GHz
Capacidad Disco Duro:	160 GB
Sistema Operativo - Kernel:	Fedora 16 - Kernel v3.3.0-4
Arquitectura:	x86_64

Tabla I. Características del Equipo Anfitrión

Memoria RAM:	1GB
Procesador:	Intel Core 2 Duo 2,40GHz
Capacidad Disco Duro:	15GB
Sistema Operativo - Kernel:	Fedora 16 - Kernel v3.2.7-1
Arquitectura:	i386

Tabla II. Características de la Máquina Virtual

3.3. Software

Las diferentes herramientas que se han usado para la creación de la central y desarrollo del sitio web se describen a continuación de manera más detallada.

3.3.1. Central Telefónica

El software que se utiliza para hacer las veces de central telefónica o PBX se describen en la Tabla III.

Sistema Operativo:	Fedora 16 - Kernel v3.2.7-1
Software PBX:	Asterisk v1.8.7.1
Software Base de Datos:	MySQL v5.5.20 Community Server (GPL)

Tabla III. Software utilizado en la PBX

En la PBX, el uso de MySQL es indispensable para el manejo del ARA, el cual permite a Asterisk mapear los archivos de configuración necesarios para funcionar, directamente desde una base de datos. Evitando así la necesidad de recargar cada uno de los módulos de manera individual.

3.3.2. Sitio Web

En el desarrollo web de la agenda virtual, se ha estudiado, adecuado y desarrollado con las herramientas descritas en la Tabla IV según las necesidades de TuxFono; tanto en el ámbito de funcionamiento, como de presentación.

Diseño de Páginas Web:	PHP 5.3.10
Software Base de Datos:	MySQL v5.5.20 Community Server (GPL)
Servidor Web:	Apache v2.2.22 (Unix)

Tabla IV. Software utilizado en el Sitio Web

En esta ocasión se muestra nuevamente MySQL, cuya funcionalidad es la de almacenar la información correspondiente a cada uno de los cliente, para así poder manejar cada uno de los perfiles de forma más rápida y precisa.

3.3.3. Softphone

Es el software que se utiliza para simular el funcionamiento de un Teléfono IP a través de la computadora, permitiendo realizar llamadas a otros teléfonos usando un VSP. En TuxFono se ha utilizado los mencionados en la Tabla V.

Softphone Windows:	X – Lite v4.0
Softphone Linux:	Twinkle 1.4.2

Tabla V. Software utilizado como Softphones

3.4. Instalación

Los diferentes paquetes utilizados para el desarrollo de la Central Telefónica y del Sitio Web, fueron obtenidos con el “gestor de paquetes” incluido en la distribución de Fedora 16 y se listan a continuación.

asterisk-18.7.1-1

asterisk-alsa-1.8.7.1-1

asterisk-apidoc-1.8.7.1-1

asterisk-devel-1.8.7.1-1

asterisk-mysql-1.8.7.1-1

asterisk-sounds-core-es-1.5.22-1

asterisk-sounds-core-es-ulaw-1.4.22-1

asterisk-voicemail-1.8..7.1-1

asterisk-voicemail-plain-1.8.7.1-1

php-5.3.10-1

php-mysql-5.3.10-1

mysql-5.5.20-1

mysql-server-5.5.20-1

3.5. Configuración

Posterior a la instalación, se debe realizar la configuración adecuada de los diferentes servicios que utilizará el servidor de TuxFono. Además de la correcta definición de los archivos de configuración de Asterisk

3.5.1. Seguridad

El ámbito de seguridad es sumamente extenso, pero en el desarrollo y pruebas de este sistema se han tomado las mínimas precauciones. Es decir que deshabilita las herramientas que puedan retrasar o detener las acciones ejecutadas por TuxFono, en lugar de establecer reglas adecuadas.

SELinux

En algunas distribuciones de Linux, incluida *Fedora*, viene anexo el SELinux, cuya función es la proveer de una gran variedad de políticas de seguridad. Por facilidad se detiene su ejecución en el servidor, y así continuar con el desarrollo y pruebas de TuxFono con normalidad.

Su archivo de configuración se encuentra ubicado en: ***/etc/selinux***.

Y se debe definir la siguiente regla: `SELINUX = disabled`

IPTables

Este servicio realiza las funciones de un Firewall y se encuentra integrado en el kernel de Linux. Este script contiene diferentes reglas para permitir ingresos o salidas de los paquetes del equipo. Por facilidad se deshabilitan, permitiendo así que los clientes puedan registrarse en el servidor sin ningún inconveniente.

Para esto, se ejecuta el siguiente comando: `service iptables stop`

3.5.2. Servicios Adicionales

Este proyecto no se encuentra basado únicamente en la central telefónica, aunque sea esta su principal característica. Depende de otras herramientas para complementarlo, las cuales se describen a continuación.

MySQL

Para que TuxFono pueda ejecutar las diversas acciones que lleva a cabo con la base de datos, tanto a nivel de la central, como del sitio web. Se debe iniciar el servicio con el siguiente comando: `service mysqld start`

Apache

Es un servidor web de distribución libre y código abierto, que permite el desarrollo de la agenda telefónica en línea y brinda a los clientes la posibilidad de poder visualizarla desde su propio equipo.

Para esto, debemos ubicar el archivo ***httpd.conf***, en la ruta: ***/etc/httpd/conf/httpd.conf*** y modificar las reglas como se muestra a continuación.

```
Listen 80

<Directory "/var/www/html">

Options Indexes FollowSymLinks

AllowOverride All

Order allow,deny Allow from all

</Directory>
```

3.5.3. Archivos de Asterisk:

Existen diversos archivos o ficheros que deben adecuarse para que la central trabaje correctamente bajo las especificaciones dadas; los cuales se encuentran ubicados en: ***/etc/asterisk/***.

Lo principal es la integración del ARA en el proyecto, para lo cual se necesitan realizar las siguientes modificaciones.

Fichero res_config_mysql.conf

Este fichero es usado estrictamente para el funcionamiento de ARA. Aquí se describe los parámetros necesarios para conectarse a la base de datos creada para TuxFono, la estructura del mismo es la siguiente:

```
[tuxfono]
dbhost = localhost
dbname = asterisk
dbuser = root
dbpass = codigo
dbport = 3306
dbsock = /var/lib/mysql/mysql.sock
requirements=createclose
```

Para una descripción completa del significado de los parámetros escritos en este literal, se recomienda revisar el ANEXO A-I.

Fichero extconfig.conf

Archivo que detallada la forma en que se relacionan las tablas de la base con los archivos de configuración de Asterisk, los cuales tienen nombre de familias ya definidos. De esta forma no será necesario recargar los módulos sino que tomará los datos requeridos en tiempo real desde la base.

El fichero contiene de la siguiente definición:

```
[general]
extensions => mysql,tuxfono,extensions
voicemail => mysql,tuxfono,voicemessages
sipusers => mysql, tuxfono,sip_buddies
sippeers => mysql, tuxfono,sip_buddies
```

En el ANEXO A-II se puede encontrar una descripción detallada del significado de las líneas de código descritas con anterioridad.

Fichero modules.conf

Este fichero se lee al iniciar el servicio de Asterisk. Aquí se especifican que módulos se van a cargar durante el arranque de la PBX. Para evitar problemas con los códecs de audio que maneja TuxFono, se recomienda verificar las siguientes líneas.

```
load => codec_ulaw.so
load => codec_gsm.so
```

Fichero sip.conf

Contiene información correspondiente a los usuarios que se conectan a la central usando SIP, por cada usuario debería existir un contexto. En el caso de TuxFono esto se realiza en la base de datos y por consiguiente en el fichero solo quedan parámetros de configuración básica, los cuales pueden ser descritos a detalle en el ANEXO A-III.

```
[general]
context = internos
allowguest = no
udpbindaddr = 0.0.0.0
tcpenable = no
qualify = yes
language = es
rtcachefriends = yes
```

Fichero extensions.conf

Este archivo contiene la parte más importante de la central telefónica, el plan de marcado. Debido al uso del ARA ya no se debe especificar nada dentro de este archivo sino que realiza una redirección, para que lo lea de la base de datos en la tabla *extensions*.

Es importante recalcar que cada contexto que se defina en la base de datos, debe tener su homónimo en este archivo de configuración. Caso contrario Asterisk no podrá encontrar el contexto y no cumplirá con los requerimientos esperados.

Una descripción más explícita de las líneas de código mostradas a continuación se encuentra en el ANEXO A-IV.

```
[general]
autofallthrough=yes
clearglobalvars=no

[internos]
switch => Realtime/@extensions

[macro-retrollamada]
switch => Realtime/@extensions

[macro-extensions]
switch => Realtime/@extensions

[extensions-inverso]
switch => Realtime/@extensions
```

3.5.4. Base de Datos.

Al trabajar con la funcionalidad del ARA y tener que almacenar información correspondiente a los diferentes perfiles de usuarios, este sistema se ve obligado a utilizar una base de datos como parte fundamental.

De esta forma Asterisk cargará y actualizará los objetos, de manera dinámica, según los requiera. Además de permitir al sitio web poder manejar de manera eficiente toda la información de los clientes, como es su información crediticia, personal y su agenda.

El modelo lógico de la base de datos, denominada *asterisk*, desarrollada para TuxFono, es el que se muestra en el Figura B-I.

Tabla: users

Es usada por el sitio web de TuxFono para poder registrar a los clientes, guardando sus datos personales, información crediticia, saldo, usuario y clave. De esta tabla se verifican datos para realizar diversas acciones como: ingresar al sistema, obtención de tiempo para realizar una llamada, mostrar una foto por defecto en el perfil dependiendo de su género y demás. Su descripción se muestra en la Tabla VI.

Campo	Tipo	Descripción
iduser <PK>	VARCHAR(15)	Usuario único que tiene el cliente para acceder al sistema.
pass	VARCHAR(15)	Clave de acceso a su agenda virtual.
nombre	VARCHAR(45)	Nombres completos del cliente.
apellido	VARCHAR(45)	Ambos apellidos del cliente.
mail	VARCHAR(45)	Correo electrónico del cliente
sexo	VARCHAR(1)	Especifica el sexo del cliente.
país	VARCHAR(15)	País donde habita el cliente.
ciudad	VARCHAR(15)	Ciudad de residencia.
foto	VARCHAR(50)	Ruta de la foto para el perfil del usuario.
tarjeta	VARCHAR(10)	Elige entre tres tipos de tarjeta. Visa, MasterCard y American Express.
cod_tarjeta	VARCHAR(15)	Número de la tarjeta de crédito.
fecha_caduca	VARCHAR(15)	Fecha de vencimiento de la tarjeta de crédito.
estado	INT(1)	Indica si la cuenta está activa o no.
saldo	FLOAT	Saldo en dólares que tiene el usuario

Tabla VI. Descripción de la Tabla: users

Tabla: agenda

Es utilizada para guardar la red de contactos de cada uno de los clientes. Se almacena el identificador del dueño y el del contacto que se ha agregado a la agenda, convirtiendo a ambos en claves primarias para evitar así repeticiones. Su esquema se indica en la Tabla VII.

Campo	Tipo	Descripción
iduser <PK><FK>	VARCHAR(15)	Usuario del propietario de la agenda.
idcontacto <PK><FK>	VARCHAR(15)	Usuario del contacto agregado a la agenda

Tabla VII. Descripción de la Tabla: agenda

Tabla: sip_buddies

Guarda relación directa con la central telefónica, debido a que cada registro representa un contexto que debería existir por cada sipbuddy de TuxFono. Según las especificaciones del proyecto, cada usuario dispone de tres sipbuddies, los mismos que son creados añadiendo los números: uno, dos y tres después del nombre del usuario. Es decir que si el usuario es *sorymb*, los sipbuddies corresponden a: *sorymb1*, *sorymb2* y *sorymb3*.

A continuación se describe los campos más relevantes para las acciones ejecutadas por TuxFono en la Tabla VIII, mientras que la tabla completa se encuentra descrita en el ANEXO C-I.

Campo	Tipo	Descripción
idsip_buddies <PK>	INT(11)	SipBuddy asignado al usuario
name	VARCHAR(80)	Nombre del propietario del sipbuddy.
host	VARCHAR(31)	Dirección o host donde se encuentra el dispositivo remoto.
type	ENUM	Puede ser: peer, user o friend.
callerid	VARCHAR(80)	Define el identificador
context	VARCHAR(80)	Indica el contexto asociado en el plan de marcado.
secret	VARCHAR(80)	Contraseña utilizada para autenticación.
disallow	VARCHAR(100)	Permite deshabilitar un códec.
allow	VARCHAR(100)	Permite habilitar un códec.
extensión	VARCHAR(20)	Extensión asignada al sipbuddy
iduser <FK>	VARCHAR(15)	Usuario al que pertenece el sipbuddy

Tabla VIII. Descripción parcial de la Tabla: sip_buddies

Tabla: extensions

Aquí se almacena todo el plan de marcado, incluidos los diferentes contextos y macros que se han definido previamente en el *extensions.conf*. Los campos que la conforman se detallan en la Tabla IX.

Campo	Tipo	Descripción
idextension	INT(11)	Se lo utiliza como referencia para crear las extensiones de los clientes. Se incrementa automáticamente.
context <PK>	VARCHAR(20)	Nombre del contexto o de la macro.
exten <PK>	VARCHAR(20)	Número de la extensión que se marca.
priority <PK>	TINYINT(4)	Orden de ejecución de las instrucciones dentro de una extensión.
app	VARCHAR(20)	Aplicación o acción que se va a ejecutar
appdata	VARCHAR(200)	Parámetros que puede o no requerir la aplicación.

Tabla IX. Descripción de la Tabla: extensions

Estos son los diferentes contextos que se han desarrollado para la Agenda Telefónica en línea:

- **Contexto: Internos (Parte 1)**

En la Figura 3.1 se muestran las tres extensiones que asignan al cliente cuando se registra en el sistema.

context	exten	priority	app	appdata
internos	225	1	Macro	extensions,ringomb1,\${varSIPBuddyOrigen}
internos	226	1	Macro	extensions,ringomb2,\${varSIPBuddyOrigen}
internos	227	1	Macro	extensions,ringomb3,\${varSIPBuddyOrigen}

Figura 3.1. Extensiones por cliente registrado

El número de la extensión se obtiene tomando el último valor del *idextensions* ingresado e incrementándolo. Mientras que la variable *\${varSIPBuddyOrigen}* es el sipbuddy de quien origina la llamada.

- **Macro: Extensions**

En este contexto se encuentra las reglas que se ejecutan para todas las llamadas entre usuarios que se llevan a cabo a través de la central telefónica de TuxFono. La definición de la macro se muestra en la Figura 3.2.

context	exten	priority	app	appdata
macro-extensions	h	1	AGI	setearSaldo.php, \${TIEMPO_DISPONIBLE}, \${arg2}
macro-extensions	s	1	AGI	consultarTiempo.php, \${arg2}
macro-extensions	s	2	Dial	SIP/\${arg1},30,L(\${TIEMPO_DISPONIBLE})
macro-extensions	s	3	Set	SIP_BUDDY_DESTINO=\${arg1}
macro-extensions	s	4	Set	SIP_EXTEN=\${MACRO_EXTEN}
macro-extensions	s	5	Set	SIP_BUDDY_ORIGEN=\${arg2}
macro-extensions	s	6	Background	opciones
macro-extensions	s	7	WaitExten	10
macro-extensions	t	1	Playback	vm-goodbye
macro-extensions	t	2	Hangup	

Figura 3.2. Macro – Extensions

Primero se consulta el tiempo disponible que tiene el usuario para hablar, con esto se evita que al marcar la extensión directamente se hagan llamadas gratuitas. Posteriormente se realiza la llamada, estableciendo el destino, parámetro *\${arg1}*, y el tiempo que tiene disponible para hablar, parámetro *\${TIEMPO_DISPONIBLE}*. De igual forma se definen ciertas variables que pueden ser utilizadas en otros

contextos posteriormente, y se reproduce el menú de opciones para el usuario en caso de que el tiempo de timbrado se sobrepase.

Los registros con extensión 't' se ejecutan cuando no el usuario se excede del tiempo de espera definido. Y finalmente se ejecuta la extensión especial 'h', para reducir el saldo del cliente una vez finalizada la llamada.

- **Contexto: Internos (Parte 2)**

La grabación que especifica las opciones que tiene TuxFono se encuentra en la macro-extensions, pero las opciones que el cliente elija se toman del contexto que llama a la macro (contexto: internos). Al elegir la opción uno, la cual permite dejar un mensaje de voz, se ejecutan las reglas indicadas en la Figura 3.3.

context	exten	priority	app	appdata
internos	1	1	GoTolf	\$_["\${DIALSTATUS}" = "BUSY"]?2:4
internos	1	2	VoiceMail	\$_{SIP_EXTEN}@default,b
internos	1	3	Hangup	
internos	1	4	VoiceMail	\$_{SIP_EXTEN}@default,u
internos	1	5	Hangup	

Figura 3.3. Opción: Mensaje de Voz

En la Figura 3.4 se muestran las acciones a realizar cuando se elige la opción nueve para activar la retrollamada.

context	exten	priority	app	appdata
internos	9	1	Macro	retrollamada,\${SIP_BUDDY_DESTINO},\${SIP_BUDDY_ORIGEN}
internos	9	2	Playback	auth-thankyou
internos	9	3	Hangup	

Figura 3.4. Opción: Retrollamada

En caso de que no elija ninguna opción válida del menú o que el tiempo de espera haya expirado se ejecuta alguna de las reglas indicadas en la Figura 3.5.

context	exten	priority	app	appdata
internos	i	1	Playback	pbx-invalid
internos	i	2	Background	opciones
internos	i	3	WaitExten	5
internos	t	1	Playback	vm-goodbye
internos	t	2	Hangup	

Figura 3.5. Opción: Inválida o Fuera de Tiempo

Finalmente en la Figura 3.6 se muestra una de las opciones extra que guarda este contexto, que es la revisión del buzón de voz. Para ellos se marca a la extensión *282.

context	exten	priority	app	appdata
internos	*282	1	VoiceMailMain	s\${varExten_orig}@default

Figura 3.6. Llamada al Buzón de Voz

Las opciones que se envían como parámetros a la aplicación tienen su función definida. Por ejemplo: la opción “s” sirve para omitir la digitación de clave para acceder al buzón, la variable $\${varExten_orig}$ toma la extensión que origina la llamada para así evitar que el usuario especifique la misma.

- **Macro: Retrollamada**

Esta macro se ejecuta cada vez que los usuarios eligen la opción de retrollamada en el menú del sistema. Es el encargado de crear un archivo con todos los parámetros necesarios, para posteriormente enviarlo a un directorio específico de Asterisk y así ejecutarlo.

Es importante recalcar el uso de aplicación *System*, cuya función es la de dar ordenes al sistema como si estas fueran escritas directamente sobre la consola, es por esto que se usa el comando *echo* para crear una línea y luego escribirla en un archivo.

context	exten	priority	app	appdata
macro-retrollamada	s	1	System	echo Channel:SIP/\${arg1}>>/tmp/callback_\${arg1}.call
macro-retrollamada	s	2	System	echo "CallerId:TuxFono - \${arg2}">>/tmp/callback_\${arg1}.call
macro-retrollamada	s	3	System	echo WaitTime:40>>/tmp/callback_\${arg1}.call
macro-retrollamada	s	4	System	echo Maxretries:3>>/tmp/callback_\${arg1}.call
macro-retrollamada	s	5	System	echo Retrytime:60>>/tmp/callback_\${arg1}.call
macro-retrollamada	s	6	System	echo Context:extensions-inverso>>/tmp/callback_\${arg1}.call
macro-retrollamada	s	7	System	echo Extension:1234>>/tmp/callback_\${arg1}.call
macro-retrollamada	s	8	System	echo Priority:1>>/tmp/callback_\${arg1}.call
macro-retrollamada	s	9	System	echo Setvar:sip_origen=\${arg2}>>/tmp/callback_\${arg1}.call
macro-retrollamada	s	10	System	echo Setvar:sip_destino=\${arg1}>>/tmp/callback_\${arg1}.call
macro-retrollamada	s	11	System	mv /tmp/callback_\${arg1}.call /var/spool/asterisk/outgoing

Figura 3.7. Macro - Retrollamada

El contenido de la Macro: *retrollamada* se muestra en la Figura 3.7, donde se puede observar la definición del tiempo de espera para ejecutar la retrollamada, *WaitTime*, además del número de intentos que se realizarán, *Maxretries*, y el intervalo entre cada uno de ellos, *Retrytime*. Además se establece la identificación del llamante usando el *CallerID*.

La acción a ejecutar es una desviación al contexto: *extensions-inverso* en la extensión *1234* con la prioridad *1*; donde se envía además la variable *sip_origen*, que indica el sipBuddy que origina la retrollamada y el *sip_destino*, que es el usuario a quien se desea contactar.

- **Contexto: Extensions-inverso**

Este contexto se observa en la Figura 3.8 y es utilizado únicamente por la retrollamada. Su función es la de contactar la llamada y realizar el cobro respectivo. Inicialmente se ejecuta un script AGI para consultar el saldo disponible para hablar, luego se establece correctamente el nombre del destinatario de la retrollamada. Finalmente realiza la llamada y utiliza una extensión *h* para asegurarse de realizar la disminución de saldo correspondiente.

context	exten	priority	app	appdata
extensions-inverso	1234	1	AGI	consultarTiempo.php, \${sip_origen}
extensions-inverso	1234	2	Set	CALLERID(name)=\${sip_destino}
extensions-inverso	1234	3	Dial	SIP/\${sip_origen},60,L(\${TIEMPO_DISPONIBLE})
extensions-inverso	h	1	AGI	setearSaldo.php, \${TIEMPO_DISPONIBLE}, \${sip_origen}

Figura 3.8. Contexto: extensions - inverso

Tabla: voicemessages

Contiene un registro, equivalente a un buzón de mensajes, por cada sipbuddy registrado en el sistema. Esta opción incluye una alerta de correo electrónico, y acceso automático sin necesidad de digitar la extensión y clave del buzón. La Tabla X. describe su contenido.

Campo	Tipo	Descripción
uniqueid <PK>	INT(11)	Identificador único de la llamada en curso.
customer_id	VARCHAR(11)	Usado para especificar el "accountcode" del usuario.
context	VARCHAR(50)	Contexto por defecto.
mailbox	VARCHAR(11)	Número de la extensión.
password	VARCHAR(10)	Contraseña para acceder al buzón de voz.
fullname	VARCHAR(150)	Nombre completo del propietario del buzón.
email	VARCHAR(50)	Correo electrónico para el envío de la notificación
pager	VARCHAR(50)	Se lo utilizaba para el uso de beepers.
tz	VARCHAR(10)	Uso horario predefinido.
attach	VARCHAR(4)	Indica si el mensaje de voz se adjunta o no con el correo.
saycid	VARCHAR(4)	Especifica si se escucha o no la extensión de quien dejó el mensaje.
dialout	VARCHAR(10)	Contexto para llamar desde (opción 4)
callback	VARCHAR(10)	Contexto para llamar desde el menú del buzón de voz al remitente del mensaje.
review	VARCHAR(4)	Permite al usuario que deja el mensaje escucharlo antes de enviarlo.
operator	VARCHAR(4)	Permite al llamante presionar 0 antes/durante/después está dejando el mensaje para buscar una operadora.
envelope	VARCHAR(4)	Especifica si antes de reproducir el mensaje de voz reproduce los datos del mensaje o no.
sayduration	VARCHAR(4)	Anuncia o no la duración del correo de voz.
sendvoicemail	VARCHAR(4)	Permite o no enviar un correo de voz a otra extensión.
delete	VARCHAR(4)	Indica si se debe o no borrar el mensaje del servidor una vez notificada su llegada.
nextaftercmd	VARCHAR(4)	Indica si se activa o no el desplazamiento entre mensaje con las teclas 7 o 9.
forcename	VARCHAR(4)	Obliga al usuario a grabar su nombre la primera vez.
forcegreetings	VARCHAR(4)	Obliga al usuario a grabar un mensaje de bienvenida.
hidefromdir	VARCHAR(4)	Indica si el usuario aparece o no en el directorio.
stamp	TIMESTAMP	Fecha y Hora actual de la creación del buzón.
fromstring	VARCHAR(50)	Nombre del remitente del e-mail.
emailsubject	VARCHAR(100)	Título del mensaje que se enviará por e-mail.
emailbody	VARCHAR(300)	Cuerpo del mensaje que se enviará por e-mail.

Tabla X. Descripción de la Tabla: voicemessages

Tabla: cdr

Contiene un registro único, por cada una de las llamadas realizadas usando la PBX de TuxFono. Se la utiliza para poder obtener la duración de la llamada. y establecer el precio de la misma. Su descripción se aprecia en la Tabla XI.

Campo	Tipo	Descripción
calldate <PK>	DATETIME	Fecha/Tiempo en que se realizó la llamada.
clid	VARCHAR(80)	Caller ID con texto
src	VARCHAR(80)	Caller ID del usuario que origina la llamada.
dst <PK>	VARCHAR(80)	Extensión de destino.
dcontext	VARCHAR(80)	Contexto destino.
channel	VARCHAR(80)	Canal usado.
dstchannel	VARCHAR(80)	Canal de destino, si es apropiado.
lastapp	VARCHAR(80)	Última aplicación, si es apropiado.
lasdata	VARCHAR(80)	Último parámetro de la aplicación, si es apropiado.
duration	INT(11)	Tiempo total en segundos desde que se marco el número hasta cerrar la llamada.
billsec	INT(11)	Tiempo total en segundos desde que se contesto la llamada hasta cerrar la misma.
disposition	VARCHAR(45)	Estado de la llamada. ANSWER, NOT ANSWER, etc.
amaflags	INT(11)	Indica que banderas debe usar. DOCUMENTATION, BILL, etc.
accountcode <PK>	VARCHAR(20)	Número de cuenta a usar.
userfield	VARCHAR(255)	Campo definido por el Usuario.
uniqueid	VARCHAR(32)	Identificador de canal único.

Tabla XI. Descripción de la Tabla: cdr

3.5.5. Scripts.

Para poder realizar acciones que requieren una lógica se utilizan programas externos para controlar el plan de marcado o la central en sí. En esa ocasión se ha basado el desarrollo de los scripts en lenguaje PHP.

Scripts AGI

Archivo: consultarTiempo.php

Ubicación: /usr/share/asterisk/agi-bin/

Descripción: Recibe como variable el usuario SIP, *sipbuddy*, que originó la llamada y obtiene el valor de la variable *TIEMPO_DISPONIBLE*, la cual indica el tiempo que tiene el usuario para realizar su llamada y retorna este valor al contexto quien ejecuta el script. El código correspondiente al script antes mencionado se ilustra en la Figura 3.9.

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

include_once("/var/www/html/tesis/objetos/saldo.class.php");
include_once("phpagi.php");
$agi = new AGI();
$con = new Saldo();
$sipbuddy = $argv[1];
$idUsuario = substr($sipbuddy,0,-1);
$timeout = '100';
$saldo = $con->obtenerSaldo($idUsuario);
$tiempoDisponible = ( $saldo * 100 * 20 * 1000);
$agi->set_variable("TIEMPO_DISPONIBLE",$tiempoDisponible);

?>
```

Figura 3.9. Script consultarTiempo

Archivo: setearSaldo.php

Ubicación: /usr/share/asterisk/agi-bin/

Descripción: En la Figura 3.10, se observa que recibe como variables el tiempo disponible y el usuario SIP que originó la llamada, con finalidad de realizar el cobro y hacer la disminución del saldo en la cuenta del usuario. Todo esto se ejecuta una vez terminada la misma, con ayuda de la extensión especial *h*, como se mencionó anteriormente.

```
#!/usr/bin/php -q
<?php
include_once("/var/www/html/tesis/objetos/saldo.class.php");
include_once("phpagi.php");
$con = new Saldo();
$agi = new AGI();

$tiempoDisponible = $argv[1];
$sipbuddy = $argv[2];

$idUsuario = substr($sipbuddy,0,-1);
$nuevoSaldo = $con->reducirSaldo($tiempoDisponible,$idUsuario);
?>
```

Figura 3.10. Script setearSaldo

Scripts AMI

Archivo: realizarLlamada.php

Ubicación: /var/www/html

Descripción: Realiza la llamada desde la página web con un solo click sobre la ubicación del destinatario. Recibe variables como *exten_destino*, que contiene el número de la extensión a la cual se desea llamar, y *sipbuddy_origen*, que corresponde al usuario SIP que originó la llamada. Para finalmente enviar este último valor de regreso al plan de marcado y poder realizar otras acciones pertinentes, como se indica en la Figura B-II.

CAPITULO 4

FUNCIONAMIENTO Y PRUEBAS

Una vez concluida las diferentes configuraciones e instalaciones necesarias para el desarrollo del servidor de TuxFono, se debe preparar el inicio del sistema. Para posteriormente realizar pruebas definitivas y constatar que la realización de llamadas funciona adecuadamente, además de poder acceder sin problema a los servicios de retrollamada y buzón que ofrece el proyecto.

4.1. Iniciando servicios

Los servicios que se necesitan ejecutar son *mysql*, *apache* y *asterisk* que tienen los nombres *mysqld*, *httpd* y *asterisk* respectivamente; mientras que el servicio que necesita ser detenido es el de *iptables*. Los siguientes comandos permiten controlar el estado de los demonios y deben ser ejecutados como root:

<code>service [nombre_demonio] start</code>	-	Inicia el servicio
<code>service [nombre_demonio] status</code>	-	Muestra el estado del servicio
<code>service [nombre_demonio] stop</code>	-	Detiene el servicio

Entonces las órdenes que se dan a la terminal con sus respectivas respuestas son:

```
# service iptables stop
    Redirecting to /bin/systemctl stop iptables.service
# service mysqld start
    Redirecting to /bin/systemctl start mysqld.service
# service httpd start
    Starting httpd (via systemctl): [ OK ]
# service asterisk start
    Redirecting to /bin/systemctl start asterisk.service
```

4.2. Guía del Usuario

Normalmente para ingresar a un sitio web se escribe el nombre del dominio, que luego es traducido a una dirección IP pública que es localizada en uno de los servidores que está conectado a Internet.

Para simular este proceso se configuró el VirtualHost de Apache y algunos equipos en una red local para que puedan acceder a la página a través de: www.tuxfono.com, en lugar de usando la dirección IP de la máquina virtual.

La página de inicio de la agenda telefónica en línea se puede observar en la Figura 4.1.

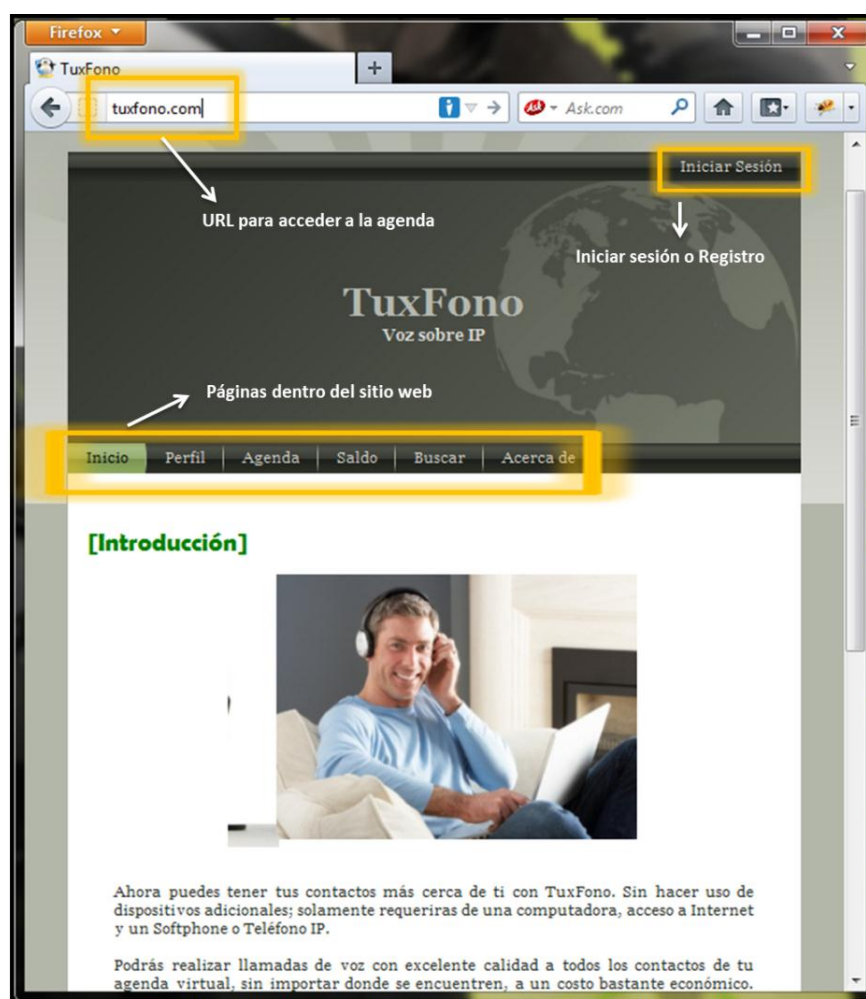


Figura 11. Página Principal

4.2.1.Registro del Cliente

Para poder realizar llamadas y hacer uso de los servicios incluidos en la agenda telefónica virtual, el cliente debe estar registrado en el sistema; para dicho propósito existen dos opciones.

En la Figura 4.2 se pueden observar ambas opciones en conjunto para facilidad de entendimiento del lector de esta guía.

La opción principal se encuentra en el “Inicio de sesión”; mientras que la segunda esta ubicada en la parte inferior de la introducción mostrada en la página principal como un enlace en la palabra “Únete”. Ambas opciones se encuentran resaltadas en la figura antes mencionada.



Figura 12. Enlace de Registro

Posteriormente el navegador redirrecciona al cliente a una nueva página donde tendrá que completar los campos de un formulario con información personal, crediticia y claves de acceso; como se observa en la Figura 4.3. El usuario elegido por el cliente es único dentro del sistema y será su nuevo nombre dentro del mismo.

TuxFono
Voz sobre IP
Inicio Perfil Agenda Saldo Buscar Acerca de




Iniciar Sesión

[Registro de Clientes:]

Datos Personales

Nombres:
Apellidos:
Mail:
Sexo: Hombre Mujer
Pais:
Ciudad:

Información de Pago

Forma de Pago:   
Número de Tarjeta:
Caducidad:

Datos del Usuario

Usuario:
Contraseña:
Confirmar Contraseña:

Enviar Limpiar

Figura 13. Página de Registro

Una vez culminado el registro, el navegador lleva al usuario a la página de “agenda”, la cual se encuentra en blanco porque es nuevo en el sistema y no ha agregado a nadie aún. Aquí encontrará diversas opciones, como son: revisión de su buzón de voz, posibilidad de agregar o quitar contactos a su agenda y finalmente la opción de elegir su ubicación para realizar llamadas, como se indica en la Figura 4.4.

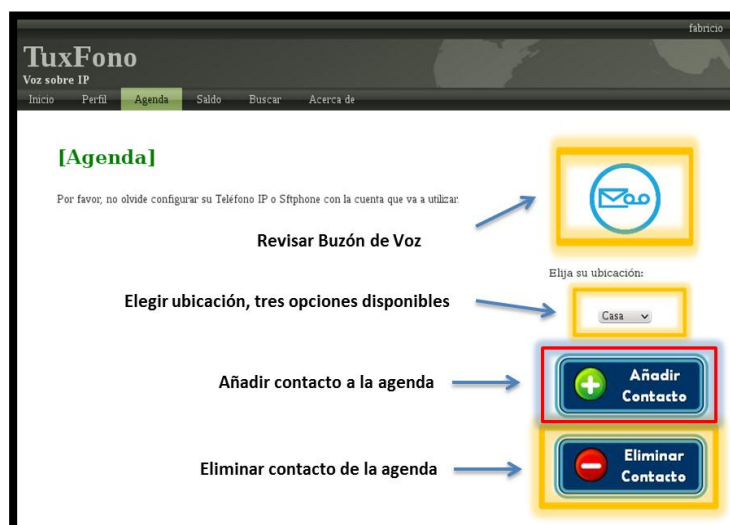


Figura 14. Agenda del Usuario

4.2.2. Agregar Contactos

En la Figura 4.5 se ilustra lo que ocurre al seleccionar la opción de “Añadir un contacto” a la agenda, esto redirige a la página de búsqueda del sistema. Se digita el nombre del cliente y a continuación se desplegará una lista de coincidencias. Al elegir uno de ellos, se muestra su información y la opción de “Añadir”, como en la Figura 4.6.



Figura 15. Búsqueda de Contactos



Figura 16. Resultado de la búsqueda de usuarios

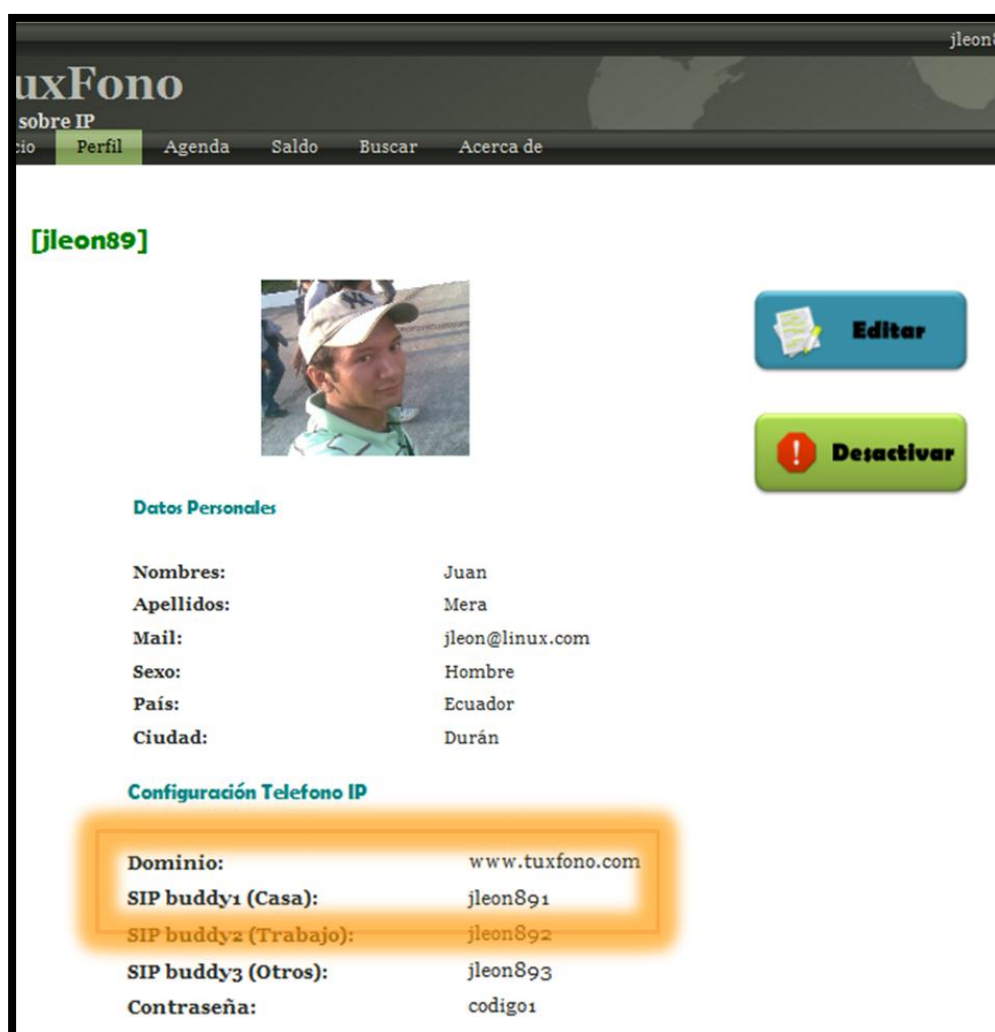
Cuando se agrega el usuario a la agenda, se muestra algo similar a la Figura 4.7.



Figura 17. Contacto agregado en la agenda

4.2.3. Configuración de Softphone

Para poder obtener los datos de configuración necesarios para el Softphone o teléfono IP, el usuario debe dirigirse a la pestaña de “Perfil” y ubicarse en la parte de “Configuración de Teléfono IP”, como se indica en la Figura 4.8.



The screenshot shows the user profile page for 'jleon89' on the TuxFono website. The page is titled 'TuxFono sobre IP' and has a navigation menu with 'Perfil' selected. The user's name is [jleon89] and there is a profile picture of a man wearing a cap. To the right of the picture are 'Editar' and 'Desactivar' buttons. Below the picture is the 'Datos Personales' section with the following information:

Nombres:	Juan
Apellidos:	Mera
Mail:	jleon@linux.com
Sexo:	Hombre
País:	Ecuador
Ciudad:	Durán

Below this is the 'Configuración Telefono IP' section, which is highlighted with an orange box. It contains the following information:

Dominio:	www.tuxfono.com
SIP buddy1 (Casa):	jleon891
SIP buddy2 (Trabajo):	jleon892
SIP buddy3 (Otros):	jleon893
Contraseña:	codigo1

Figura 18. Perfil del Usuario - Info. Teléfono

Es importante mencionar que el *sipbuddy* que elija para configurar su teléfono coincida con el seleccionado en la página de la agenda, previo a la realización de la llamada. De no realizarse correctamente este paso, no se establecerá la misma correctamente.

X – Lite v4.0

Cuando se trabaja con sistemas operativos Windows existen diversas opciones en lo que respecta a Softphones. Para la realización de pruebas se decidió utilizar el X-Lite. Su configuración se realiza en la pestaña “Softphone” - “Account Settings” y posteriormente se mostrará una ventana como en la Figura 4.9, donde se ingresarán los datos destacados en la Figura 4.8.

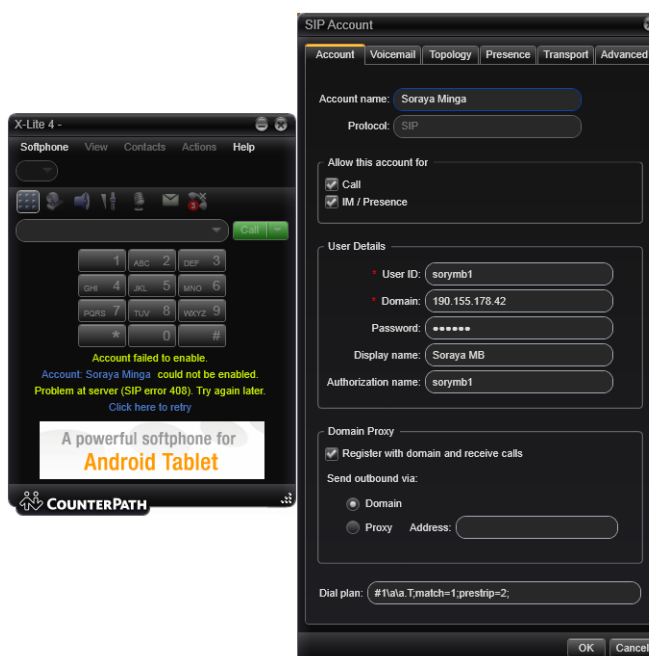


Figura 19. Configuración X-Lite v4.0

Twinkle

Para las pruebas en un sistema operativo basado en Linux se ha utilizado el Twinkle, el cual soporta el protocolo SIP utilizado por TuxFono. Al ejecutar la aplicación por primera vez, el programa solicita al usuario ingresar un nombre para la cuenta y los datos correspondientes para poder conectarse con el VSP, como se muestra en la Figura 4.10 a continuación.

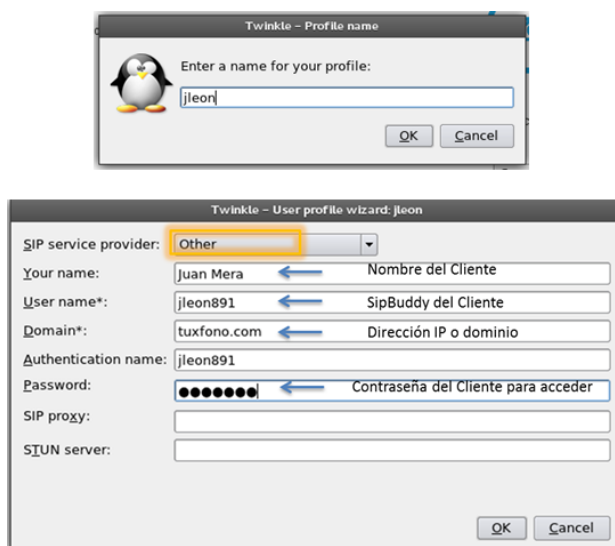


Figura 20. Configuración Twinkle

Posteriormente cuando la cuenta ha sido configurada, aparecerá una ventana como la indicada en la Figura 4.11, donde debe ser seleccionada como cuenta por defecto. Finalmente se puede observar que el registro ha sido exitoso, si obtenemos un mensaje similar al de la Figura 4.12.

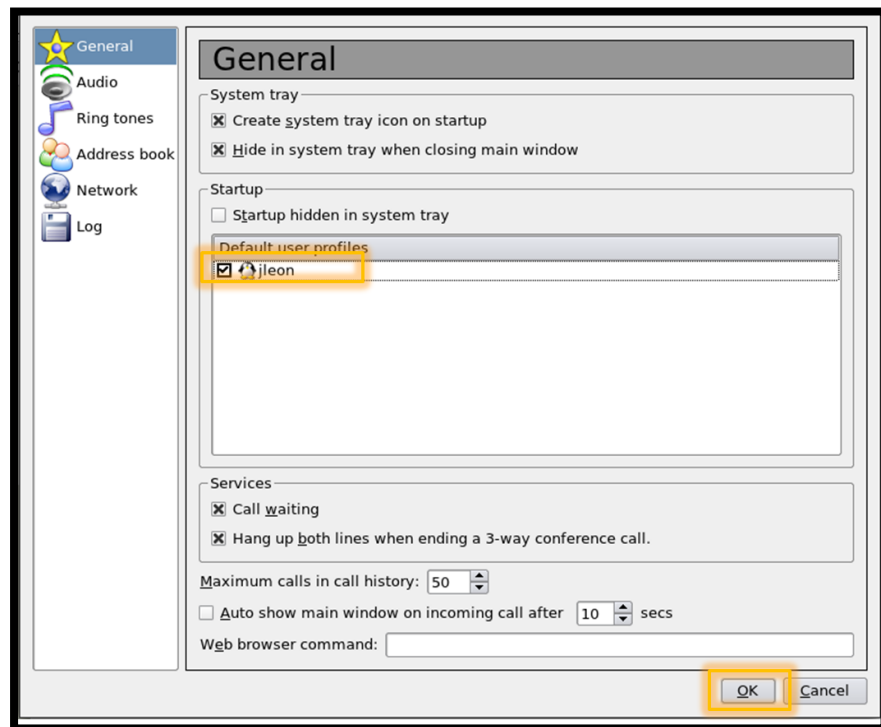


Figura 21. Twinkle - Establecer cuenta por defecto



Figura 22. Twinkle - Correctamente registrado

4.2.4. Compra de Saldo

Previo a la realización de la llamada es necesario adquirir las tarjetas virtuales que equivalen a saldo disponible para hablar. En la página de “saldo” indicada en la Figura 4.13, se muestra el saldo disponible que tiene el usuario en su cuenta en ese momento para realizar llamadas y en la parte inferior las diferentes opciones de compra.

TuxFono
Voz sobre IP
Inicio Perfil Agenda **Saldo** Buscar Acerca de

[Saldo]

Para poder realizar llamadas, debe tener saldo disponible en su cuenta; recordando que las mismas tienen un valor de \$0,01 el minuto.

Saldo Disponible:

[Tarjetas Virtuales]

Adquiere una de nuestras tarjetas virtuales y sigue hablando con tus amigos. El saldo que adquieras, no requiere de ningún código de activación. Se descontará el monto elegido de tu tarjeta.

- Tarjeta de \$5,00 
- Tarjeta de \$10,00 
- Tarjeta de \$20,00 
- Tarjeta de \$30,00 
- Tarjeta de \$50,00 

Figura 23. Página de Saldo

4.2.5. Realización de la llamada

Lo primero que debe hacer el usuario es dirigirse a su agenda y establecer correctamente su ubicación. En la Figura 4.14, se puede observar que se ha elegido el primer sipbuddy, que corresponde a la extensión 'casa'; mismo que se usó para la configuración del Softphone en la Figura 4.9.

Elija su ubicación:

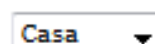


Figura 24. Elegir ubicación del llamante

Posteriormente se selecciona una de las tres extensiones que tiene el contacto para realizar la llamada, como se indica en la Figura 4.15.

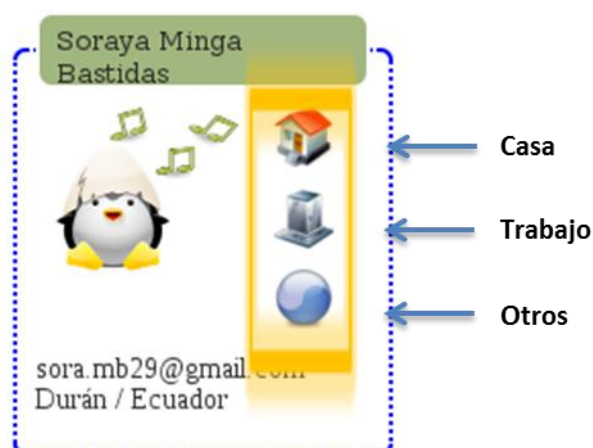


Figura 25. Contacto

El sistema se encuentra programado para que primero suene el teléfono del llamante. Una vez que contesta la llamada, la central hace sonar el teléfono del destinatario, que ya se encuentra previamente configurado con los datos respectivos. Finalmente cuando el destinatario conteste el teléfono, la PBX se encarga de puentear o conectar ambas llamadas directamente.

4.2.6. Configuraciones opcionales

Además de las opciones especificadas para este proyecto, como son la llamada directa y retollamada. El sistema cuenta con algunas características adicionales que se describen a continuación.

Modificar Perfil

TuxFono permite al usuario editar los campos de su perfil, como son su nombre, ubicación, email, etc. Entre esas modificaciones se encuentra la opción de agregar una foto, para que así su red de contactos pueda localizarlo más rápido.

Para esto debe dirigirse a la página de 'Perfil' y presionar el botón de 'Editar'. La página mostrada es similar a la indicada en la Figura 4.16.

[Modificar Perfil]

Datos Personales

Nombres:

Apellidos:




Mail:

Sexo: Hombre Mujer

País:

Ciudad:

Información de Pago

Forma de Pago:   

Número de Tarjeta:

Caducidad:


Datos del Usuario

Usuario:

Contraseña:

Confirmar Contraseña:

(Opcional) ¿Deseas agregar una foto a tu Perfil?



Nota: Primero debes ENVIAR tus datos!!!

Figura 26. Página para editar Perfil

La flecha que se encuentra en la parte inferior del formulario lleva al usuario a otra página, la cual se muestra en la Figura 4.17, donde podrá buscar la foto, que desee colocar, en su computadora. Es importante mencionar que esta opción solamente estará habilitada una vez concluido el registro inicial al sistema, más no durante el mismo.

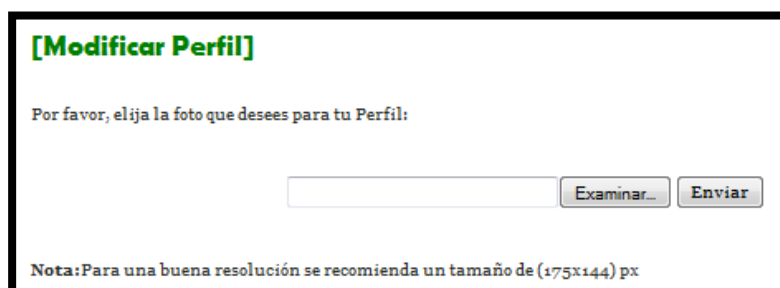


Figura 27. Elegir foto del Perfil

Desactivar Cuenta

En la Figura 4.18 se resalta el botón con el cual el usuario puede dar de baja su cuenta cuando lo considere conveniente. Los datos del mismo no se eliminan, solamente se deshabilita la posibilidad de ingresar al sistema.

Para cambiar el estado de su cuenta nuevamente debe comunicarse con los encargados de TuxFono cuyos datos se encuentran en la página “Acerca De”.



Figura 28. Desactivar cuenta

Buzón de Voz

El usuario tiene la posibilidad de revisar su buzón de mensajes de voz con un solo click. Todo esto desde su agenda, gracias al icono mostrado en la Figura 4.19. Es importante recordar que se escuchará el buzón del sipbuddy que se encuentre seleccionado en la página en ese momento.



Figura 29. Botón del Buzón de Voz

Eliminar Contactos

Como en toda agenda, se puede tanto agregar como quitar contactos. Para ello tenemos el botón 'Eliminar' ilustrado en la Figura 4.20. Previo a la acción del botón se eligen el o los usuarios a eliminar.



Figura 30. Eliminar Contactos

4.3. Funcionamiento

Una vez que ambos usuarios tienen configurados los teléfonos IP adecuadamente, como se indicó previamente en la guía del usuario. Se da paso a la realización de pruebas, para comprobar que TuxFono cumple con los objetivos planteados inicialmente.

A manera demostrativa se ha utilizado a *jllmera89*, como el llamante, y a *sorymb*, como el destinatario.

4.3.1. Llamada

Inicialmente el llamante debe establecer correctamente su ubicación, luego seleccionar la extensión del contacto que desea llamar; en el ejemplo se ha elegido la primera opción: casa, como se indica en la Figura 4.21.



Figura 31. Elección de la extensión: casa

El usuario que realiza la llamada, visualiza en su Softphone algo similar a lo ilustrado en la Figura 4.22.

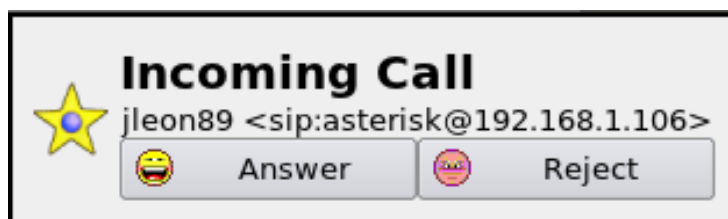


Figura 32. Softphone del Llamante

Una vez contestada esta llamada, el teléfono del destinatario comenzará a sonar; indicando el nombre del llamante original. En la Figura 4.23 se muestra el momento que suena el teléfono y se establece la llamada.



Figura 33. Llamada en el Softphone del destinatario

Cuando el llamante da click sobre una de las extensiones del destinatario, en la página web se visualiza un mensaje similar al de la Figura 4.24, donde se indica el nombre completo del cliente al que se llama.

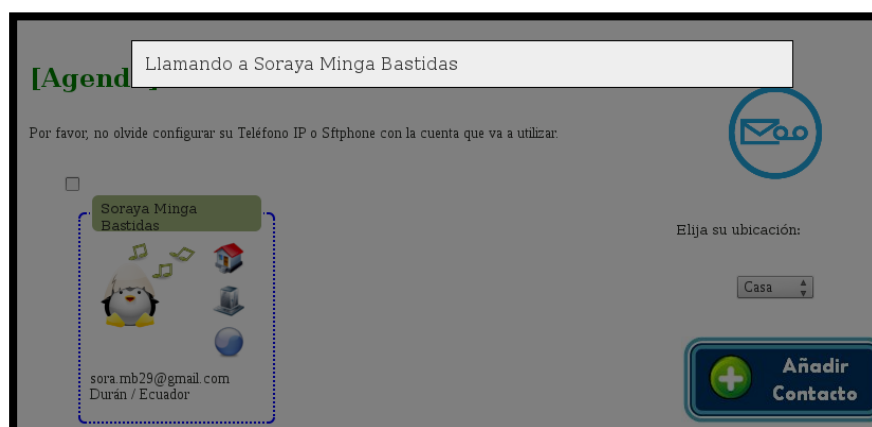


Figura 34. Llamado al contacto

En segundo plano podemos observar en la Figura 4.25, que es lo que ocurre en el servidor de la central telefónica de Asterisk.

```
server*CLI>
== Manager 'tuxfono' logged on from 127.0.0.1
== Using SIP RTP CoS mark 5
> Channel SIP/jleon891-0000000f was answered.
-- Executing [331@internos:1] Macro("SIP/jleon891-0000000f", "extensions,sorymb1,17945000,jleon891")
-- Executing [s@macro-extensions:1] Dial("SIP/jleon891-0000000f", "SIP/sorymb1,10,L(17945000)")
-- Setting call duration limit to 17945.000 seconds.
== Using SIP RTP CoS mark 5
-- Called SIP/sorymb1
-- SIP/sorymb1-00000010 is ringing
-- SIP/sorymb1-00000010 answered SIP/jleon891-0000000f
-- Remotely bridging SIP/jleon891-0000000f and SIP/sorymb1-00000010
server*CLI>
```

Figura 35. Realización de la llamada en la central

Aquí sobresale la llamada a la Macro: *extensions*, la cual esta asignada para todos los *sipbuddies* en TuxFono. Posteriormente la ejecución de la aplicación *Dial*, la cual realiza la llamada y especifica el tiempo total

que dispone para la misma en el parámetro “L” y finalmente la contestación y puenteo de la misma.

Cuando la llamada finaliza en la pantalla del navegador se pueden observar una imagen similar a la Figura 4.26, donde se indica la duración de la misma y el consumo de saldo del cliente. Mientras que el proceso de cobro que se realiza internamente es ilustrado en la Figura 4.27.

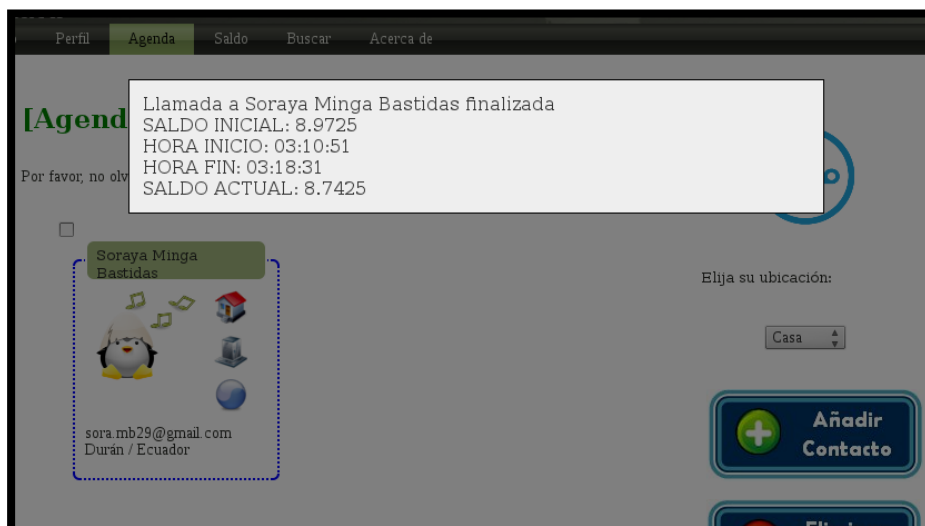


Figura 36. Llamada finalizada

```
server*CLI>
== Spawn extension (macro-extensions, s, 1) exited non-zero on 'SIP/jleon891-0000000f' in macro 'extensions'
== Spawn extension (internos, 331, 1) exited non-zero on 'SIP/jleon891-0000000f'
== Manager 'tuxfono' logged off from 127.0.0.1
server*CLI>
```

Figura 37. Llamada finalizada en la central

4.3.2. Menú: Buzón de Mensajes.

Si el destinatario no se encuentra disponible o la extensión se encuentra ocupada, en la central se podrá observar un comportamiento similar al de la Figura 4.28, donde claramente se observa que la llamada nunca es contestada.

```
server*CLI>
== Manager 'tuxfono' logged on from 127.0.0.1
== Using SIP RTP CoS mark 5
> Channel SIP/jleon891-00000013 was answered.
-- Executing [331@internos:1] Macro("SIP/jleon891-00000013", "extensions,sorymb1,17284000,jleon891")
-- Executing [s@macro-extensions:1] Dial("SIP/jleon891-00000013", "SIP/sorymb1,10,L(17284000)")
-- Setting call duration limit to 17284.000 seconds.
== Using SIP RTP CoS mark 5
-- Called SIP/sorymb1
-- SIP/sorymb1-00000014 is ringing
-- Nobody picked up in 10000 ms
```

Figura 38. Llamada no contestada

En estos casos TuxFono establece el valor de tres variables, que pueden ser usadas posteriormente en otros contextos, como son: SIP_BUDDY_DESTINO, SIP_EXTEN y SIP_BUDDY_ORIGEN. Posteriormente se reproduce el menú de opciones para el usuario y espera por un lapso de diez segundos a que el usuario digite algún número, como se indica en la Figura 4.29.

```
server*CLI>
-- Executing [s@macro-extensions:2] Set("SIP/jleon891-00000013", "SIP_BUDDY_DESTINO=sorymb1")
-- Executing [s@macro-extensions:3] Set("SIP/jleon891-00000013", "SIP_EXTEN=331")
-- Executing [s@macro-extensions:4] Set("SIP/jleon891-00000013", "SIP_BUDDY_ORIGEN=jleon891")
-- Executing [s@macro-extensions:5] Background("SIP/jleon891-00000013", "opciones")
-- <SIP/jleon891-00000013> Playing 'opciones.gsm' (language 'es')
== Manager 'tuxfono' logged off from 127.0.0.1
server*CLI>
-- Executing [s@macro-extensions:6] WaitExten("SIP/jleon891-00000013", "10")
server*CLI>
```

Figura 39. Reproducción de menú

En la Figura 4.30 se muestra lo que ocurre cuando el usuario desea dejar un mensaje de voz al destinatario, utilizando la opción uno del menú.

```
server*CLI>
== CDR updated on SIP/jleon891-00000013
-- Executing [1@internos:1] GoToIf("SIP/jleon891-00000013", "072:4")
-- Goto (internos,1,4)
-- Executing [1@internos:4] VoiceMail("SIP/jleon891-00000013", "331@default,u")
-- <SIP/jleon891-00000013> Playing 'vm-theperson.ulaw' (language 'es')
-- <SIP/jleon891-00000013> Playing 'digits/3.ulaw' (language 'es')
-- <SIP/jleon891-00000013> Playing 'digits/3.ulaw' (language 'es')
-- <SIP/jleon891-00000013> Playing 'digits/1.ulaw' (language 'es')
-- <SIP/jleon891-00000013> Playing 'vm-isunavail.ulaw' (language 'es')
-- <SIP/jleon891-00000013> Playing 'vm-intro.ulaw' (language 'es')
-- <SIP/jleon891-00000013> Playing 'beep.ulaw' (language 'es')
-- Recording the message
-- x=0, open writing: /var/spool/asterisk/voicemail/default/331/tmp/U7byk9 format: wav, 0xb730e908
-- User ended message by pressing #
-- <SIP/jleon891-00000013> Playing 'auth-thankyou.ulaw' (language 'es')
== Parsing '/var/spool/asterisk/voicemail/default/331/INBOX/msg0001.txt': == Found
== Parsing '/var/spool/asterisk/voicemail/default/331/INBOX/msg0001.txt': == Found
== Parsing '/var/spool/asterisk/voicemail/default/331/INBOX/msg0001.txt': == Found
-- Executing [1@internos:5] Hangup("SIP/jleon891-00000013", "")
== Spawn extension (internos, 1, 5) exited non-zero on 'SIP/jleon891-00000013'
server*CLI>
```

Figura 40. Dejar mensaje de voz

La aplicación de *VoiceMail* reproduce el número de la extensión, el mensaje según el estado de la línea (no disponible u ocupada), graba el mensaje después del tono y termina la llamada.

4.3.3. Menú: Retrollamada

En caso de que el llamante no desee dejar un mensaje de voz sino activar la opción de retrollamada, este debe elegir la opción nueve del menú. Es decir que el proceso que ocurre hasta este punto es el mismo que en las Figuras 4.28 y 4.29.

En la Figura 4.31 se observa como se ejecuta la Macro: *retrollamada*, la cual se encarga de crear un archivo con los valores necesarios para su ejecución y posteriormente trasladarlo a una nueva ubicación, donde la central lo ejecutará automáticamente.

```
-- Executing [9@internos:1] Macro("SIP/jleon891-0000001f", "retrollamada,sorymb1,jleon891")
-- Executing [s@macro-retrollamada:1] System("SIP/jleon891-0000001f", "echo Channel:SIP/sorymb1>>/tmp/callback_sorymb1.call")
-- Executing [s@macro-retrollamada:2] System("SIP/jleon891-0000001f", "echo "CallerId:TuxFono - jleon891">>/tmp/callback_sorymb1.call")
-- Executing [s@macro-retrollamada:3] System("SIP/jleon891-0000001f", "echo WaitTime:40>>/tmp/callback_sorymb1.call")
-- Executing [s@macro-retrollamada:4] System("SIP/jleon891-0000001f", "echo Maxretries:3>>/tmp/callback_sorymb1.call")
-- Executing [s@macro-retrollamada:5] System("SIP/jleon891-0000001f", "echo Retrytime:60>>/tmp/callback_sorymb1.call")
-- Executing [s@macro-retrollamada:6] System("SIP/jleon891-0000001f", "echo Context:extensions-inverso>>/tmp/callback_sorymb1.call")
-- Executing [s@macro-retrollamada:7] System("SIP/jleon891-0000001f", "echo Extension:1234>>/tmp/callback_sorymb1.call")
-- Executing [s@macro-retrollamada:8] System("SIP/jleon891-0000001f", "echo Priority:1>>/tmp/callback_sorymb1.call")
-- Executing [s@macro-retrollamada:9] System("SIP/jleon891-0000001f", "echo Setvar:sip_origen=jleon891>>/tmp/callback_sorymb1.call")
-- Executing [s@macro-retrollamada:10] System("SIP/jleon891-0000001f", "echo Setvar:sip_destino=sorymb1>>/tmp/callback_sorymb1.call")
-- Executing [s@macro-retrollamada:11] System("SIP/jleon891-0000001f", "mv /tmp/callback_sorymb1.call /var/spool/asterisk/outgoing")
-- Attempting call on SIP/sorymb1 for 1234@extensions-inverso:1 (Retry 1)
Using SIP RTP CoS mark 5
-- Executing [9@internos:2] Playback("SIP/jleon891-0000001f", "auth-thankyou")
-- <SIP/jleon891-0000001f> Playing 'auth-thankyou.ulaw' (Language 'es')
-- Executing [9@internos:3] Hangup("SIP/jleon891-0000001f", "")
```

Figura 41. Ejecución macro-retrollamada

El archivo generado por la retrollamada se encarga de pasarla al contexto: *extensions-inverso*, donde el script AGI: *consultarTempo* se encargarán de obtener el saldo disponible para realizar la retrollamada y ejecutarla, como se muestran en la Figura 4.32.

```
server*CLI>
> Channel SIP/sorymb1-0000000d was answered.
-- Executing [1234@llamada:1] AGI("SIP/sorymb1-0000000d", "consultarTiempo.php,jleon891")
-- Launched AGI Script /usr/share/asterisk/agi-bin/consultarTiempo.php
-- <SIP/sorymb1-0000000d>AGI Script consultarTiempo.php completed, returning 0
-- Executing [1234@llamada:2] Dial("SIP/sorymb1-0000000d", "SIP/jleon891,60,L(17345000)")
-- Setting call duration limit to 17345.000 seconds.
== Using SIP RTP CoS mark 5
-- Called SIP/jleon891
-- SIP/jleon891-0000000e is ringing
-- SIP/jleon891-0000000e is ringing
server*CLI>
```

Figura 42. Ejecución contexto: extensions-inverso

Finalmente en la Figura 4.33, se aprecia la forma en que el sistema realiza el cobro de la retrollamada, en caso de que esta haya sido contestada.

```
server*CLI>
-- SIP/jleon891-0000000e answered SIP/sorymb1-0000000d
-- Remotely bridging SIP/sorymb1-0000000d and SIP/jleon891-0000000e
[Apr 5 05:21:09] DEBUG[6168]: pbx_spool.c:432 scan_service: Delaying retry since we're currently running '/var/spool/asterisk/outgoing/callback_sorymb1'
[Apr 5 05:21:29] DEBUG[6168]: pbx_spool.c:432 scan_service: Delaying retry since we're currently running '/var/spool/asterisk/outgoing/callback_sorymb1'
-- Executing [?@llamada:1] AGI("SIP/sorymb1-0000000d", "saldo.php,17345000,jleon891")
-- Launched AGI Script /usr/share/asterisk/agi-bin/saldo.php
-- <SIP/sorymb1-0000000d>AGI Script saldo.php completed, returning 0
== Spawn extension (llamada, 1234, 2) exited non-zero on 'SIP/sorymb1-0000000d'
[Apr 5 05:21:34] NOTICE[6300]: pbx_spool.c:366 attempt_thread: Call completed to SIP/sorymb1
server*CLI>
```

Figura 43. Retrollamada contestada

4.3.4.Llamada al Buzón de Voz

Como se indico previamente, el usuario puede acceder a su buzón de mensajes a través de un solo click sobre el icono que se encuentra en su agenda, ilustrado en la Figura 3.19. Mientras que internamente la central presenta un comportamiento similar al de la Figura 4.34

```
server*CLI>
== Manager 'tuxfono' logged on from 127.0.0.1
== Using SIP RTP CoS mark 5
> Channel SIP/jleon891-00000002 was answered.
-- Executing [*282@internos:1] VoiceMailMain("SIP/jleon891-00000002", "s334@default")
-- <SIP/jleon891-00000002> Playing 'vm-youhave.ulaw' (language 'es')
-- <SIP/jleon891-00000002> Playing 'digits/6.ulaw' (language 'es')
-- <SIP/jleon891-00000002> Playing 'vm-messages.ulaw' (language 'es')
-- <SIP/jleon891-00000002> Playing 'vm-0ld.ulaw' (language 'es')
-- <SIP/jleon891-00000002> Playing 'vm-onefor.ulaw' (language 'es')
-- <SIP/jleon891-00000002> Playing 'vm-messages.ulaw' (language 'es')
-- <SIP/jleon891-00000002> Playing 'vm-0ld.ulaw' (language 'es')
-- <SIP/jleon891-00000002> Playing 'vm-opts.ulaw' (language 'es')
-- <SIP/jleon891-00000002> Playing 'vm-helpexit.ulaw' (language 'es')
-- <SIP/jleon891-00000002> Playing 'vm-options.ulaw' (language 'es')
== Spawn extension (internos, *282, 1) exited non-zero on 'SIP/jleon891-00000002'
== Manager 'tuxfono' logged off from 127.0.0.1
server*CLI>
```

Figura 44. Llamada al buzón de voz en la central

CONCLUSIONES Y RECOMENDACIONES

Al terminar con la implementación del sistema TuxFono y al haber llevado a cabo una detallada fase de pruebas y demostraciones, se ha llegado a obtener las siguientes resoluciones.

Conclusiones

1. Tuxfono muestra una interfaz web sencilla, la cual facilita la interacción con el usuario mostrando directamente la información necesaria para comunicarse con sus contactos.
2. El sistema es independiente de la instalación de agentes o extensiones en el navegador web, para la realización automática de llamadas con un solo click. Demostrando así su funcionamiento en: Mozilla Firefox y Google Chrome. Además de ofrecer la posibilidad de trabajar con una gran variedad de Softphones, basados en el protocolo SIP, en lugar de vincular su funcionamiento a uno solo.
3. Las herramientas de software libre que han sido utilizadas para el desarrollo de este proyecto guardan una gran ventaja sobre las privativas. Esto gracias a que no tienen un costo por licencia y además la comunidad que las elabora permite el acceso al código

fuentes y documentación, para ser usados en la creación de proyectos más sofisticados y ajustables a las necesidades del cliente.

4. TuxFono cumple con el objetivo de convertirse en una herramienta que ofrece alta disponibilidad y eficiencia, debido a que brinda diversas oportunidades al llamante de comunicarse con su destino haciendo uso del correo de voz o retrollamada.
5. Al desarrollar una pequeña central telefónica con Asterisk, se logró ver el gran potencial que tiene esta herramienta; el cual permite la creación de un sistema estable, de bajo costo, escalable, altamente personalizable, y que además se integra con otros lenguajes de programación para realizar acciones de lógica más avanzada.

Recomendaciones

En lo referente a la realización de TuxFono es importante tomar ciertas consideraciones:

1. Establecer el códec adecuado para la realización de la llamada, tomando en cuenta la calidad de audio que ofrece y el ancho de banda del mismo. Para así cargarlo al iniciarse el servicio de Asterisk.

2. Revisar que las políticas de seguridad del servidor se encuentren correctamente establecidas, o en el peor de los casos deshabilitadas, para que el usuario pueda llegar a él sin problema.

3. Contar con las herramientas adecuadas, micrófono y audífonos, para poder evitar problemas de audio o ecos que pueden hacer creer que el problema es el sistema, cuando en realidad se trata de las terminales.

4. Establecer adecuadamente las condiciones de ejecución de la retrollamada, para evitar sobrecarga de datos en el servidor o un congestionamiento de la intranet.

TRABAJO A FUTURO

A medida que se avanzó en el desarrollo de TuxFono y la experiencia que sus desarrolladores adquirieron en el campo laboral. Se puede considerar la extensión de este proyecto a nivel empresarial, donde sería más rentable, y no personal. Esto quiere decir que habría una cuenta para la empresa e individualmente para el empleado, manejando a nivel de la compañía el cobro de las llamadas. Se tendría además que manejar contextos para llamadas fuera de la empresa a teléfonos convencionales y celulares. La empresa se beneficiaría al evitar la necesidad de compra y mantenimiento de los equipos necesarios para una central telefónica, los costos del minuto por llamada se esperan que sean insignificantes para que la ganancia se vea reflejada en la cantidad de comunicaciones que maneja el sistema.

APÉNDICE

GLOSARIO DE TERMINOS

Asterisk: Software libre y de código abierto que realiza las funciones de una central telefónica tradicional/IP.

Telefonía IP: Se refiere a las comunicaciones telefónicas realizadas a través de una red TCP/IP transportando la voz en paquetes de datos.

Voz sobre IP: Conjunto de recursos que hacen posible transformar las señales analógicas de audio, en señales digitales o paquetes; para transportarlas a través de una red de datos, usando un protocolo de Internet.

PBX: Central telefónica privada, conectada directamente a la red de telefonía pública, que gestiona las llamadas telefónicas internas, entrantes y salientes dentro de una organización.

Macro: Es un conjunto de instrucciones que se ejecutan de manera secuencial mediante una sola llamada o ejecución. Sirven para reutilizar código.

PHP: Lenguaje de programación usado en la creación de páginas web dinámicas. La interpretación y ejecución de este tipo de lenguaje lo realiza el servidor, mientras que el navegador o cliente muestra el resultado de esta acción en código HTML.

MySQL: Sistema de gestión de base de datos relacional, multihilo, multiusuario y de código abierto.

UDP: Protocolo a nivel de transporte, que esta basado en el intercambio de datagramas. Permite el envío de paquetes sin establecer conexión previa, tampoco tiene confirmación de recepción o control de flujo.

TCP: Protocolo a nivel de transporte que establece una conexión previo al envío de paquetes. Garantiza que los datos llegarán íntegramente a su destino y en el orden en que se transmitieron.

RTP: Protocolo a nivel de sesión, que se utiliza para la transmisión de información en tiempo real, como por ejemplo: audio y video. Se sitúa sobre UDP en el modelo OSI.

ANEXOS

ANEXO A: CÓDIGO

I. Fichero “res_config_mysql.conf”

```
[tuxfono]
```

Nombre del contexto que contendrá la información de conexión.

```
dbhost = localhost
```

Dirección IP del equipo que contiene la base de datos.

```
dbname = asterisk
```

Nombre asignado a la base de datos.

```
dbuser = root
```

Nombre del usuario que controla la base.

```
dbpass = codigo
```

Clave de acceso a la base de datos.

```
dbport = 3306
```

Puerto que usa para conectarse con la base, por defecto es el 3306.

```
dbsock = /var/lib/mysql/mysql.sock
```

Ubicación del socket que debe utilizar para interactuar con MySQL.

```
requirements=createclose
```

Instrucción que permite crear un campo adicional en la tabla en que caso de que el mismo no se encuentre y Asterisk lo requiera.

II. Fichero “extconfig.conf”

Las líneas de código que se describen a continuación tienen un formato de escritura predeterminado, de otra forma Asterisk no podrá reconocerlas correctamente; y obedecen a la siguiente regla:

nombre_familia => driver, nombre_contexto_base, nombre_tabla

Siguiendo el formato descrito, la estructura del archivo es como sigue:

```
[general]
```

Nombre del contexto.

```
extensions => mysql, tuxfono, extensions
```

Redirecciona el extensions.conf a la tabla llamada extensions.

```
voicemail => mysql, tuxfono, voicemail
```

Redirecciona el voicemail.conf a la tabla llamada voicemail.

```
sipusers => mysql, tuxfono, sip_buddies
```

Redirecciona los sip users en el sip.conf a la tabla llamada sip_buddies.

```
sippeers => mysql, tuxfono, sip_buddies
```

Redirecciona los sip peers en el sip.conf a la tabla llamada sip_buddies.

III. Fichero “sip.conf”

[general]

Nombre del contexto.

context=internos

Nombre del contexto por defecto donde buscará a los usuarios, cada vez que se realiza una llamada.

allowguest=no

Opción que rechaza llamadas de invitados

udpbindaddr=0.0.0.0

Dirección IP de la interfaz por la que asterisk escucha, 0.0.0.0 indica todas las interfaces.

tcpenable=no

Evita que el servidor permita conexiones TCP

qualify=yes

Verifica si el cliente es alcanzable, revisa su estado cada 60 segundos.

language=es

Especifica el lenguaje usado por las aplicaciones usadas en el plan de marcado, en caso de que no se desee usar las que vienen por defecto con el software.

rtccachefriends=yes

Permite al sistema supervisar el estado de los dispositivos conectados.

IV. Fichero “extensions.conf”

[general]

Contexto, donde se especificaran opciones generales para la central.

autofallthrough=yes

Si está activado y una extensión se queda sin cosas por hacer, termina la llamada de la mejor forma posible. Si no está activada se queda esperando otra extensión.

clearglobalvars=no

Si está activado se liberan las variables globales cuando se recargan las extensiones o se reinicia Asterisk.

[internos]

Contexto por defecto, donde la central buscará las extensiones cuando se realice una llamada.

switch => Realtime/@extensions

Indica que debe leer esta información desde la tabla *extensions*.

[macro-retrollamada]

Contexto donde se define los pasos necesarios para la retrollamada.

switch => Realtime/@extensions

Indica que debe leer esta información desde la tabla *extensions*.

[macro-extensions]

Macro donde se describe los pasos a llevar a cabo por la central, cada vez que un usuario intenta llamar a otro.

switch => Realtime/@extensions

Indica que debe leer esta información desde la tabla *extensions*.

[extensions-inverso]

Contexto que llama a los scripts AGI para que se realice el cobro por la retrollamada que se esta realizando.

switch => Realtime/@extensions

Indica que debe leer esta información desde la tabla *extensions*.

ANEXO B: FIGURAS

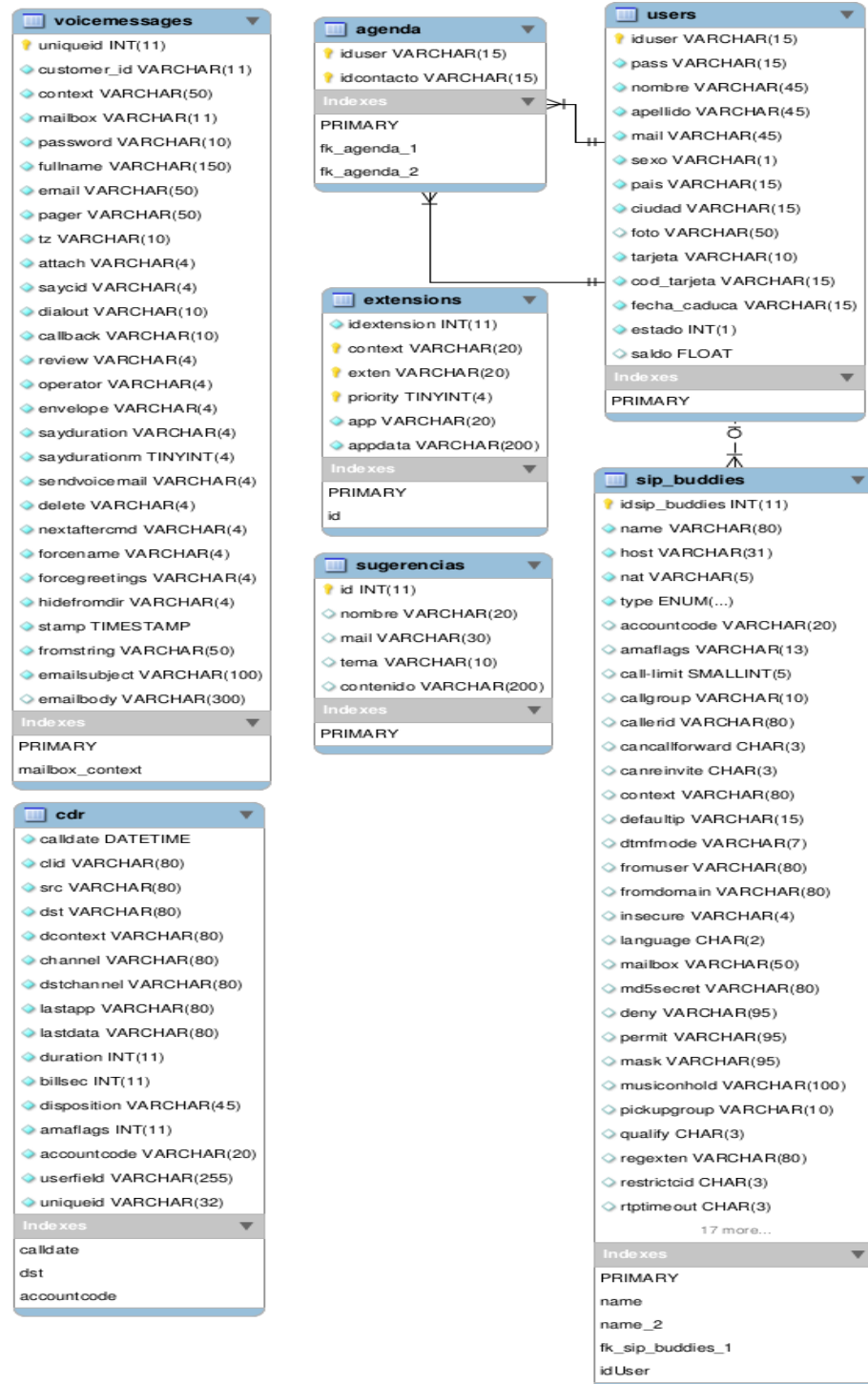


Figura 45. Modelo lógico de la Base de Datos: asterisk

```

<?php
session_start();
include_once("objetos/saldo.class.php");
$con = new Saldo;
$exten_destino = $_POST["exten_dest"];
$sipbuddy_origen = $_POST["sipbuddy"];
$idUsuario = substr($sipbuddy_origen,0,-1);
$timeout = '100';
$saldo = $con->obtenerSaldo("$idUsuario");
if($saldo <= 0){
    echo "Ud. NO dispone de saldo suficiente para realizar la llamada...";
}else{

    $tiempoDisponible = ( $saldo * 100 * 60 * 1000)/3 ;
    $socket=fsockopen("localhost","5038", $errno, $errstr, $timeout);
    fputs($socket, "Action: Login\r\n");
    fputs($socket, "UserName: tuxfono\r\n");
    fputs($socket, "Secret: codigol\r\n\r\n");
    fputs($socket, "Action: Originate\r\n");
    fputs($socket, "Channel: SIP/" . $sipbuddy_origen . "\r\n");
    fputs($socket, "Timeout: 50000\r\n");
    fputs($socket, "CallerID: $idUsuario\r\n");
    fputs($socket, "Context: internos\r\n");
    fputs($socket, "Exten: " . $exten_destino . "\r\n");
    fputs($socket, "Priority: 1\r\n");
    fputs($socket, "Variable: varSIPBuddyOrigen=" . $sipbuddy_origen . "\r\n");
    fputs($socket, " \r\n\r\n");
    $wrets=fgets($socket);
    $fin = false;
    date_default_timezone_set( 'America/Guayaquil' );
    $tiempoInicio = '0';
    $tiempoControl = '0';
    while (!$fin) {
        $linea = fgets($socket);
        //echo "-> " . $linea ;
        if(strpos(trim($linea), "Event: Bridge") == '0'){
            $tiempoInicio = time();
            $fechaInicio = date ( "h:i:s" );
        }
        if(strpos(trim($linea), "Event: Hangup") == '0'){
            $fin = true;
            $tiempoFin = time();
            $fechaFin = date ( "h:i:s" );
        }
    }
    $nuevoSaldo = $con->obtenerSaldo("$idUsuario");

    echo "HORA INICIO: " . $fechaInicio . "<br />" ;
    echo "HORA FIN: " . $fechaFin . "<br />" ;
    echo "SALDO INICIAL: " . $saldo . "<br />" ;
    echo "SALDO ACTUAL: " . $nuevoSaldo . "<br />" ;
    fclose($socket);
}
?>

```

Figura 46. Script realizarLlamada

ANEXO C: TABLAS

Campo	Tipo	Descripción
idsip_buddies <PK>	INT(11)	SipBuddy asignado al usuario
name	VARCHAR(80)	Nombre del propietario del sipbuddy.
host	VARCHAR(31)	Dirección IP del dispositivo remoto.
nat	VARCHAR(5)	Indica si el dispositivo está detrás de un NAT o no.
type	ENUM	Puede ser: peer, user o friend.
accountcode	VARCHAR(20)	Los usuarios pueden o no asociarse a uno. Se usa para facturación.
amaflags	VARCHAR(13)	Se usa para guardar en los CDR y temas de facturación.
call-limit	SMALLINT(5)	Número de llamadas simultáneas por este sipbuddy.
callgroup	VARCHAR(10)	Define un grupo de llamadas.
callerid	VARCHAR(80)	Define el identificador cuando no hay ninguna otra información disponible
cancallforward	CHAR(3)	Permita el envío de llamadas.
canreinvite	CHAR(3)	Permite o no que los puntos finales intercambien mensajes RTP directamente.
context	VARCHAR(80)	Indica el contexto asociado en el plan de marcado.
defaultip	VARCHAR(15)	IP por defecto del cliente host= cuando es especificado como "dynamic"
dtmfmode	VARCHAR(7)	El modo en el que se transmiten los tonos. Pueden ser "RFC2833" o "INFO".
fromuser	VARCHAR(80)	Pone el nombre de usuario en el from por encima de lo que diga el callerID
fromdomain	VARCHAR(80)	Pone el campo From: de los mensajes SIP
insecure	VARCHAR(4)	Define como manejar las conexiones con peers.
language	CHAR(2)	Define el lenguaje.
mailbox	VARCHAR(50)	Extensión del contestador
md5secret	VARCHAR(80)	Contraseña cifrada con MD5.
deny	VARCHAR(95)	No permite una IP.
permit	VARCHAR(95)	Permite una IP.

Campo	Tipo	Descripción
mask	VARCHAR(95)	Máscara correspondiente a la IP establecida con permit o deny.
musiconhold	VARCHAR(100)	Una de las clases especificadas en musiconhold.conf.
pickupgroup	VARCHAR(10)	Define el grupo de llamadas validas para una aplicación pickup()
qualify	CHAR(3)	Para determinar cuando el dispositivo puede ser alcanzado
regexten	VARCHAR(80)	Campo utilizado para cálculos internos del RealTime.
restrictcid	CHAR(3)	Se usa para esconder el ID del llamante. Anticuada y en desuso
rtptimeout	CHAR(3)	Termina la llamada cuando llega a ese timeout si no ha habido tráfico rtp
rtpholdtimeout	CHAR(3)	Termina la llamada cuando llega a ese timeout si no ha habido tráfico rtp "on hold"
secret	VARCHAR(80)	Contraseña utilizada para autenticación.
setvar	VARCHAR(100)	Variable de canal establecida en todas las llamadas.
disallow	VARCHAR(100)	Permite deshabilitar un códec.
allow	VARCHAR(100)	Permite habilitar un códec.
fullcontact	VARCHAR(80)	Es el contacto SIP URI para los usuarios RealTime.
ipaddr	VARCHAR(45)	Dirección IP del peer. Usado para los usuarios de RealTime.
port	VARCHAR(5)	Puerto UDP en el que responderá Asterisk
regserver	VARCHAR(100)	Usado por el driver que relaciona MySQL-Realtime, para procesos internos.
regseconds	INT(11)	Usado por el driver que relaciona MySQL-Realtime, para procesos internos.
lastms	INT(11)	"0" Desconocido, "-1" Inalcanzable. Sin este campo no podrá realizar actualizaciones en la BD
username	VARCHAR(80)	Establece el nombre de usuario que Asterisk intenta conectar cuando una llamada es recibida.
defaultuser	VARCHAR(80)	Nuevo nombre para la variable 'username'.
subscribcontext	VARCHAR(80)	Determina el contexto donde se busca en caso de petición

Campo	Tipo	Descripción
useragent	VARCHAR(20)	Permite definir la cabecera del SIP "User-Agent"
extensión	VARCHAR(20)	Extensión asignada al sipbuddy
iduser <FK>	VARCHAR(15)	Usuario al que pertenece el sipbuddy

Tabla C-I. Descripción Completa de la Tabla: sip_buddies

BIBLIOGRAFÍA

- [1] Van Meggelen J., Smith J. y Madsen L., Asterisk The Future of Telephony Second Edition, Editorial O'Reilly Media, 2007, fecha de consulta: 23 Junio 2011, Pág. 45 – 52, 73 - 80.
- [2] Gonçalves E. Flavio, Configuration Guide for Asterisk PBX Second Edition, 2007, fecha de consulta: Junio 2011, Pág. 65 – 66.
- [3] Bahjons, Asterisk-ARA, <http://bahjons.com/stuff/asterisk-realtime-installation-guide>, Noviembre 2007, fecha de consulta: Junio 2011.
- [4] VozToVoice, Retrollamada, <http://voztovoice.org/?q=node/63>, Octubre 2008, fecha de consulta: Agosto 2011.
- [5] Asterisk.Name, Módulos de Asterisk modules.conf, <http://www.asterisk.name/modules.conf.html>, 2008, fecha de consulta: Julio 2011.
- [6] PHPAGI, Asterisk-AGI-PHP, <http://phpagi.sourceforge.net/>, Septiembre 2010, fecha de consulta: Agosto 2011.
- [7] VoIP-Info, Asterisk-Variables, <http://www.voip-info.org/wiki/view/Asterisk+variables>, 2011, fecha de consulta: Agosto 2011.
- [8] VozToVoice, Asterisk-Voicemail.conf, <http://voztovoice.org/?q=node/53>, 2008, fecha de consulta: Junio 2011.
- [9] Systemadmin, SELinux, <http://systemadmin.es/2008/12/desactivar-selinux>, Diciembre 2008, fecha de consulta: Junio 2011.

[10] Asterisk-rd, Funcionalidades de una Central Asterisk,

<http://asterisk-rd.blogspot.com/2011/07/funcionalidades-centralita->

[asterisk.html](http://asterisk-rd.blogspot.com/2011/07/funcionalidades-centralita-asterisk.html), Julio 2011, fecha de consulta: Agosto 2011.

[11] Protocolo de inicio de sesión (SIP), Apuntes de la materia de graduación VoIP usando software libre, fecha de consulta: Junio 2011.

[12] VoIP-Info, Asterisk Manager Interface (AMI),

<http://www.voip-info.org/wiki/view/Asterisk+manager+API>, 2012, fecha de consulta: Agosto 2011.