

# **Diseño e implementación de una agenda telefónica en línea que permita al usuario llamar a sus contactos desde una página web.**

Mera Mejía Juan León <sup>(1)</sup>, Minga Bastidas Soraya Mariela <sup>(2)</sup>, Astudillo Brocel Gabriel Antonio <sup>(3)</sup>  
Facultad de Ingeniería en Electricidad y Computación <sup>(1) (2) (3)</sup>  
Escuela Superior Politécnica del Litoral (ESPOL)  
Campus “Gustavo Galindo V.”, Km. 30.5 vía Perimetral  
Apartado 09-01-5863. Guayaquil-Ecuador  
[emera@fiec.espol.edu.com](mailto:emera@fiec.espol.edu.com) <sup>(1)</sup>, [sminga@fiec.espol.edu.ec](mailto:sminga@fiec.espol.edu.ec) <sup>(2)</sup>, [gastudil@fiec.espol.edu.ec](mailto:gastudil@fiec.espol.edu.ec) <sup>(3)</sup>

## **Resumen**

*El presente proyecto consiste en la elaboración de una Agenda Telefónica Virtual, misma que podrá ser accedida por el cliente desde cualquier navegador web. Permitiendo realizar llamadas entre los usuarios registrados, haciendo uso de cualquier teléfono IP, con un solo click y evitar la necesidad de instalar complementos adicionales para su uso exclusivo.*

*Además incluye un sistema de cobranzas que es manejado con la compra de tarjetas virtuales dentro de la página web, las cuales deberán descontarse de su tarjeta de crédito, y la opción de activar una retrollamada en caso de que la línea se encuentre ocupada o el usuario no este disponible.*

*El sitio web fue desarrollado con lenguaje PHP y montado sobre un servidor Apache para su visualización, mientras que el manejo de las llamadas lo realiza una central telefónica basada en Asterisk. La fusión de ambas partes se llevo a cabo mediante el uso de una base de datos MySQL, permitiendo el uso de la Arquitectura en Tiempo Real de Asterisk (ARA) para crear, modificar o eliminar datos en la central en tiempo de ejecución.*

**Palabras Claves:** ESPOL, VoIP, Asterisk, MySQL, PHP, AGI, AMI, PHP-AGI PHP-AMI, ARA.

## **Abstract**

*This project involves the development of a Virtual Phonebook; same may be accessed by the client from any web browser. It allows calls between registered users, using any IP phone, with a single click and avoids the need to install additional plugins for its exclusive use.*

*It also includes a collection system that is managed by buying virtual cards within the website, which must be deducted from client's credit card, and the option to trigger a callback in case of the line is busy or the user is not available.*

*The website was developed with PHP and Apache server mounted for display, while the call handling is done by Asterisk-base PBX. The merger of both parties was carried out using a MySQL database, allowing the use of Asterisk Real-Time Architecture (ARA) to create, modify or delete data in the PBX at runtime.*

**Keywords:** ESPOL, VoIP, Asterisk, MySQL, PHP, AGI, AMI, PHP-AGI PHP-AMI, ARA.

## Introducción

En la actualidad se requiere mantener una constante comunicación con la red de contactos, para así poder desarrollar mejores relaciones tanto a nivel laboral como personal. El principal problema que se presenta es la disponibilidad de la información, que puede estar distribuida en varios dispositivos, lo cual puede generar confusión, o encontrarse centralizarla, lo que crea dependencia de un solo equipo.

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., que permita la realización de llamadas telefónicas pre-pagadas a través de VoIP mediante el acceso a un sitio web, el mismo que contiene la información de la red de contactos con los cuales se podrá comunicar de manera rápida y directa con un solo click.

## 1. Planteamiento

### 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, se puede 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 para trabajar con el sistema.

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

## 1.3. Objetivos

### 1.3.1. General

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. Específicos

- Integrar diversas herramientas de software libre aprendidas a lo largo de la carrera.
- Proveer de gran movilidad y alta disponibilidad a los usuarios, ya que podrán hablar con su red de contactos, solamente al acceder a Internet y tener un teléfono IP disponible.
- Simplificar la conexión entre llamantes, de tal forma que se elimina la necesidad de marcado, de recordar el número del destinatario y la dependencia de un dispositivo para almacenar esta información.

## 1.4. Metodología

Para la centralita telefónica 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, cada vez que se realicen cambios.

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.

## 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.

### 2.2. Protocolo de inicio de sesión (SIP)

Su 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. Esto se traduce en que cada comunicación hace uso de dos canales. [11]

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

### 2.3. Interfaz de administración de Asterisk (AMI)

Como su nombre lo indica 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]

### 2.4. Interfaz de puerta de enlace de Asterisk (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 recursos externos; permitiendo realizar acciones que de otro modo serían demasiado complicadas o imposibles. [1]

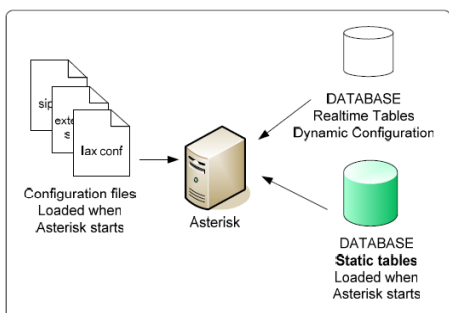
Inicialmente Asterisk pasa al script un conjunto de variables con sus respectivos valores, este traslado de información culmina cuando la central envía una línea en blanco. Posteriormente AGI toma el control del plan de marcado enviando comandos a Asterisk por medio del STDOUT, cada uno de estas sentencias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

### 2.5. Arquitectura en tiempo real de Asterisk (ARA)

La configuración de Asterisk se basa en la modificación de ciertos archivos, esto presenta algunas desventajas al encontrarse en un entorno de producción, porque no se puede detener el servicio para modificar o actualizar algún fichero, ya que representaría pérdidas monetarias considerables e inconformidad por parte de los usuarios.

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.

Una de las formas de aplicación del ARA es de manera “estática”, en la cual resulta lo mismo cargar la información desde un archivo o de la base de datos, para luego almacenarlo en memoria. Por otro lado de manera “dinámica”, los objetos no son guardados en memoria sino que son creados durante la ejecución de una llamada y luego eliminados. En la figura 1 se puede observar las diferentes formas que tiene la central para cargar nueva información.



**Figura 1.** Funcionamiento de la Arquitectura en Tiempo Real de Asterisk

Para TuxFono se usa el método dinámico, es decir que todo va a ser almacenado en una base de datos y ser cargado sobre la marcha, permitiendo así añadir nuevos usuarios y modificar los planes de marcado en tiempo real.

### 3. Implementación

#### 3.1. Hardware

El equipo sobre el cual se trabajó no es una estación física sino virtual, puesto que durante el desarrollo del proyecto 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 1 y 2 se describen tanto las características del equipo anfitrión, como de la máquina virtual respectivamente.

**Tabla 1.** Características del Equipo Anfitrión

<b>Memoria RAM:</b>	4 GB
<b>Procesador:</b>	Intel Core 2 Duo 2,40 GHz
<b>Capacidad Disco:</b>	160 GB
<b>S.O. – Kernel:</b>	Fedora 16 – kernel v3.3.0-4
<b>Arquitectura:</b>	X86_64

**Tabla 2.** Características de la Máquina Virtual

<b>Memoria RAM:</b>	1 GB
<b>Procesador:</b>	Intel Core 2 Duo 2,40 GHz
<b>Capacidad Disco:</b>	15 GB
<b>S.O. – Kernel:</b>	Fedora 16 – kernel v3.2.7-1
<b>Arquitectura:</b>	I386

#### 3.2. 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.

##### 3.2.1. Central Telefónica

El software que se utiliza para hacer las veces de central telefónica o PBX se describen en la Tabla 3.

**Tabla 3.** Software utilizado en la PBX

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

En la PBX, el uso de MySQL es indispensable para el manejo del ARA, ya que permitirá crear usuarios SIP y modificar el plan de marcado sin la necesidad de recargar cada uno de los módulos de manera individual.

##### 3.2.2. Sitio Web

En el desarrollo web de la agenda virtual, se ha estudiado, adecuado y desarrollado con las herramientas descritas en la Tabla 4 según las necesidades de TuxFono; tanto en el ámbito de funcionamiento, como de presentación.

**Tabla 4.** Software utilizado en el Sitio Web

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

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

##### 3.2.3. 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 proveedor de Voz sobre IP. Para la realización de pruebas, se ha usado los mencionados en la Tabla 5.

**Tabla 5.** Software utilizado como Softphone

<b>Softphone Windows:</b>	X – Lite v4.0
<b>Softphone Linux:</b>	Twinkle 1.4.2

### 3.3. Configuración

Posterior a la instalación de los paquetes correspondientes, 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.

#### 3.3.1. Seguridad

El ámbito de seguridad es sumamente extenso, pero en el desarrollo del sistema se han tomado las precauciones mínimas ya que no se encuentra en un ambiente de producción.

En el servidor se deshabilitará el SELinux, cuya función es la de proveer de diversas políticas de seguridad, para así facilitar la ejecución de las tareas de TuxFono, sin la necesidad de configurarlo.

Además se detiene el servicio de IPTables, el cual realiza las funciones de un firewall, permitiendo así que los clientes se registren en el sistema sin ningún inconveniente. Este script contiene diferentes reglas para permitir ingresos o salidas de los paquetes del equipo.

#### 3.3.2. 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.

Para que TuxFono pueda interactuar con la base de datos, tanto a nivel de la central, como del sitio web. Se debe iniciar el servicio de MySQL.

Se debe iniciar y configurar adecuadamente el servidor web, Apache, donde se encuentra alojada la página web. De esta forma los usuarios podrán utilizar los servicios de TuxFono desde cualquier navegador web.

#### 3.3.3. Archivos de Asterisk

Lo principal es la integración del ARA en el proyecto, para lo cual se necesitan establecer ciertos parámetros en los archivos de configuración de Asterisk.

En el fichero *res\_config\_mysql.conf*, se encuentra la información para establecer conexión con la base de datos creada para TuxFono, como son: la dirección del servidor, el nombre de la base, usuario, clave, puerto y el socket usado para interactuar con la misma.

Mientras que en el archivo *extconfig.conf* se detalla la forma en que se relacionan las tablas de la base con los archivos de configuración de Asterisk, los cuales tienen nombres 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.

En el archivo *sip.conf*, contiene información correspondiente a los usuarios que se conectan a la central usando el protocolo SIP, por cada usuario debería existir un contexto. Pero 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.

Finalmente en el archivo del plan de marcado, *extensions.conf*, se va a encontrar un homónimo de los contextos definidos en la tabla, *extensions*, con su respectiva redirección a la misma. Es importante que ambas partes se mantengan actualizadas porque de lo contrario pueden ocurrir errores en tiempo de ejecución.

#### 3.3.4. Base de Datos

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

En lo que respecta a la base, se ha dividido las tablas que la conforman en dos grupos.

Tablas Relacionadas:

- **users.-** Almacena los datos personales, saldo, usuario y clave de acceso correspondientes del cliente.
- **agenda.-** Guarda la red de contactos de cada uno de los usuarios.
- **sip\_buddies.-** Cada registro representa a un contexto que debería existir en el archivo *sip.conf*

Tablas NO Relacionadas:

- **extensions.-** Contiene todo el plan de marcado, incluidos los diferentes contextos y macros que se han definido previamente en el *extensions.conf*.

- **voicemessages.-** Cada registro equivale a un buzón de voz de un usuario SIP registrado en el sistema.
- **cdr.-** Contiene un registro único por cada una de las llamadas realizadas usando la PBX de TuxFono.
- **sugerencias.-** Guarda los comentarios que los clientes dejan en el sitio web.

### 3.3.5. Scripts AGI - AMI

Para poder realizar acciones que requieren una lógica avanzada se utilizan programas externos, permitiendo controlar el plan de marcado o la central en sí. En esta ocasión se hará uso del lenguaje PHP.

El script denominado: *consultarTiempo.php*, se encarga de obtener el tiempo disponible que tiene el usuario para hablar, haciendo una conversión de dólares a milisegundos y retornando el resultado al plan de marcado en la variable *TIEMPO\_DISPONIBLE*. La metodología usada se explica en la figura 2.

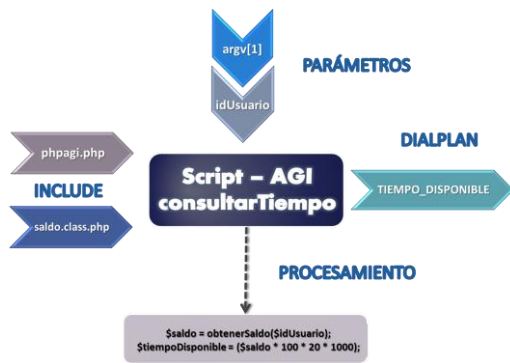


Figura 2. Descripción del script ConsultarTiempo.php

El fichero de *setearSaldo.php* permite actualizar el saldo del cliente en la base de datos una vez que ha terminado la llamada o retrolamada. En la figura 3 se puede observar el procedimiento efectuado.

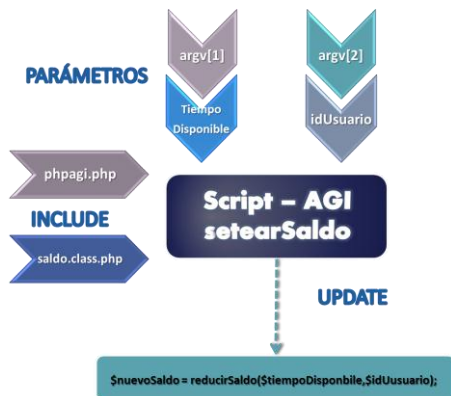


Figura 3. Descripción del script setearSaldo.php

Finalmente el script de *realizarLlamada.php*, el cual es el más importante del sistema, permite realizar la llamada desde la página web dando click sobre la ubicación del destinatario. Su descripción se puede observar en la figura 4.



Figura 4. Descripción del script realizarLlamada.php

## 4. Manual del Usuario

### 4.1. Registro en el sistema

Para hacer uso de los servicios que brinda el sistema es necesario estar registrado. En la parte superior derecha de la página se encuentra la opción para “Iniciar Sesión”, haciendo uso del usuario y contraseña respectivo, caso contrario la alternativa “Registrarse” es la que se debe seleccionar, como lo indica la figura 5.



Figura 5. Opción para ingresar al sistema o registrarse.

Para crear una cuenta se debe completar el formulario de registro con la respectiva información personal, crediticia y claves de acceso.

Posteriormente al dar clic en “Enviar” se crea la cuenta, los tres usuarios SIP respectivos para el cliente y automáticamente el navegador redirecciona al usuario a su Agenda personal, donde se podrá agregar los usuarios de la red contactos del cliente.



## 4.2. Añadir contactos

Desde la Agenda se tiene varias opciones, como son: llamar al buzón de voz, elegir ubicación, añadir contacto, eliminar contacto. Como se muestra en la figura 6.



Figura 6. Opciones de la Agenda de TuxFono.

Al dar click sobre el botón “Añadir Contacto” se muestra la página de Buscar, en la cual hay un cuadro de texto que al ingresar el nombre del contacto que se desea añadir, se mostrarán las coincidencias encontradas según los clientes registrados en el sistema.

Lo que muestra a continuación en la figura 7 es la información del usuario seleccionado de las coincidencias mostradas y un botón para añadirlo. Finalmente redirige al propietario a su Agenda donde podrá ver el contacto que ha agregado recientemente.



Figura 7. Resultado de la búsqueda de usuarios.

## 4.3 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 8.

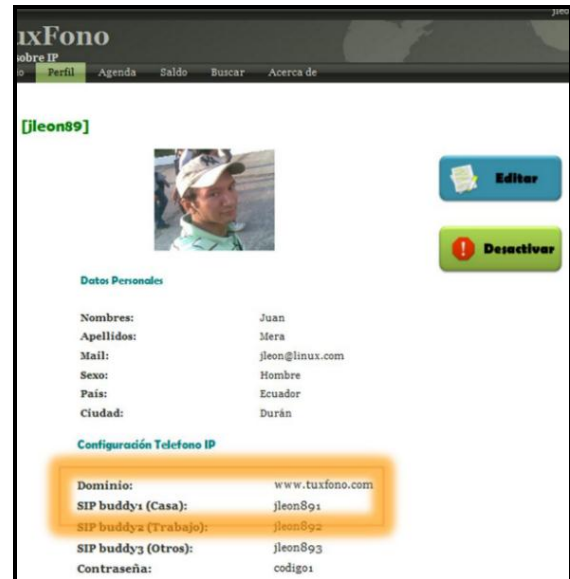


Figura 8. Perfil del Usuario – Info. Teléfono.

Cuando se trabaja con sistemas operativos Windows existen diversas opciones en lo que respecta a Softphones, como son: X-Lite, Zoiper por dar un ejemplo. Mientras que en los sistemas basados en Linux se encuentra Twinkle y Ekiga.

Para la realización de pruebas se decidió usar X-Lite y Twinkle. Ya que ambos soportan el protocolo SIP y presentan una interfaz bastante intuitiva, bajo nuestro criterio.

Se ingresan los datos mostrados en el perfil del usuario correctamente y se registra el teléfono. Es importante tener en consideración que la extensión que configuremos debe ser la misma que elijamos antes de realizar una llamada.

## 4.3 Compra de saldo

Previo a la realización de la llamada es necesario adquirir las tarjetas virtuales que equivalen a tener saldo disponible para hablar.

En la página de Saldo 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.

## 4.4 Realización de llamada

Lo primero que debe hacer el usuario es dirigirse a su agenda y establecer correctamente su ubicación. En esta ocasión se ha decidido tomar el primer usuario SIP, que corresponde a la extensión “casa”; mismo que se usó para la configuración del Softphone previamente.

Posteriormente se selecciona una de las tres extensiones que tiene el destinatario para recibir una llamada, como se indica en la figura 9.

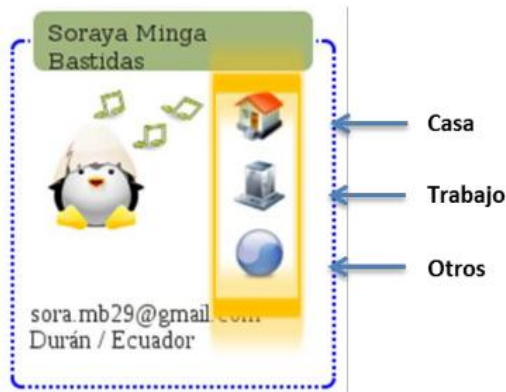


Figura 9. Imagen de un contacto en la Agenda.

El sistema se encuentra programado para que inicialmente realice una llamada de confirmación al teléfono del llamante. Una vez que es contestada, la central hace timbrar 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.

## 5. 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 jlmera89, como el llamante, y a sorymb, como el destinatario.

### 5.1. Llamada entre Usuarios

Inicialmente se establece la ubicación correcta en la página web, en la Agenda se ubica al destinatario y se da click sobre una de sus tres extensiones, que están representadas con iconos.

Asterisk se encarga de realizar una llamada de confirmación a quien origina la llamada, jleon89, como se observa en la figura 10.

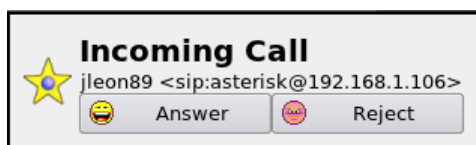


Figura 10. Llamada de confirmación de Asterisk.

Una vez contestada, el sistema inicia una llamada hacia al destinatario, sorymb, indicando el nombre del llamante original. Al finalizar se muestra en la pantalla del navegador, de quien llama, la información correspondiente a la duración y costo de la misma.

Lo que ocurre de manera interna es la ejecución de la macro “extensions”, donde se ubica la extensión del destino para iniciar la llamada con la aplicación, Dial, la cual usa un parámetro especial “L”, que establece la duración que tendrá la misma en base al saldo del cliente.

### 5.2. Activación Retrollamada

Una retrollamada consiste en la ejecución de manera automática de un determinado número de intentos de timbrado a un usuario, cuando este contesta se inicia como subproceso una llamada a quién activo la retrollamada en primer lugar. Esto se lo hace para incrementar la posibilidad de establecer comunicación entre ambos, sin la necesidad de que el usuario lo haga por sí mismo.

Esta función se activa con la opción nueve del menú en caso de que la línea se encuentre ocupada. Inicialmente se ejecuta la macro, retrollamada, la cual crea un archivo con los parámetros necesarios para su ejecución, para posteriormente moverlo al directorio: `var/spool/asterisk/outgoing` donde Asterisk lo ejecutará.

La macro antes menciona origina el timbrado al destinatario de la retrollamada, cuando este contesta entra en ejecución la macro “extensions-inverso”, que ejecuta la llamada hacia quien activo la retrollamada, realizando los cobros respectivos cuando esta termina con ayuda de los scripts AGI.

## CONCLUSIONES

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.

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.

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

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. 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.

## RECOMENDACIONES

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.

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.

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.

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

## 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>, 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.