



## Implementación de un sistema para la generación automática de llamadas salientes en un Call Center.

Pontón, M.; Yáñez, W.; Astudillo, G. Ing.

Facultad de Ingeniería en Electricidad y Computación

Escuela Superior Politécnica del litoral (ESPOL)

Campus Gustavo Galindo, Km. 30.5 vía Perimetral

Apartado 09-01-5863. Guayaquil, Ecuador

[melmpont@espol.edu.ec](mailto:melmpont@espol.edu.ec), [wenpayan@espol.edu.ec](mailto:wenpayan@espol.edu.ec), [gastudil@espol.edu.ec](mailto:gastudil@espol.edu.ec)

### Resumen

*El proyecto que se implementó consiste en un sistema telefónico basado en Asterisk, para lo cual se desarrolló una aplicación PHP que hace uso de la Interfaz de Administración de Asterisk (AMI), capaz de generar automáticamente llamadas salientes programadas en base al monto adeudado por el cliente. Para esto se estableció como escenario de prueba una empresa de gestión de cobranzas, que mantiene una base de datos de los clientes que poseen una tarjeta de crédito, dicha base es actualizada periódicamente por el administrador del sistema.*

*Al momento de ejecución de las llamadas se valida que la grabación es escuchada, si no es así se programa la realización de una segunda llamada. Adicionalmente, periódicamente se genera un reporte de las llamadas realizadas a los clientes, el cual es entregado a la empresa.*

**Palabras claves:** Asterisk, AMI, llamadas automáticas, VoIp.

### Abstract

*The project that was implemented consist on an Asterisk based phone system, for which was developed a PHP application that uses the Asterisk Management Interface (AMI), capable of generating outgoing calls automatically scheduled based on the amount owed by the customer. For this test scenario set as a collection management company, which maintains a database of customers who own a credit card, the database is regularly updated by the system administrator.*

*Upon execution of the call is validated that the recording is heard, if not the program makes a second call. Additionally, each month a report is generated for calls made to customers, which is delivered to the company.*

**Key Words:** Asterisk, AMI, automatic calls, VoIp.

## 1. Introducción

Las llamadas de Voz sobre IP a través de Internet, o telefonía por Internet, se han convertido en una manera muy popular de ahorro en la infraestructura de telecomunicaciones de diversas empresas y organizaciones, ya que resultan muy baratas y, en muchas ocasiones, incluso gratis al hacer uso de las redes de transporte de datos para la transmisión de voz, lo que ha resultado en que la telefonía tradicional pierda terreno entre aquellos clientes que se adaptan fácilmente a las nuevas tecnologías, pues todo lo que se requiere es una única conexión a un red IP.

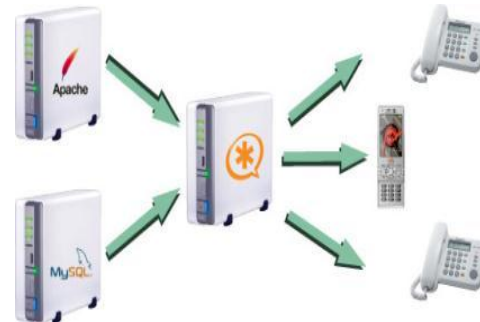
Un ejemplo de esto lo podemos apreciar en las principales empresas del sector público y privado tales como: Ministerio de Salud, Ministerio de Cultura, Instituto Nacional de Contratación Pública, CNT, AeroGal, RTS, Banco ProCredit, UNIVISA, entre otros; en las cuáles se encuentran implementado Elastix[1], que es un software libre de VoIP basado en Asterisk, desarrollado por una empresa ecuatoriana.

Mediante una adecuada configuración de una central telefónica Asterisk y haciendo uso de los módulos DAHDI podremos comunicarnos con clientes que se encuentren conectados a la PSTN. Por otro lado, la generación automática de llamadas permite tener un control más eficiente de los clientes que fueron contactados, evitando la contratación de una persona para la implementación de éste tipo de soluciones.

## 2. Diseño y Metodología

Para cumplir con los objetivos del proyecto, se instaló Asterisk sobre un servidor con sistema operativo Centos 5.4. Además se agregó el servicio de base de datos MySQL, para poder almacenar los datos de los clientes de la empresa, así como también el servicio web Apache, que permite ejecutar la aplicación web. Se procedió a instalar el compilador de PHP 5, lenguaje de programación en el cuál se desarrollaron los scripts con los que se generan las llamadas del sistema. Se instaló la tarjeta Digium TDM410P con 4 puertos FXO en el servidor, que posteriormente se configuró con su respectiva señalización.

El esquema a utilizar en este proyecto se define en la Figura 1.



**Figura 1:** Esquema del proyecto

Este sistema puede realizar llamadas automáticas haciendo uso de Interfaz de Administración de Asterisk, manejada a través de una aplicación web. Las llamadas son originadas por los canales Dahdi con los que cuenta la tarjeta de telefonía.

El usuario puede escuchar un mensaje con sus datos y un menú de opciones para realizar diferentes acciones.

## 3. Asterisk, FXO/FXS, DAHDI y AMI

### 3.1 Asterisk

La central telefónica de código abierto, es un software bajo licencia GPL, capaz de convertir a una simple computadora en una central de telefonía IP[2]. Este software proporciona servicios VoIP y puede interoperar con equipos de telefonía estándar básicos usando un hardware relativamente de bajo coste. Incluye muchas características tales como: buzón de voz, conferencias, IVR (Interactive Voice Response por sus siglas en inglés), distribución automática de llamadas, y muchas otras funcionalidades que pueden ser creadas por los usuarios, diseñando un plan de marcado que puede interactuar con scripts desarrollados en lenguajes como: Pascal, Ruby, PHP, Java, Perl, entre otros.

Esta centralita permite la conectividad entre las redes PSTN y las redes IP en tiempo real, para lo cual utiliza las tarjetas electrónicas FXS/FXO fabricadas por Digium u otros proveedores. SIP, H.323, IAX y MGCP son algunos de los muchos



protocolos VoIP que soporta Asterisk. Los protocolos básicamente son utilizados para registrar, autenticar y negociar direcciones IP y puertos, además de controlar el estado de las llamadas.

### 3.2 FXS/FXO

Se conoce con el nombre de FXO y FXS a los puertos usados por las líneas telefónicas analógicas (también denominados Servicio Telefónico Básico y Antiguo – POTS)[3].

#### - Interfaz de Central Externa FXO

Es el puerto que se comunica con la central telefónica o PBX. Sólo reciben tono de marcado desde las líneas (FXS). Son más conocidos como las entradas de las troncales.

#### - Interfaz de Abonado Externo FXS

Es el puerto que permite simular el comportamiento de una línea telefónica, pues envía tono de marcado, corriente para la batería y tensión de llamada. Se las conoce como salidas de extensión.

### 3.3 DAHDI, Interfaz de Dispositivos Digium de Asterisk

DAHDI brinda soporte para hardware, además de proporcionar controladores para tarjetas. Hasta el 19 de Mayo del 2008 era conocido con el nombre de Zaptel. Este paquete de DAHDI está compuesto por dos subpaquetes que son: kernel y tools. Para configurar DAHDI se deben realizar cambios en los archivos:

```
/etc/dahdi/system.conf
```

```
/etc/asterisk/chan_dahdi.conf[4]
```

### 3.4 AMI, Interfaz de Administración de Asterisk

La interfaz de administración de Asterisk permite que programas clientes se conecten a Asterisk mediante el puerto TCP/IP: 5038, que es el que se utiliza por defecto. Una vez conectados y autenticados pueden comenzar a ejecutar comandos y leer eventos, lo que proporciona la posibilidad de alterar el comportamiento de Asterisk desde un programa hecho a medida.

## 4. Componentes para la implementación

La implementación en general se basa en componentes de software libre como se muestran a continuación.

A nivel de software:

- Servidor PBX: Asterisk que se utilizó como centralita telefónica.
- Servidor de Base de Datos: MySQL, para administrar la base de datos proporcionada por la organización.
- Servidor Web: Apache, para generar la página de administración del sistema de cobranza.
- Festival: Paquete Texto-a-Voz para crear el archivo de audio, que escuchará el cliente durante la llamada.
- Sox: Para cambiar el formato del archivo generado por Festival, se utilizó el paquete Sound eXchange.

A nivel de hardware:

- Servidor instalado con el sistema operativo Linux distribución Centos 5.4.
- Tarjeta analógica TDM410p, FXO de 4 puertos.

## 5. Configuración en Asterisk

La configuración de Asterisk requerida para el proyecto se realizó básicamente en cuatro archivos: extensions.conf, manager.conf, system.conf y chan\_dahdi.conf.

### 5.1. Archivo extensions.conf

Este archivo se encuentra en la ruta /etc/asterisk/extensions.conf, y contiene el plan de marcado de la central telefónica. Es el archivo más importante de Asterisk y tiene como objetivo principal definir el plan de marcado, que determina el comportamiento que seguirá la central telefónica.

A continuación se detalla la configuración del mismo:

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



```
[salientes]
exten =>
200,1,Playback(outboundCall/archivoAudio${AUDIO})
exten =>
200,n,SayUnixTime(${FECHA},,ABdY)
exten =>
200,n,Background(outboundCall/opcion)
exten => 200,n,WaitExten(7)
```

```
;El cliente presiona un dígito inválido
exten => i,1,Playback(outboundCall/invalido)
exten => i,n,Goto(200,3)
```

```
;El cliente desea escuchar nuevamente el mensaje
exten => 3,1,Goto(200,1)
```

```
;El cliente escuchó el mensaje completo
exten => 5,1,MYSQL(Connect connid localhost
root password base_clientes)
exten => 5,n,MYSQL(Query resultid ${connid}
UPDATE cliente SET estado_llamada=1 where
no_cuenta=${IDCLIENTE})
exten => 5,n,MYSQL(Disconnect ${connid})
exten =>
5,n,Playback(outboundCall/despedida)
exten => 5,n,Hangup(98)
```

```
exten => t,1,Playback(vm-goodbye)
exten => t,n,Hangup()
```

## 5.2. Archivo manager.conf

Este archivo se encuentra en la ruta /etc/asterisk/manager.conf, y contiene las configuraciones del AMI (Asterisk Manager Interface). En la sección [general] se habilitó al administrador para que “escuche” los requerimientos del programa cliente, el puerto por defecto es el 5038.

```
[general]
enabled=yes
port=5038
bindaddr=0.0.0.0
webenabled=no
```

Luego se crearon usuarios, se les estableció una contraseña y se les habilitaron permisos de lectura y escritura.

```
[outboundCallX]
```

```
secret=secretpass
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.255
read=system,call,log,verbose,command,agent,u
ser,originate
write=system,call,log,verbose,command,agent,u
ser,originate
```

## 5.3. Archivo system.conf

Este archivo se encuentra en la ruta /etc/dahdi/system.conf, y en éste se configuró la señalización que utilizan los puertos presentes en la tarjeta analógica. Es autogenerado al ejecutar el comando dahdi\_genconf en la consola.

A continuación se detalla la configuración del mismo:

```
# Autogenerated by /usr/sbin/dahdi_genconf on
Fri Jun 3 09:56:02 2011
fxsks=1
echocanceller=mg2,1
fxsks=2
echocanceller=mg2,2
fxsks=3
echocanceller=mg2,3
fxsks=4
echocanceller=mg2,4

# Global data
loadzone = us
defaultzone = us
```

## 5.4. Archivo chan\_dahdi.conf

Este archivo se encuentra en la ruta /etc/asterisk/chan\_dahdi.conf, y en éste se configuraron los puertos dahdi con que cuenta la tarjeta analógica.

A continuación se detalla la configuración del mismo:

```
[channels]
; canales físicos: opciones por defecto para
todos los canales
usecallerid=yes
hidecallerid=no
callwaiting=no
threewaycalling=yes
transfer=yes
```

```
echocancel=yes
echotraining=yes
callprogress=yes
busydetect=yes
busycount=4
language=es
```

**; definicion de canales:**

```
group=0
context=haciaPSTN
signalling=fxs_ks ; señalizacion FXS para canal FXO
channel => 1-4 ; PSTN se conecta al puerto 4.
```

## 6. Scripts y Página PHP

### Scripts de generación de llamadas

Son los que se conectan con la Interfaz de Administración de Asterisk y obtienen los datos de la base para crear los archivos de audio que serán reproducidos a los clientes, se ubican en la ruta `/etc/asterisk/outboundCall/`.

#### llamada.php

El siguiente script, ubicado en la ruta `/etc/asterisk/outboundCall/`, es una clase que incluye dos funciones:

- realizarLlamada, genera la llamada mediante el Action:Originate haciendo uso de los parámetros que se le pasan. Devuelve 0 si la llamada se realizó con éxito o 1 si no fue así.
- PresionaCinco, devuelve 1 si el cliente presionó 5 al escuchar el mensaje o 0 si no fue así.

#### administradorCobranzas.php

El siguiente archivo, ubicado en la ruta `/var/www/html/`, es la aplicación web (programa cliente) desde la cual se ejecutan los scripts `amiCanal#.php`, `ami2Canal#.php` y `generaInforme.php`.

#### generaInforme.php

El siguiente script, ubicado en la ruta `/var/www/html/`, utiliza la clase `class.ezpdf.php` para generar un informe, en formato pdf, de todas las llamadas realizadas a los clientes. Este informe será entregado a la organización.

Después de la instalación de los componentes necesarios para la implementación de este proyecto, de configurar correctamente los archivos de Asterisk y de desarrollar los scripts que interactúan con la central telefónica se procede a verificar el correcto funcionamiento del proyecto tanto para las llamadas simultáneas a diferentes clientes, como el procedimiento que se sigue al intentar contactarse con un cliente.

Presionando el botón para generar las llamadas en la aplicación web del sistema se generan simultáneamente las llamadas por los cuatro canales disponibles de la tarjeta de telefonía, a cuatro clientes diferentes.

En la Figura 2 se observa como una de las llamadas a un cliente resulta exitosa, pero es rechazada inesperadamente, por lo que el sistema vuelve a llamar al mismo número por segunda ocasión.

```
root@host-200-126-13-189:/etc/asterisk/outboundCall - [x]
Archivo Editar Ver Terminal Solapas Ayuda
root@host-200-12... x root@host-200-12... x root@host-200-12... x
[root@host-200-126-13-189 outboundCall]# php amiCanal3.php
resp: Asterisk Call Manager/1.1
Response: Success
Message: Authentication accepted

resp: Response: Error
resp: ActionID: llamada1
Message: Originate failed

Event: Hangup
Privilege: call_all
Channel: DAHD1/3-1
Uniqueid: 1307007853.1
CallerIDNum: <unknown>
CallerIDName: <unknown>
Cause: 0
Cause-txt: Unknown

resp: Response: Success
resp: ActionID: llamada1
Message: Originate successfully queued
```

**Figura 2:** Eventos generados por AMI al realizarse una llamada exitosa y colgarse

En la Figura 3 se muestra como al escuchar las opciones y presionar 3, el cliente vuelve a escuchar el archivo de audio con sus datos

## 7. Resultados de Funcionamiento





# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



- Se aprovechan en gran medida los recursos de la empresa al poder realizar llamadas simultáneas a diferentes clientes, pudiendo efectuar mayor número de llamadas en el día con la misma efectividad que si se contactara a un cliente a la vez.
- El sistema está desarrollado para que se vuelva a llamar una vez más al mismo número, en caso de que la llamada sea interrumpida mientras se reproduce el mensaje. Esto puede ser fácilmente modificado de acuerdo a las necesidades de la empresa.

## 9. Recomendaciones

- Determinar las características de los equipos servidores en función del número de usuarios y tipo de servicios.
- Recordar cambiar el código de provincia para llamadas locales según la ciudad donde se quiera utilizar el sistema.
- La información de los clientes que se mantiene en la base de datos, debe ser actualizada mensualmente.
- Se recomienda como trabajo a futuro implementar casillas de buzón de voz para los clientes, de tal forma que, aun cuando no se pueda contactar con el cliente, se le pueda dejar un mensaje, que podrá ser revisado posteriormente.

## 10. Referencias

- [1] Elastix, Voz sobre ip en Ecuador, Disponible en:  
<http://listas.asle.ec/pipermail/asociacion/2009-October/012083.html>.  
Octubre 2009.
- [2] Wikipedia, Asterisk. Disponible en:  
<http://es.wikipedia.org/wiki/Asterisk>.  
Junio 2011.
- [3] 3cx, ¿Qué significan los términos FXS y FXO? Disponible en:  
<http://www.3cx.es/voip-sip/fxs-fxo.php>  
Enero 2011.
- [4] Blogs.digium.com, Zaptel project being renamed to DAHDI, Disponible en:  
<http://blogs.digium.com/2008/05/19/zaptel-project-being-renamed-to-dahdi/>  
Enero 2011.