

# Diseño e Implementación de un Sistema de Distribución Automática de Llamadas Entrantes (ACD) para un Callcenter

Luis Miguel Andino Montalvo<sup>(1)</sup>, Javier Alejandro Urquiza Guevara<sup>(2)</sup>, Gabriel Antonio Astudillo Brocel<sup>(3)</sup>  
Facultad de Ingeniería en Electricidad y Computación<sup>(1) (2) (3)</sup>  
Escuela Superior Politécnica del Litoral  
Campus “Gustavo Galindo V.”, Km. 30.5, vía Perimetral  
Apartado 09-01-5863, Guayaquil, Ecuador  
[landino@espol.edu.ec](mailto:landino@espol.edu.ec)<sup>(1)</sup>, [jaurquiz@espol.edu.ec](mailto:jaurquiz@espol.edu.ec)<sup>(2)</sup>, [gastudil@espol.edu.ec](mailto:gastudil@espol.edu.ec)<sup>(3)</sup>

## Resumen

*El presente trabajo es una implementación de una aplicación para un centro de contacto. Consiste en la distribución de las llamadas entrantes al mismo, en base a parámetros estadísticos obtenidos de los registros de las llamadas realizadas al mismo centro de contacto.*

*La implementación se la realizara mediante Asterisk, una PBX implementada en software y de código abierto. Además se utilizaran programas adicionales como motores de bases de datos, códigos en PHP, servidores web, entre otros.*

*El proyecto busca mejorar la calidad del servicio de un centro de contacto, usando recursos eficientemente. Implementando características diferentes a las que se encuentran disponibles en Asterisk por defecto, con el objetivo de poder brindar servicios a usuarios con privilegios, minimizar tiempo de espera de clientes y dirigir las llamadas a los agentes más idóneos.*

**Palabras Claves:** *callcenter, VoIP, distribución automática de llamadas, agentes, políticas de redireccionamiento, colas, bases de datos, MySQL, Asterisk.*

## Abstract

*This paper is an implementation of an application for a callcenter. It consists of the distribution of incoming calls, based on statistical parameters obtained from the details of calls made to the callcenter.*

*The implementation will be using Asterisk software implemented PBX and also open source. Also additional programs were used like database engines, PHP code, and Web servers, among others.*

*The project aims to improve the quality of service of a callcenter, using resources efficiently. Implementing different characteristics from those that are available in Asterisk by default, in order to provide services to privileged users, minimize customer wait time and direct calls to the most appropriate agents.*

## Introducción

Hoy en día el mundo requiere gran cantidad de información y las comunicaciones se han vuelto imprescindibles para el desarrollo de las empresas, mediante la comunicación con sus clientes. Una de las aplicaciones que permiten lograr esto son los centros de contacto.

Estos centros de contacto dependen mucho de la capacidad y del conocimiento de la persona contestando el teléfono, así como su habilidad para resolver problemas del usuario y satisfacer sus necesidades.

Por lo tanto para mejorar y tener un sistema más eficiente, el centro de contacto debe dirigir sus llamadas a los agentes con mejor desempeño, para minimizar tiempos de llamado y aprovechar habilidades de los agentes.

El objetivo de éste trabajo es el de implementar un sistema de distribución automático de llamadas entrantes para un centro de contacto con la posibilidad de manejar diferentes políticas eficientemente.

## 1. Antecedentes

La actualidad de las comunicaciones y los servicios van de la mano con la tecnología y la rapidez con las que se realizan. Por lo tanto, es necesario estar al tanto de las nuevas tecnologías y optimizar recursos a través de la implementación de sistemas eficientes.

Con el transcurso de los años las tecnologías crecen aceleradamente, las empresas van actualizando sus sistemas de comunicaciones. El uso de las PBX, los centros de contacto y los sistemas de distribución de llamadas utilizando las redes PSTN van quedando atrás, dando paso a las tecnologías VoIP.

En los primeros años de la década del 2000 aparece Asterisk, actualmente el más popular proyecto de comunicación implementado en código abierto y que funciona sobre redes VoIP.

Asterisk es capaz de convertir un ordenador en un complejo servidor de comunicación de voz a través del protocolo IP con excelentes características y servicios telefónicos como: centros de contacto, sistemas de respuesta interactiva de voz, puertas de enlace de voz sobre IP, entre otras.

Para la implementación de este proyecto se seleccionó Asterisk como servidor de comunicaciones por las características antes mencionadas y por la capacidad de compatibilidad con diferentes herramientas de

desarrollo como PHP y MySQL para manejo de información a través de bases de datos.

## 2. Objetivos

### 2.1. General

Diseñar e implementar un sistema de distribución automática de llamadas entrantes (ACD) para un Centro de Contacto, utilizando un algoritmo basado en información estadística para decidir la prioridad de asignación de llamadas a agentes conectados. El algoritmo considerará el tiempo de conexión del agente, el promedio de la duración de las llamadas tomadas por el agente en el día, el tiempo de espera del cliente en cola y la prioridad de la llamada basada en el identificador de la llamada entrante del emisor.

### 2.2 Específicos

- Implementar soluciones de Voz sobre IP como aplicación a una necesidad real en el mundo laboral.
- Lograr una fácil transición a un sistema implementado sobre una red VoIP a través de un sistema amigable.
- Acoplar diferentes programas y servicios para proveer un sistema eficiente.

## 3. Marco Teórico

### 3.1 Asterisk

En su definición más simple, Asterisk es una implementación en software de una PBX. Una PBX (Private Branch eXchange por sus siglas en inglés) es una central telefónica privada, que maneja o administra llamadas entrantes y salientes en una empresa.

Asterisk fue originalmente desarrollado para el sistema GNU/Linux. Hoy en día funciona en varios sistemas operativos incluyendo Microsoft Windows, Mac OSX, Solaris, BSD, entre otros. Aun así es en las distribuciones Linux que se popularizó como un software de código abierto con capacidades de dar soluciones de PBX. [1]

Entre las funcionalidades más importantes de Asterisk, cabe destacar aquellas funcionalidades que serán de gran utilidad en el desarrollo del proyecto; el encolamiento de llamadas, los detalles de los registros de llamadas, el uso de agentes y las funcionalidades para llamadas en espera como: llamadas con música en espera, música en transferencia, llamada en espera, identificador de llamada y transferencia de llamadas. [2]

Finalmente, Asterisk es ampliamente utilizado por su capacidad de interoperabilidad con varios lenguajes de programación, permitiendo que pueda ser administrado desde diferentes fuentes. Entre otros puede ser administrado a través de C, C++, Java, PHP y MySQL.

### 3.1 Interfaz de puerta de enlace de Asterisk (AGI)

La puerta de enlace de Asterisk, provee un estándar mediante el cual diferentes lenguajes de programación como PHP, Pearl, Python, etc. pueden controlar el plan de marcado y las extensiones que se encuentren configuradas. Esto la convierte en una interfaz muy útil en el mundo laboral, en la cual no es necesario entrar en la consola de Asterisk para controlar las acciones del plan de marcado, lo cual constituye una ventaja para los usuarios del sistema, pues mediante la misma interfaz que usan a diario es posible incorporar Asterisk sin alterar su entorno de trabajo.

La clave del funcionamiento de la interfaz AGI es la forma en la que se comunica con Asterisk, usando los estándares STDIN, STDOUT y STDERR. Estos son utilizados por la mayoría de lenguajes de programación para manejo de archivos. STDIN maneja la entrada de los datos ya sea por teclado o por alguna aplicación. STDOUT, involucra el envío de datos a una aplicación. STDERR, se encarga de los mensajes de error. [3]

En este proyecto, AGI será de gran utilidad, pues permitirá el almacenamiento de los datos de llamadas, agentes, colas, etc. en bases de datos para luego mediante consultas, poder determinar, al agente al cual hay que dirigir la llamada, esto dependiendo de la política vigente al momento de ejecución. Además permitirá verificar si el usuario de llamada entrante se encuentra registrado como usuario con privilegios o no.

### 3.2 Distribuidor Automático de Llamadas (ACD)

Es un dispositivo que distribuye llamadas entrantes a un grupo específico de agentes del sistema. Es muy común encontrar esta aplicación cuando se llama a grandes empresas, para redirigir llamadas a servicio técnico, secretaria, etc. [4]

En el caso de este trabajo, la distribución de llamadas será valorada en función a parámetros fijados como tiempo de conexión del agente, duración de la última llamada tomada por el agente, tiempo de espera del

cliente en la cola y prioridad de la llamada basada en la identificación del emisor.

### 3.3 Centro de Contacto (Callcenter)

Generalmente se encuentran en compañías proveedoras de servicios y que se encargan de proveer asistencia a clientes sobre los productos, servicios o cualquier información adicional que consideren relevante. Las llamadas pueden ser entrantes o salientes, desde o hacia socios comerciales, compañías asociadas, clientes potenciales, etc.

Generalmente, las personas encargadas de recibir la llamada, deben entender la necesidad del que llama, atender su requerimiento de la mejor manera o, redirigirlo a la persona más idónea para ayudarlo si no está capacitado para resolver su inquietud.

## 4. Implementación

### 4.1 Hardware

El proceso de selección del hardware a utilizar debe ser realizado con cautela. El servidor debe cumplir por lo menos con los requerimientos mínimos de para los servicios que va a brindar.

El esquema de la figura 1 muestra el hardware que se va a utilizar en la implementación. Para efectos de prueba se utilizarán 3 agentes y 3 clientes para todas las políticas. El servidor que se aprecia en la figura es el que posee todos los servicios que se van a utilizar.

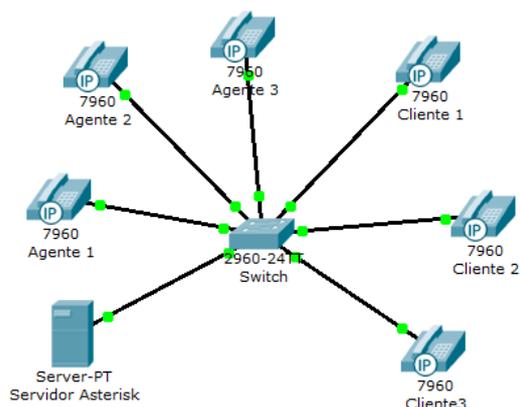


Figura 1: Esquema de Hardware a utilizar

**4.1.1 Servidor.-** En la implementación del servidor se trabajó con tres servicios en el mismo ordenador. El servidor Asterisk, el servidor de MySQL y el servidor web Apache. Las características del computador se detallan en la tabla 1.

**Tabla 1: Características del servidor**

CPU	RAM	Disco Duro
Intel Core 2 Duo 2.66GHz	2GB	300GB

**4.1.2 Hardphone.-** Para realizar las pruebas del proyecto se utilizó un teléfono IP, el cual tiene la capacidad de funcionar en redes VoIP. El teléfono utilizado es de marca GrandStream modelo GPX2000. Fue utilizado por la disponibilidad del equipo al momento de la implementación.

## 4.2 Software

De la misma manera que el hardware, el software es de vital importancia para la implementación del sistema. En la tabla 2 se detallan las versiones del sistema operativo utilizado en la implementación con su respectiva distribución. De la misma manera se describen las versiones de los servidores Asterisk, MySQL, Apache y el complemento PHP para trabajar con AGI.

**Tabla 2: Software utilizado en el proyecto y sus versiones**

<b>Sistema Operativo</b>	Linux, Distribución CentOS 5.5
<b>Software PBX</b>	Asterisk versión 1.8.3.2
<b>Motor base de datos</b>	MySQL Server 5.0.8
<b>Servidor Web</b>	Httpd-2.2.3-53.el5.centos.3
<b>PHP</b>	PHP 5.1.6-27.el5_5.3

**4.2.1 Softphone.-** Un softphone es una aplicación diseñada para permitir la comunicación a través de una red VoIP, de la misma manera que un equipo de teléfono IP.

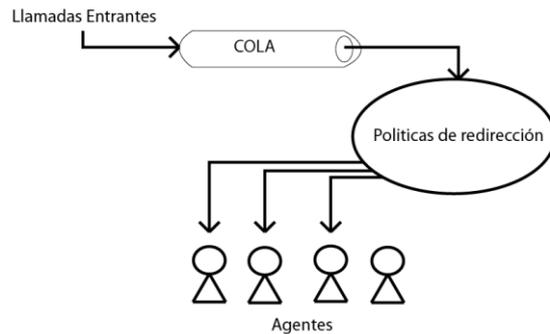
El programa escogido para la realización del proyecto fue el Zoiper free versión 2.37. El mismo fue escogido por la familiaridad que se tenía con el software de experiencias previas, además brindaba una interfaz amigable que permitía configurar rápidamente y con facilidad las interfaces SIP e IAX a diferencia de otros softphones.

**4.2.2 Qloader.-** Es un script que migra la información contenida en el archivo de texto queue\_log.txt a una base de datos para ser almacenados en tablas. Este archivo es de vital importancia para el proyecto pues contiene la información de los eventos ocurridos en la cola y se actualiza en tiempo real.

El qloader es un script desarrollado en el lenguaje de programación Pearl, utilizado por el software Queuemetrics, el cual monitorea llamadas para centros de contactos basados en Asterisk. [5]

## 4.3 Implementación

La figura 2 muestra los pasos a seguir al momento de la implementación. En primer lugar las llamadas entrantes ingresarán a una cola, en la cual irán almacenándose en el orden de su llegada e irán saliendo en ese mismo orden. Al momento de salir las llamadas pasaran por un proceso de redirección en el cual, dependiendo de la política vigente, se redirigirán a un único agente. Este agente deberá estar conectado y disponible para recibir llamadas.



**Figura 2: Esquema de Implementación**

**4.3.1 Elección de la política de redirección.-** La elección de la política de redirección se realiza mediante una aplicación web. En esta aplicación, se debe seleccionar una de las 4 políticas disponibles. Al momento de la selección, la aplicación web envía los datos de la opción elegida a una base de datos para almacenamiento de la información en la tabla opc. Esta tabla será consultada varias veces durante el desarrollo de la implementación, por medio del archivo eleccion.php para verificar si ha habido un cambio en la política vigente.

**4.3.2 Inicio de sesión de los agentes.-** Para el inicio de sesión de los agentes es necesario un número de identificación del agente y su contraseña. Estos parámetros deberán ser configurados en el archivo agents.conf con el siguiente formato:

Agent=> agentid, agentpassword, name

El archivo de configuración `agents.conf` es aquel que rige el comportamiento de los agentes en Asterisk.

Una vez realizadas las configuraciones respectivas en el archivo `agents.conf`, se debe añadir una extensión en el archivo `extensions.conf` siguiendo el siguiente formato:

```
exten=> 1001,1,AgentLogin()
```

Esta extensión será aquella que llamarán los agentes para iniciar sesión en el programa. En ese momento los agentes escucharán la petición del identificador del agente y de la contraseña los cuales deberán ser ingresados mediante el teclado del teléfono. Luego de ser autenticados los agentes estarán disponibles para recibir llamadas.

El archivo `extensions.conf` es aquel que controla la operación y el comportamiento de la PBX, es decir contiene el plan de marcado según el cual las extensiones se van a comunicar. [6]

**4.3.3 Encolamiento de las llamadas.-** Es el proceso mediante el cual se coloca las llamadas en una cola al momento de marcar al centro de contacto. Al momento de ingresar las llamadas a las colas se inactivarán a todos los agentes conectados que no se encuentren contestando una llamada. Esto es debido a que al momento de aplicar la política solo se activará al agente elegido para contestar la llamada.

El comportamiento de las colas es regido por el archivo de configuración `queues.conf`. En éste archivo se configurará el nombre de la cola, el peso y los miembros de la misma. El nombre y el peso de la cola son muy importantes, pues en la implementación del proyecto se tendrán 2 colas, la primera con un peso mayor para aquellos usuarios con una prioridad alta en base al identificador de llamada entrante, la segunda con un peso menor para usuarios con prioridad normal.

Además es importante recordar que por definición el comportamiento de la cola es FIFO, lo que quiere decir “first in first out”, que por su traducción al español significa el primero que entra es el primero en salir. Con esta premisa se cumple el objetivo de que la redirección se base en el tiempo de espera del cliente en cola, pues aquel cliente cuya llamada ingrese a la cola primero será atendido primero.

**4.3.4 Captura de detalle de llamadas.-** Esta sección del proyecto captura los datos y variables de las llamadas para ser utilizadas en el algoritmo de redirección.

Para el envío de información sobre las llamadas es necesario configurar el archivo `cdr_mysql.conf` en el

cual se detallan parámetros como el nombre de la base de datos a utilizar, la contraseña de la misma y el usuario que tiene acceso.

Mediante ésta configuración se creará la tabla `cdr`. Esta tabla proporciona toda la información de las llamadas realizadas como la fecha, el origen, canal y destino de la llamada, así como también la duración de las mismas.

Esta sección utiliza además un script que es parte del software `Queuemetrics` para conversión de la información de las colas desde el archivo `queue_log.txt` a la base de datos en la tabla `queue_log`.

La tabla `queue_log` es fundamental para las políticas de redirección cuyo fin es utilizar los criterios de elección de agente que tienen que ver con el tiempo de conexión del mismo, mediante los campos `time_id`, `agent` y `verb`. [7]

La tabla `cdr` principalmente, será de gran utilidad para el proyecto en las políticas de redirección que tengan como objetivo utilizar el criterio de elección de agente por promedios de llamada, en cambio la tabla `queue_log` tendrá inherencia en las políticas que tengan como parámetro principal el tiempo de conexión del agente.

**4.3.5 Filtrado de llamada por identificador de llamada entrante.-** Esta sección envía las llamadas entrantes de usuarios con prioridad a la cola con un peso mayor, en cambio envía las llamadas entrantes de usuarios sin prioridad a una cola con un peso menor.

La designación de usuarios con prioridad se realiza mediante una aplicación web en la que se registran los usuarios “vip” que únicamente será utilizada por el administrador. De la misma manera que con las políticas, los usuarios vip son almacenados en una base de datos en una tabla llamada `vip`.

Esta tabla será consultada cada vez que ingrese una llamada al centro de contacto para enviarla a la cola correspondiente. Esta acción será realizada por el archivo `usuariovip.php`. Con esta acción se cumple el objetivo de dar prioridad a llamadas por medio del identificador de llamada entrante si pertenece a un usuario vip.

**4.3.6 Redirección de llamada al agente elegido.-** Esta sección se encargará de activar únicamente al agente que ha sido seleccionado en base a la política vigente para recibir la llamada.

El archivo `mayortconectado.php`, se encarga de la opción en la cual se elige al agente que tiene mayor tiempo conectado en el sistema. Para esto la consulta

debe devolver el agente y el tiempo de conexión que lleva el mismo, lo cual debe encontrarse en la tabla log\_queue.

Un inconveniente es que la tabla log\_queue contiene toda la información de los eventos ocurridos en la cola. Por lo que hay información del inicio de sesión de los agentes (AGENTLOGIN), el cierre de sesión de los agentes (AGENTLOGOFF), cuando se contesta una llamada (CONNECT), etc. Por lo cual hay que elegir los agentes que han iniciado sesión, pero que no hayan aun cerrado la sesión y que no se encuentren actualmente en una llamada.

Para lograr esto se implementó un código de consulta, que devuelve una lista de los agentes que están conectados, pero ordenados en orden ascendente considerando el parámetro time\_id. Debido a que no hay un parámetro de tiempo de conexión del agente en la tabla queue\_log se tomo el parámetro time\_id para obtener el tiempo de conexión. El parámetro time\_id se escribe con un identificador de la hora a la que se conectó el agente, con lo cual el agente que tenga el menor time\_id es aquel que se registró primero, por lo tanto el que lleva más tiempo conectado. Es por esto que se ordena de menor a mayor para obtener el agente de mayor tiempo conectado.

De la misma manera con el archivo menortconectado.php, pues en esencia es la misma consulta que el archivo mayortconectado.php, solo que en lugar de elegir el agente que tiene más tiempo conectado, se elige al agente que menos tiempo tiene conectado. Para esto se usa un código de consulta muy similar al anterior, pero se ordenan los agentes en forma descendente, pues el agente que menos tiempo tiene conectado es aquel que tiene el mayor time\_id.

En cambio los archivos mayorpromedio.php y menorpromedio.php se basan en el promedio de todas las llamadas recibidas por los agentes en el día. La consulta que debe realizarse es en base a los agentes que se encuentran disponibles en la tabla log\_queue como se explicó anteriormente. Cada agente disponible se buscó como destino en la tabla cdr, la cual contiene los registros de todas las llamadas, para obtener el promedio de todas las llamadas que han sido realizadas a cada agente, incluyendo en la búsqueda que solo aparezcan las llamadas del día actual. De la misma manera se ordena en orden ascendente para elegir el mayor promedio y de forma descendente para el menor promedio.

## 5. Pruebas

En esta sección se presentan las pruebas y el detalle de las mismas, para el correcto funcionamiento del centro

de contacto utilizando tres agentes y tres clientes, como se especificó anteriormente.

El proceso de la prueba inicia con la elección de la política por parte del administrador del programa y el inicio de sesión de los agentes para contestar las llamadas entrantes.

Al momento que ingrese una llamada, se hará un proceso de filtrado del identificador de la misma para verificar si se trata de un usuario con prioridad o no. Este proceso se realiza mediante una verificación de la base de datos de usuarios vip. En caso de ser usuario vip con privilegios se encolará la llamada en la cola con mayor peso, caso contrario se encolará en la cola con menor peso.

Una vez que se encuentren encoladas las llamadas se inactivarán a todos aquellos agentes que se encuentren conectados y que no estén contestando una llamada al momento.

Luego pasará al proceso de selección del agente por medio de la interfaz AGI y un archivo escrito en PHP. Dependiendo de la política elegida y del detalle de las llamadas actual se elegirá un solo agente responsable de contestar cada llamada, con los códigos de consulta antes explicados. En caso de no haber ningún agente disponible para contestar la llamada se mantendrá en espera hasta que algún agente se desocupe.

La tabla 3 muestra todas las opciones de políticas que se tiene en el proyecto, con su respectiva extensión en el plan de marcado que contiene la descripción del comportamiento que va a tener para cumplir con esa política y el archivo que realiza el redireccionamiento mediante el código de consulta a la extensión.

**Tabla 3: Tabla de elección de agentes**

Opción	Extensión	AGI	Descripción
1	211	mayortconectado.php	Escoge al agente que más tiempo tiene conectado
2	212	menortconectado.php	Retorna al último agente en conectarse
3	213	mayorpromedio.php	Devuelve el agente con mayor promedio de tiempo en llamadas
4	214	menorpromedio.php	Muestra el agente con menor promedio en sus llamadas

## CONCLUSIONES

Se cumplió el objetivo principal de implementar el centro de contacto utilizando Asterisk como software de PBX en conjunto con diversas herramientas como bases de datos, servidores web, códigos en PHP y un script parte del software Queuemetrics. Mediante las bases de datos y la interfaz AGI combinada con códigos PHP se logró dirigir las llamadas entrantes en base a los parámetros planteados.

En base a las pruebas realizadas se determinó que el proyecto puede ser aplicado en diversas empresas con diferentes políticas, con lo cual cubre una necesidad importante en el mundo ya que puede ser utilizado en varios sistemas de consulta de información y atención al cliente. Gracias a la cantidad de parámetros disponibles en las tablas sobre el detalle de las llamadas, es posible implementar diferentes políticas cambiando únicamente el código de consulta dependiendo de la política deseada.

El sistema es amigable al uso de los agentes, por lo que facilita la transición a la tecnología VoIP, pues no requiere un tiempo de capacitación prolongado para uso del sistema. El programa le indica de forma auditiva los pasos para inicio de sesión del agente, advertencia ante errores, etc.

El sistema permite la utilización de diferentes políticas, por lo cual la eficiencia de la atención de los clientes depende de la elección de las mismas por parte de la empresa que implemente esta solución planteada.

## RECOMENDACIONES

Usar programas que emulen teléfono IP (Softphones), lo que permitiría ahorrar en costo de adquisición de equipos. Además mediante el uso de diademas permite mantener un buen ambiente reduciendo el ruido causado por el timbrar de los teléfonos.

Utilizar aplicaciones de código abierto, los cuales abaratan un poco el costo de implementación. Los programas utilizados en la implementación de este trabajo son de código abierto y no representaron un costo adicional adquirirlos pues todos son gratuitos.

Reducir estrés producido por tiempo de espera del cliente, reproduciendo música menos monótona como el tono de marcado y anunciando a los usuarios que sus inquietudes serán atendidas en brevedad.

Revisar el uso de herramientas para manejo de datos en tiempo real de Asterisk. Las herramientas presentaron problemas en la versión 1.8 de Asterisk

por una incompatibilidad en cuanto al tipo de datos de la tabla `queue_log`.

Plantearse AMI como solución alternativa para este trabajo, pues se presenta como una poderosa interfaz. No fue utilizada en esta implementación por la necesidad de utilizar bases de datos y la facilidad de la interfaz AGI para la utilización de las mismas mediante códigos PHP.

Mejorar la eficiencia del sistema utilizando la normalización de las tablas en las bases de datos. La normalización disminuye la redundancia de los datos y agiliza la búsqueda de información.

## BIBLIOGRAFÍA

[1] WIKIPEDIA, "Definición de Asterisk" <http://es.wikipedia.org/wiki/Asterisk>, fecha última visita: julio del 2011.

[2] Paul Mahle, VoIP Telephony with Asterisk, Signate, LLC, Estados Unidos, 2004, pp. 4-18.

[3] Jim Van Meggelen, Leif Madsen, Jared Smith, "Asterisk: The Future of Telephony", O'Reilly, Estados Unidos, 2007, pp 119-124, 207-225.

[4] WIKIPEDIA, "Definición de ACD", [http://en.wikipedia.org/wiki/Automatic\\_call\\_distributo\\_r](http://en.wikipedia.org/wiki/Automatic_call_distributo_r), fecha última visita: julio del 2011.

[5] QUEUEMETRICS, "Definición Qloader", <http://queuemetrics.com/manuals/QLoader-chunked/>, fecha de última visita: diciembre del 2011.

[6] Flavio E. Gonçalves, Configuration Guide for Asterisk™ PBX, V. Office Networks Ltda, 2007, pp. 68.

[7] Voip-Info.org, "Información sobre `queue_log`", [http://www.voip-info.org/wiki/view/Asterisk+log+queue\\_log](http://www.voip-info.org/wiki/view/Asterisk+log+queue_log), fecha de última visita: noviembre del 2011.