

ESCUELA SUPERIOR POLITECNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación
“Diseño e Implementación de un Sistema de Distribución
Automática de Llamadas Entrantes (ACD) para un
Callcenter”

INFORME DE MATERIA DE GRADUACIÓN

Previa a la obtención del Título de:

INGENIERO EN TELEMÁTICA

Presentada por:

LUIS MIGUEL ANDINO MONTALVO

JAVIER ALEJANDRO URQUIZO GUEVARA

GUAYAQUIL-ECUADOR

AÑO

2012

AGRADECIMIENTO

A Dios padre, por todas las bendiciones que nos otorga cada día, salud a nuestras familias y la oportunidad de crecer en conocimientos y como personas.

A nuestras familias y amigos por el amor, cariño y apoyo incondicional para cumplir con nuestras metas y objetivos planteados.

Agradecemos a nuestros profesores que con paciencia, supieron enriquecer nuestros conocimientos con su sabiduría y experiencia.

DEDICATORIA

Dedico este trabajo a Dios que a sido mi motivación para seguir adelante. A mis padres y hermanos que siempre han estado a mi lado apoyándome e impulsándome para terminar este proyecto lo más rápido posible.

Luis Miguel Andino Montalvo

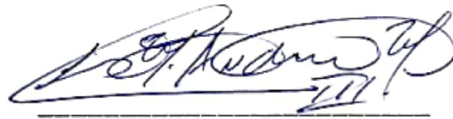
A Dios, por darme la vida y tantas bendiciones. A mis padres por ser ejemplo de personas y modelos a seguir, a mis hermanos y mi familia por el apoyo incondicional y a mi enamorada y mis amigos por todo lo que compartimos en el día a día.

Javier Urquizo Guevara

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este proyecto 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)



Luis Miguel Andino Montalvo

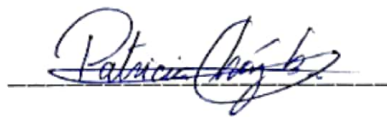


Javier Alejandro Urquizo Guevara

TRIBUNAL DE SUSTENTACIÓN

Ing. Gabriel Astudillo Brocel

PROFESOR DE LA MATERIA DE GRADUACIÓN



A handwritten signature in blue ink, reading "Patricia Chávez Burbano", is written over a horizontal line.

Ing. Patricia Chávez Burbano

PROFESOR DELEGADO POR EL DECANO DE LA
FACULTAD

RESUMEN

Este proyecto consistió en la realización de una central de contacto con distribución automática de llamadas (ACD, por sus siglas en inglés) sobre Asterisk, Software usado para crear centrales telefónicas, que permite distribuir las llamadas entrantes a los agentes con mejores características, distinguiendo tipos de usuario.

El proyecto buscaba mejorar la calidad del servicio de un centro de llamadas, usando recursos eficientemente. Implementando características diferentes de las que se encuentran en Asterisk para una central telefónica con ACD, tales como: Brindar servicios de usuarios especiales, minimizar tiempo del cliente, aprovechar las características de los agentes.

En la implementación se alcanzó el resultado deseado usando programas adicionales como motores de base de datos, código php y los conocimientos adquiridos en la materia de graduación.

ÍNDICE DE CONTENIDO

AGRADECIMIENTO	I
DEDICATORIA	II
RESUMEN	V
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS	XI
TABLA DE ABREVIATURAS	XII
INTRODUCCIÓN	XIII
CAPÍTULO 1: ANTECEDENTES Y JUSTIFICACIÓN	1
1.1 ANTECEDENTES	1
1.2 JUSTIFICACIÓN	3
1.3 OBJETIVO GENERAL.....	4
1.4 OBJETIVOS ESPECÍFICOS	4
1.5 METODOLOGÍA.....	6
1.6 PERFIL DE LA TESIS	7
CAPÍTULO 2: ASTERISK	9
2.1 ASTERISK.....	9
2.1.1 FUNCIONALIDADES DE ASTERISK.....	11
2.2 CENTRO DE LLAMADAS	12
2.3 DISTRIBUIDOR AUTOMÁTICO DE LLAMADAS (ACD)	13
2.4 INTERAZ DE PUERTA DE ENLACE DE ASTERISK (AGI)	14
CAPÍTULO 3: IMPLEMENTACIÓN	16
3.1 HARDWARE.....	17
3.1.1 SERVIDOR.....	17
3.1.2 HARDPHONES	18

3.2 SOFTWARE	19
3.2.1 SOFTPHONES.....	20
3.2.2 QLOADER.....	20
3.3 INSTALACIÓN.....	21
3.3.1 INSTALACIÓN LIBRERIAS BASE	21
3.3.2 INSTALACIÓN ASTERISK.....	23
3.4 CONFIGURACIÓN DE ARCHIVOS	25
3.4.1 CONFIGURACIÓN SIP.CONF	25
3.4.2 CONFIGURACIÓN EXTENSIONS.CONF	26
3.4.3 CONFIGURACIÓN AGENTS.CONF	27
3.4.4 CONFIGURACIÓN QUEUES.CONF	28
3.4.5 CONFIGURACIÓN CDR_MANAGER.CONF	31
3.4.6 CONFIGURACIÓN CDR_MYSQL.CONF.....	32
3.4.7 PAGINAS WEB INDEX.HTML, USERS.HTML.....	33
3.4.8 ARCHIVOS ELECCION.PHP, USUARIOVIP.PHP	36
3.4.9 ARCHIVOS PHP	38
CAPÍTULO 4: FUNCIONAMIENTO Y PRUEBAS.....	45
4.1 INICIALIZANDO ASTERISK.....	46
4.2 CONFIGURACIÓN DE LA BASE DE DATOS	47
4.2.1 TABLA CDR	47
4.2.2 TABLA QUEUE_LOG.....	48
4.2.3 TABLA VIP	49
4.2.4 TABLA OPC	50
4.3 PRUEBA DE LLAMADA AL CALL CENTER	51
4.4 ESPECIFICACIONES DE AGI	54
CONCLUSIONES Y RECOMENDACIONES	56
CONCLUSIONES.....	56
RECOMENDACIONES	58
ANEXOS	

ANEXO A

ANEXO B

ANEXO C

ANEXO D

ANEXO E

ANEXO F

ANEXO G

ANEXO H

ANEXO I

ANEXO J

ANEXO K

ANEXO L

ANEXO M

ANEXO N

ANEXO O

BIBLIOGRAFÍA

ÍNDICE DE FIGURAS

Figura 1.1 Esquema de implementación	6
Figura 3.1. Teléfono IP GPX2000	19
Figura 3.2. Esquema funcionamiento del sistema	30
Figura 3.3 Ejemplo de la información en la tabla log_queue	40
Figura 3.4 Códigos de consulta mayorconectado.php y menorconectado.php	42
Figura 3.5 Ejemplo tabla cdr	43
Figura 3.6 Códigos de consulta mayorpromedio.php, menorpromedio.php	44
Figura 4.1 Descripción de la tabla cdr	48
Figura 4.2 Tabla queue_log	49
Figura 4.3 Tabla vip	50
Figura 4.4 Pagina web para el ingreso de Usuarios vip	50
Figura 4.5 Tabla opc	51
Figura 4.6 Pagina web para el ingreso de políticas de decisión	51
Figura 4.7 Proceso de llamada al centro de contacto	52

Figura 4.8 Inicio de la llamada	53
Figura 4.9 Elección de AGI	53
Figura 4.10 Tipo de usuario	54

ÍNDICE DE TABLAS

Tabla I.	Características del Servidor	18
Tabla II.	Software relevante utilizado en el proyecto y sus versiones	20
Tabla III.	Versión del softphone	21
Tabla IV.	Tabla de elección de agentes	55

TABLA DE ABREVIATURAS

VoIP	Voz sobre IP
PBX	Central telefónica privada
ACD	Distribución automática de llamadas
IVR	Respuesta Interactiva de voz
IP	Protocolo de Internet
AGI	Interfaz de puerta de enlace de Asterisk
SIP	Protocolo de Inicio de Sesión

INTRODUCCIÓN

Hoy en día el mundo requiere gran cantidad de información y las personas obtenerla de varias formas, una de ellas es mediante centros de contacto.

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

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

Asterisk proporciona servicios de ACD direccionando llamadas hacia agentes, lo que permite minimizar costos de implementación e instalación, ya que se trata de un software gratuito.

CAPÍTULO 1

ANTECEDENTES Y JUSTIFICACIÓN

1.1 ANTECEDENTES

La actualidad de las comunicaciones y los servicios van de la mano y la tecnología y la rapidez con la que se realizan las actividades van marcando el prestigio, solvencia y eficiencia. Por lo tanto, es necesario

estar al tanto de las nuevas tecnologías y optimizar recursos. Por esto, se decidió implementar un sistema de comunicación eficaz para cualquier empresa y sus clientes.

En el transcurso de los años las tecnologías crecen aceleradamente, así como los costos incurridos en su adquisición. Sin embargo, existe lo que denominamos Código Abierto, software desarrollado y distribuido libremente, entre los cuales encontramos un programa llamado Asterisk, el más popular proyecto de comunicación Código Abierto. Este software es capaz de convertir una computadora ordinaria en un complejo servidor de comunicación de voz con excelentes características. Asterisk hace que sea simple crear y desarrollar una amplia gama de aplicaciones y servicios telefónicos, incluyendo IP PBXs, puertas de enlace de Voz sobre IP (VoIP), Centro de Contacto (ACD) y sistemas de Respuesta Interactiva de voz (IVR).

Asterisk nos da la opción de ser empleado como un Centro de Contacto gracias a su flexibilidad. Varios desarrolladores de centros de llamado y de contactos han construido sistemas ACD completos basados en el mismo. Asterisk también posee capacidades de agentes remotos IP,

habilidades de enrutamiento avanzadas, marcación predecible, y mucho más.

Una gran ventaja que tiene este tipo de soluciones de Código Abierto son los bajos costos de implementación y mantenimiento en comparación con los sistemas de Centro de Llamadas implementados tradicionalmente a través de centrales telefónicas propietarias. Otros de los grandes beneficios que posee Asterisk es la compatibilidad que tiene con diferentes programas y herramientas de desarrollo tales como C, C++, java, PHP, MySQL.

1.2 JUSTIFICACIÓN

Debido a los altos costos que producen las soluciones de PBX tradicionales, telefonía y centrales telefónicas, además de los recursos requeridos en hardware para una central de llamadas, surge la necesidad de buscar otras alternativas que reduzcan los gastos sin perder calidad en el servicio. Este es el caso de los Centros de Contactos de varias empresas. Una de éstas es utilizar software libre para la implantación en sistemas de VoIP.

Por consiguiente, este proyecto propone como solución para los Centros de Contacto, un sistema basado en datos de registros de llamadas, resulte más eficiente y económico para el desarrollo de cualquier empresa, utilizando Asterisk como PBX de Código Abierto.

1.3 OBJETIVO 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 las llamadas a agentes conectados.

1.4 OBJETIVOS ESPECÍFICOS

- Implementar soluciones de Voz sobre IP como aplicación a una necesidad real en el mundo laboral.
- Lograr que la transición de la red pública telefónica conmutada (PSTN) a Voz sobre IP sea fácil de manejar para los usuarios del centro de llamadas y a las empresas en general, a través de un sistema amigable.
- Acoplar los diferentes programas y servicios antes mencionados para proveer un sistema eficiente en cuanto a redirección de llamadas a través

de parámetros estadísticos, aplicables según las políticas de cualquier empresa.

El proyecto deberá implementar un sistema telefónico que permita recibir llamadas telefónicas a través de troncales IP y asignarlas automáticamente a los agentes que se encuentren conectados al sistema.

La prioridad en la asignación de las llamadas a los agentes se basará en la información estadística que la central telefónica IP recolectará. Además se implementará un algoritmo que 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 la cola y la prioridad de la llamada basada en la identificación de llamada entrante del emisor.

El esquema de la Fig. 1.1 define la implementación de este trabajo. El servidor de Asterisk será capaz de recibir llamadas desde teléfonos IP y desde un software que simula el funcionamiento de un teléfono IP, conocido como softphone. El servidor deberá decidir en base a

parámetros estadísticos almacenados en bases de datos a que agente debe ser redirigido y en qué momento será aceptada su llamada.

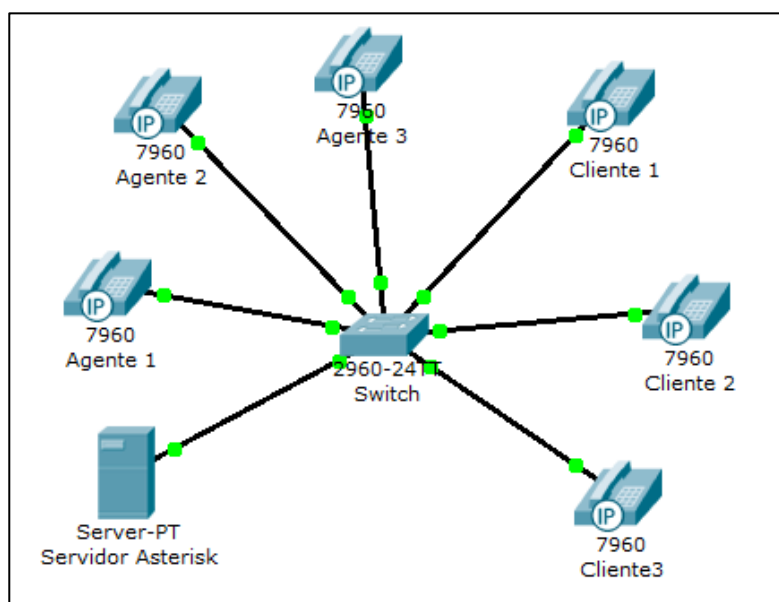


Figura 1.1 Esquema de implementación

1.5 METODOLOGÍA

Para cumplir los objetivos se realiza una instalación de Asterisk sobre un servidor con una distribución CentOS de Linux, utilizando MySQL como sistema de base de datos para almacenamiento y recolección de datos estadísticos correspondientes a las llamadas. Además se utilizará Apache como servidor web para almacenar una aplicación web que

servirá para seleccionar las políticas de redirección de llamadas. Finalmente se empleará Asterisk como software para la PBX integrando los servicios antes mencionados.

1.6 PERFIL DE LA TESIS

Este proyecto tiene como fin implementar un sistema ACD que dirija las llamadas entrantes a agentes conectados al sistema, en base a parámetros estadísticos obtenidos de llamadas anteriores almacenadas en bases de datos. Este sistema será aplicable a cualquier empresa y mejorará la eficiencia con la que se reciben las llamadas.

En el capítulo 2, se revisarán conceptos y fundamentos teóricos, características, mecanismos para la implementación, aplicaciones y servicios que brinda Asterisk. En el capítulo 3, se detallarán las especificaciones técnicas, análisis, diseño e implementación de la solución propuesta para el proyecto.

Finalmente en el capítulo 4, se detallarán las pruebas de conexión de agentes, almacenamiento de información en las bases de datos, elección

de los protocolos para redirigir llamadas, redirección de llamadas a agentes y orden de atención al cliente en base a los parámetros estadísticos propuestos.

CAPÍTULO 2

ASTERISK

2.1 ASTERISK

En su definición más simple, Asterisk es una implementación en software de una PBX. Una PBX (Private Branch Exchange por sus siglas en

ingles), 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 de Linux que se popularizó como un software de código abierto con capacidades de dar soluciones de PBX. [1]

Asterisk posee grandes ventajas sobre la tecnología de PBX convencionales, entre las que se destaca la diferencia económica que supone implementarlo. Asterisk resulta ser más económico debido a que puede ser instalado en un ordenador con suficientes recursos y lo convierte en un sistema de comunicaciones multi-uso para uso de la empresa. Adicionalmente al computador se requiere tener una conexión de internet, pues la voz será pasada a través de una red de datos usando VoIP. No está de más subrayar que, por su calidad de software libre, es muy atractivo para las empresas en términos financieros, quienes deben de tener en cuenta el costo del hardware que se usará para dicha instalación. [2]

Dentro del hardware necesario para poner en marcha cualquier sistema de comunicaciones, siempre serán indispensables puntos terminales que permitan comunicarse, en general los teléfonos, pues son aquellos los que permiten la comunicación entre los usuarios del sistema. Un sistema VoIP requiere teléfonos IP (equipo) o teléfono IP por software. Los teléfonos IP son aquellos que sirven para usar las redes de VoIP directamente. En cambio los teléfonos IP por software son aquellos que permiten comunicarse a través de software dentro de una red VoIP.

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

2.1.1 FUNCIONALIDADES DE ASTERISK

Entre las funcionalidades más importantes de Asterisk, cabe destacar aquellas funcionalidades que serán de gran utilidad para el desarrollo del proyecto, el encolamiento de las llamadas, los detalles de los registros de llamadas 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. Todas estas funcionalidades serán aplicadas y explicadas posteriormente en este trabajo.

Además de las ya mencionadas, también se destacan muchas funcionalidades que convierten a Asterisk en una poderosa PBX en software como: correo de voz con protección con contraseña, distinción de mensajes de ausente y no disponible, múltiples carpetas para correo y notificación de buzón de voz vía correo electrónico, respuesta de voz interactiva, listado mediante directorio, conferencias, llamada de usuarios locales y remotos, escalabilidad, concurrencia de varios protocolos, entre otras. [2]

2.2 CENTRO DE LLAMADAS

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.

2.3 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. [3]

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.

2.4 INTERFAZ DE PUERTA DE ENLACE DE ASTERISK (AGI)

Es 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 interfaz de 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. Además permite el manejo de datos relacionales con bases de datos como MySQL.

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. Éstos son utilizados por la mayoría de lenguajes de programación para manejar archivos. STDIN maneja la entrada de los datos ya sea por teclado o por alguna otra aplicación. STDOUT, involucra el envío de datos, que podría ser a una aplicación externa, como Asterisk. STDERR, se encarga de los mensajes de error. [4]

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, dependiendo de la política vigente y la estadística respectiva, el agente al cual hay que dirigir la llamada y también verificar si el usuario de llamada entrante se encuentra registrado como usuario vip o no.

CAPÍTULO 3

IMPLEMENTACIÓN

Este proyecto pretende brindar una solución que permita mejorar la eficiencia en la atención al cliente, direccionando llamadas empleando parámetros estadísticos, atendiendo a los clientes en orden de prioridad

basado en la identificación de llamada entrante del emisor y que sea aplicable a las políticas de cualquier empresa.

3.1 HARDWARE

El proceso de selección del hardware a utilizar debe ser realizado con cautela. El servidor debe cumplir con los requerimientos mínimos de hardware para soportar todos los requerimientos que va a tener y los servicios que va a brindar.

Sin embargo, con la tecnología y la capacidad de los ordenadores, hoy en día es más sencillo encontrar un computador que cumpla con el hardware requerido sin tener que invertir una gran cantidad de dinero.

3.1.1 SERVIDOR

Se trabajó con tres servicios en el mismo computador. El servidor de Asterisk, servidor de MySQL y el servidor web Apache. Las características del computador se detallan en la tabla I.

CPU	INTEL CORE 2 DUO 2.66 GHz
RAM	2 GB.
Disco Duro	300 GB.

Tabla I. Características del servidor

3.1.2 HARDPHONES

Para realizar las pruebas del proyecto se utilizó un teléfono IP de marca GrandStream modelo GPX2000, mostrado en la figura 3.1.



Figura 3.1. Teléfono IP GPX2000

3.2 SOFTWARE

De la misma manera que el hardware, el software es de vital importancia para la implementación del sistema. Ésta sección incluye todo el software utilizado en el proyecto.

En la tabla II se detallan las versiones del sistema operativo utilizado en la implementación del proyecto 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.

Sistema Operativo	Linux, Distribución Centos 5.5
Software PBX	Asterisk versión 1.8.3.2
Motor de base de datos	My Sql Server 5.0.8
Servidor Web	httpd-2.2.3-53.el5.centos.3
PHP	php 5.1.6-27.el5_5.3

Tabla II. Software relevante utilizado en el proyecto y sus versiones

3.2.1 SOFTPHONES

En la tabla III, se detalla la versión del teléfono IP por software utilizado para realizar pruebas de funcionamiento del sistema. Este programa fue escogido debido a la familiaridad que se tenía del software, pues fue utilizado durante las clases de preparación para la realización del proyecto para ejecutar pruebas.

Además el software era compatible con el sistema operativo, debido a que las pruebas finales de funcionamiento fueron realizadas en el sistema operativo Microsoft Windows 7. Comprobando así que, independiente del sistema operativo, el sistema VoIP implementado en Asterisk, funciona correctamente.

Softphone	Zoiper_free_2.37
-----------	------------------

Tabla III. Versión del Softphone

3.2.2 QLOADER

Es un script que migra la información contenida en el archivo de texto queue_log.txt a una base de datos en MySQL para ser almacenados

en tablas. Este archivo es de vital importancia para el proyecto pues contiene toda la información de los eventos ocurridos en la cola y se actualiza en tiempo real.

El qloader es un script desarrollado en Pearl, utilizado por la empresa Queuemetrics, la cual desarrolló un software que monitorea llamadas para centros de contacto basados en Asterisk.[5] Para descarga de este script se puede consultar en la sección de descargas de la web de Queuemetrics. [6]

3.3 INSTALACIÓN

En esta sección se revisará la instalación completa de los componentes de Asterisk 1.8.3.2, versión utilizada en este trabajo, instalación de librerías base, instalación de Asterisk en si y los componentes necesarios para la realización de este proyecto.

3.3.1 INSTALACIÓN LIBRERIAS BASE

Antes de poder instalar Asterisk en el ordenador, se debe verificar que se encuentren instaladas las librerías básicas que necesita para

funcionar. Debido a que la instalación se hará sobre un sistema operativo CentOS, el formato para instalación de los paquetes es:

```
# yum install -y [nombre del paquete]
```

Para instalar las librerías base, se ingresan los siguientes comandos:

```
yum install -y kernel-devel
```

```
yum install -y kernel
```

```
yum install -y bison
```

```
yum install -y bison-devel
```

```
yum install -y ncurses
```

```
yum install -y ncurses-devel
```

```
yum install -y zlib
```

```
yum install -y zlib-devel
```

```
yum install -y openssl
```

```
yum install -y openssl-devel
```

```
yum install -y gnutls-devel
```

```
yum install -y gcc
```

```
yum install -y gcc-c++
```

3.3.2 INSTALACIÓN ASTERISK

Para la instalación total de Asterisk es necesario descargar, descomprimir e instalar varios paquetes individuales. Los paquetes asterisk, asterisk-addons, dahdi complete, libpri deben ser instalados de la siguiente manera en el directorio `/usr/src/` como establece el Estándar Base de Linux (LSB).

Las versiones instaladas, son aquellas que se utilizaron en el proyecto para la implementación del servidor Asterisk. Cabe señalar que Asterisk posee otras versiones y la elección de la versión depende de cada usuario y su aplicación. Se pueden obtener otras versiones de Asterisk accediendo a la página web de Asterisk. [7]

Se debe tomar en cuenta que antes de la instalación de una nueva versión de Asterisk, se debe revisar si no hay una versión anterior instalada que pueda perjudicar el funcionamiento de la nueva versión.

A continuación, se instalarán los paquetes. Para evitar errores de dependencia a la hora de correr los scripts de configuración, el orden de instalación debe ser el siguiente:

1. Libpri
2. Dadhi
3. Asterisk
4. Asterisk-addons

Al ejecutar la opción `menuselect` al instalar el paquete Asterisk, se abrirá un menú adicional, en el cual se debe seleccionar los paquetes para instalar diferentes servicios que Asterisk no instala por defecto. Entre ellos se debe escoger las opciones relacionadas a MySQL, que será el motor de base de datos para este proyecto. Una vez finalizado, se podrá realizar la conexión entre la base de datos y Asterisk.

Al finalizar la instalación completa de Asterisk, se crearán todos los archivos de configuración del sistema en la ruta `/etc/asterisk`, los cuales van a ser modificados posteriormente, para cumplir los objetivos planteados inicialmente en el proyecto.

3.4 CONFIGURACIÓN DE ARCHIVOS

En esta sección se encontrarán los archivos de configuración relevantes, utilizados para la elaboración de la aplicación que se planteó en los objetivos. Los archivos de configuración más importantes son `queues.conf` y `agents.conf`, pues son aquellas los que serán usados para redireccionar el tráfico hacia los agentes.

3.4.1 CONFIGURACIÓN SIP.CONF

El archivo de configuración `sip.conf` pertenece al protocolo SIP o Protocolo de Inicio de Sesiones, el cual es un protocolo que se encarga de iniciar, mantener o terminar las comunicaciones entre usuarios. El archivo de configuración SIP, incluye entre otras cosas información de los usuarios y las extensiones que van a interactuar con Asterisk y está ubicado en la ruta `/etc/asterisk/sip.conf`. [8]

Para el caso de la implementación de este proyecto el archivo `sip.conf`, tiene la configuración de los usuarios registrados con sus respectivos nombre de usuario, contraseñas y el contexto al que pertenecen. El código completo puede ser encontrado en el anexo A.

3.4.2 CONFIGURACIÓN EXTENSIONS.CONF

El archivo de configuración `extensions.conf` es uno de los archivos más importantes, no solo para este proyecto, sino en general para todas las aplicaciones de PBX de Asterisk. 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. En general se encarga de direccionar correctamente el tráfico de entrada y salida de Asterisk hacia las extensiones y viceversa. Este archivo está ubicado en la ruta `/etc/asterisk/extensions.conf`. [8]

Para este proyecto, en el archivo `extensions.conf` se configurarán las extensiones para que los usuarios puedan realizar llamadas, por otro lado se podrá realizar el inicio de sesión de los agentes, la dirección a la cola respectiva y la verificación de la prioridad del usuario basada en su identificación de llamada entrante.

Este archivo trabajará conjuntamente con varios códigos, escritos en PHP, que se detallarán posteriormente en este documento, con el fin de utilizar la herramienta que ofrece Asterisk para manejar opciones usando AGI. Estos códigos realizan tareas de validación, consulta con base de datos y redirección a las extensiones pertinentes para cumplir con los requerimientos.

En resumen, el archivo `extensions.conf` contiene todas las extensiones que van a ser utilizadas en el plan de marcado, tanto por los usuarios del sistema para llamar internamente, como para los clientes que llamen al centro de contacto para que se ejecuten las aplicaciones correspondientes y sean redirigidos al agente elegido según las normas estadísticas seleccionadas por el administrador del sistema en las políticas. La configuración completa del archivo `extensions.conf` puede ser encontrada en el anexo B.

3.4.3 CONFIGURACIÓN AGENTS.CONF

Este archivo de configuración es el encargado de modificar los parámetros de conexión de los agentes como el número de identificación, contraseña y nombre del agente. El archivo está ubicado en la ruta `/etc/asterisk/agents.conf`.

Los agentes son extensiones móviles, que permiten a los usuarios del sistema conectarse en cualquier terminal sin importar el número de la extensión en que se encuentren. Es de gran aplicación para los centros de contacto, pues los usuarios del sistema pueden usar cualquier teléfono y siempre le serán dirigidas sus correspondientes llamadas.

Los agentes pueden ser asignados a diferentes colas. Esto permite al agente hacerse cargo de diferentes tipos de llamadas entrantes, dependiendo de la arquitectura que sea configurada en el sistema. [9]

En este proyecto los agentes son de gran importancia, pues según los parámetros de tiempo de conexión de los agentes y de duración de sus llamadas se hacen las redirecciones, dependiendo de las políticas de redirección, hacia los agentes para la atención eficiente a los clientes. La configuración utilizada en esta aplicación para el archivo de configuración `agents.conf` estará disponible en el anexo C.

3.4.4 CONFIGURACIÓN QUEUES.CONF

Este archivo de configuración se encarga de todos los parámetros referentes a las colas y su funcionamiento dentro de Asterisk. Como su nombre lo dice, su función es encolar las llamadas para que puedan ser contestadas por los agentes que pertenezcan a la cola y que hayan completado su autenticación satisfactoriamente.

La característica principal de la cola es que todos los elementos que ingresen a ella, serán ubicados al final, y cuando sean requeridos seguirán el esquema FIFO (first in first out), es decir el primero que ingreso es el primero en salir; a menos que se utilicen varias colas con diferentes prioridades o pesos, con lo cual aquella cola que se encuentre con mayor prioridad será aquella a la cual se le atiendan los requerimientos antes que aquella cola que tenga menos prioridad. Esta opción de prioridad de las colas será utilizada en lo posterior para aplicar la funcionalidad de usuarios con prioridad basada en el identificador de llamada entrante o usuarios vip.

En una aplicación de centro de contacto, todas las llamadas son dirigidas a las colas, luego ocurre un proceso de redirección de las llamadas hacia los agentes que estén disponibles, dependiendo de las políticas que tenga la empresa para redirigir, como se puede ver en la figura 3.2.

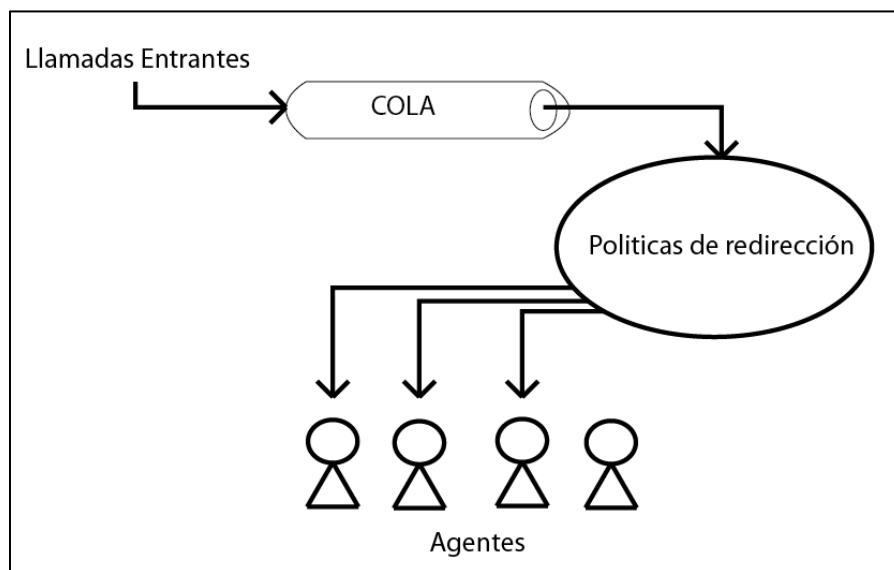


Figura 3.2. Esquema funcionamiento del sistema

Asterisk posee algunas estrategias para realizar esta redirección automáticamente, entre las cuales tenemos:

- ringall: hace sonar todos los teléfonos de los agentes disponibles hasta que uno conteste.
- roundrobin: distribuye las llamadas entre los agentes disponibles por igual.
- leastrecent: envía la llamada al agente menos reciente en atender llamadas.
- fewestcalls: envía la llamada al agente que ha contestado menos llamadas.

- random: envía llamadas a los agentes en orden aleatorio.

Estas estrategias pueden llegar a ser muy útiles, pero para la mayoría de las empresas se requiere un redireccionamiento mas específico, dependiendo de sus necesidades y políticas internas. [10]

En este proyecto se realizarán las estrategias de redireccionamiento de acuerdo a lo planteado inicialmente y a lo que seleccione el usuario como política de redirección en la página web del proyecto. El código correspondiente al archivo de configuración `queues.conf` puede ser revisado en el anexo D.

3.4.5 CONFIGURACIÓN CDR_MANAGER.CONF

Este archivo de configuración de Asterisk se lee cada vez que se inicia la sesión; dentro del mismo se configura para manipular el servidor Asterisk remotamente y revisar eventos que ocurren en la misma. Este archivo se encuentra en la ruta `/etc/asterisk/cdr_manager.conf`.

Dentro de este archivo se configuran el puerto, mediante el cual se tendrán acceso al servidor Asterisk, y las sesiones con sus respectivos permisos. Para revisión del código de este archivo de configuración revisar el anexo E.

3.4.6 CONFIGURACIÓN CDR_MYSQL.CONF

El archivo de configuración `cdr_mysql.conf` es el archivo que enlaza al servidor asterisk con el servidor de base de datos MYSQL que se encuentra en el mismo equipo mediante la línea de configuración `hostname=localhost`.

Este archivo se encuentra en la ruta `/etc/asterisk/cdr_mysql.conf`; dentro de la configuración de este también se especifican datos de conexión a la base de datos como se puede ver a continuación.

```
[global]
```

```
hostname=localhost
```

```
dbname=asterisk
```

```
password=lablecom10
```

```
user=root
```

Para que la conexión entre ambos servidores también necesitamos configurar otro archivo dentro de la carpeta asterisk/ que es el module.conf y agregar la siguiente línea en la configuración “load => cdr_addon_mysql.so”

3.4.7 PÁGINAS WEB INDEX.HTML, USERS.HTML

En un ambiente laboral, en el cual un sistema debe ser diseñado para funcionar de forma dinámica bajo las normas y políticas de una empresa, es necesaria una interfaz de selección que le permita al administrador del sistema poder cambiar las políticas del sistema sin alterar el código interno del mismo.

Es por esto que se decidió crear la interfaz de control de las políticas de redirección de la aplicación, mediante una página web. Esta página web estará alojada en el servidor web apache en la ruta /var/www/httpd/. La página web es sí, tendrá dos objetivos.

El primero, el de ser la interfaz mediante la cual, el administrador del sistema elige la política de redirección, las mismas que se especifican en los objetivos de este proyecto. Una vez seleccionada la política de redirección, la misma será almacenada en una base de datos, que será consultada antes de que ingrese alguna llamada, para determinar a que agente debe ser redirigida. El código de esta sección de la página web puede ser consultado en el anexo F.

La actualización de la política para la redirección, se realizará mediante un formulario en la pagina web, que dirigirá los datos del formulario al archivo alta2.php, el cual obtendrá los datos como variable y con los mismos, luego de realizar satisfactoriamente la conexión con la base de datos, actualizará la tabla correspondiente con los valores obtenidos del formulario HTML.

La sección más relevante del código es la obtención de datos de la página web y la elaboración de la consulta que actualizará la tabla con los datos obtenidos como se observa a continuación. El código completo del archivo alta2.php esta disponible en el anexo I.


```
$Opid=$_POST['Opid'];
```

```
$query = "UPDATE opc SET opid = ('".$Opid."');";
```

El segundo objetivo de la página web es, el de servir como formulario de ingreso de todos aquellos usuarios que se consideren como más importantes, para darles un estatus de vip. Con el ingreso de estos datos, se almacenará el nombre del usuario en la base de datos, para luego ser consultado cada vez que ingrese una llamada y poder determinar mediante el identificador de llamada entrante (nombre del usuario), si la llamada será encolada como vip o como usuario normal. El código de esta sección de la página web puede ser consultado en el anexo G.

La actualización en la base de datos se realizará de la misma manera como se realizó anteriormente, mediante un formulario en la página web y su posterior envío de los datos al archivo alta.php. La diferencia está en los datos que se están adquiriendo, lo que modifica la consulta para actualizar la base de datos, como se aprecia a continuación. El código completo del archivo alta.php se puede revisar en el anexo H.

```
$Nombre=$_POST['Nombre'];
```

```
$User=$_POST['User'];
```

```
$query = "INSERT INTO vip values('".$User."','".$Nombre."");"
```

3.4.8 ARCHIVOS ELECCIÓN.PHP, USUARIOVIP.PHP

Una vez iniciado el sistema, se debe decidir la política de elección que se va a seguir, para esto se utiliza la página web como se mencionó anteriormente. Es por esto, que ante cualquier llamada entrante es necesario consultar cual es la política vigente para realizar la elección correspondiente del agente, esto se realizará mediante la ejecución del archivo eleccion.php.

Básicamente, el archivo eleccion.php se encarga de recoger los datos de la tabla opc y dependiendo de la opción vigente, se dirige a la extensión correspondiente, como se muestra a continuación, para poder manejar la llamada de acuerdo a las políticas propuestas en este proyecto. El código completo del archivo eleccion.php puede ser revisado en el anexo J.

```
If ($tipo == "1" ) {  
  
    write ("EXEC GoTO 211,1");  
  
}  
  
If ($tipo == "2" ) {  
  
    write ("EXEC GoTO 212,1");  
  
}  
  
If ($tipo == "3" ) {  
  
    write ("EXEC GoTO 213,1");  
  
}  
  
If ($tipo == "4" ) {  
  
    write ("EXEC GoTO 214,1");  
  
}
```

Luego de la confirmación de que política va a ser utilizada, se deberá comprobar si la llamada entrante corresponde a un usuario vip o a un usuario normal. Esta confirmación la realizara el archivo usuariovip.php mediante la verificación de la existencia o no del usuario, en la tabla

vip, la misma que corresponde a todos los usuarios que han sido asignados como vip por el administrador del sistema. El código del archivo usuariovip.php se puede encontrar en el anexo K. Estos archivos mencionados están ubicados en la ruta /var/lib/asterisk/agi-bin/.

3.4.9 ARCHIVOS PHP

Una vez seleccionada la política que va a ser utilizada para la elección de los agentes, es necesario realizar varias búsquedas en las bases de datos para, dependiendo de la información encontrada en las mismas, poder realizar las operaciones de elección de agente de manera correcta. Estos archivos podrán ser encontrados en la ruta /var/lib/asterisk/agi-bin/.

En el momento en que se seleccione la política, se pondrá en marcha el archivo eleccion.php, éste a su vez dará como resultado una extensión del plan de marcado, que al ser seleccionada por Asterisk, llamara a uno de los 4 archivos que contienen las sentencias para la elección de agente, utilizando la interfaz AGI.

Los archivos antes mencionados son todos muy similares, debido a que todos deben de conectarse al motor de base de datos MySQL y a su vez, todos deben retornar la identificación de un agente, el cual será el único que podrá contestar la llamada entrante que se encuentra encolada. La única parte en la que difieren los archivos es en la consulta que va a realizarse, cambiando en el parámetro de búsqueda.

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 ejemplo de la tabla `log_queue` se puede ver en la figura 3.3.

```
mysql> select * from queue_log;
```

partition	time_id	call_id	queue	agent	verb	data1	data2	data3	data4	data5	serverid	unique_row_count
P02	1326290173	1326290165.3	NONE	Agent/1001	AGENTLOGIN	SIP/agent1-00000003						1
P02	1326290486	1326290485.4	MyQueue	NONE	ENTERQUEUE		agente2	1				4
P02	1326290486	1326290485.4	MyQueue	Agent/1001	CONNECT	0	1326290486.5	0				5
P02	1326290533	1326290485.4	MyQueue	Agent/1001	COMPLETEAGENT	0	47	1				1
P02	1326290533	1326290165.3	NONE	Agent/1001	AGENTLOGOFF	SIP/agent1-00000003	360					2
P02	1326290892	1326290885.7	NONE	Agent/1001	AGENTLOGIN	SIP/agent1-00000006						1
P02	1326290962	1326290961.10	MyQueue	NONE	ENTERQUEUE		agente2	1				1
P02	1326290962	1326290961.10	MyQueue	Agent/1001	CONNECT	0	1326290962.11	0				2
P02	1326291265	1326291264.16	MyQueue	NONE	ENTERQUEUE		jaurquiz	1				1
P02	1326291272	1326291271.17	QueueVip	NONE	ENTERQUEUE		dbanda	1				1
P02	1326291278	1326290961.10	MyQueue	Agent/1001	COMPLETEAGENT	0	316	1				1
P02	1326291324	1326291264.16	MyQueue	NONE	ABANDON	1	1	59				1
P02	1326291498	1326291271.17	QueueVip	NONE	ABANDON	1	1	226				1
P02	1326291703	1326291702.18	MyQueue	NONE	ENTERQUEUE		agente2	1				4
P02	1326291703	1326291702.18	MyQueue	Agent/1001	CONNECT	0	1326291703.19	0				5
P02	1326291751	1326291702.18	MyQueue	Agent/1001	COMPLETEAGENT	0	48	1				1
P02	1326291752	1326291730.20	MyQueue	NONE	ENTERQUEUE		jaurquiz	1				4
P02	1326291752	1326291730.20	MyQueue	Agent/1001	CONNECT	0	1326291752.22	0				6
P02	1326291752	1326291740.21	QueueVip	NONE	ENTERQUEUE		dbanda	1				9

Figura 3.3 Ejemplo de la información 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 el siguiente 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. El código de la consulta se puede ver en la parte superior de la figura 3.4. Para el archivo completo `mayortconectado.php` revisar el anexo L.

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 mas 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`, este código se puede revisar en la parte

inferior de la figura 3.4. Para el código completo del archivo `menortconectado.php` revisar el anexo M.

```
$query1 = "SELECT agent FROM (SELECT o.agent, o.time_id FROM
queue_log o WHERE o.verb = 'AGENTLOGIN' AND o.call_id NOT IN
(SELECT q.call_id FROM queue_log q WHERE q.verb = 'AGENTLOGOFF')
AND o.agent NOT IN (SELECT t.agent FROM queue_log t WHERE t.verb
= 'CONNECT' AND t.call_id NOT IN (SELECT q.call_id FROM queue_log
q WHERE q.verb = 'COMPLETECALLER' OR q.verb = 'COMPLETEAGENT'))
ORDER BY o.time_id) AS e;";
```

```
$query1 = "SELECT agent FROM (SELECT o.agent, o.time_id FROM
queue_log o WHERE o.verb = 'AGENTLOGIN' AND o.call_id NOT IN
(SELECT q.call_id FROM queue_log q WHERE q.verb = 'AGENTLOGOFF')
AND o.agent NOT IN (SELECT t.agent FROM queue_log t WHERE t.verb
= 'CONNECT' AND t.call_id NOT IN (SELECT q.call_id FROM queue_log
q WHERE q.verb = 'COMPLETECALLER' OR q.verb = 'COMPLETEAGENT'))
ORDER BY o.time_id DESC) AS e;";
```

Figura 3.4 Códigos de consulta `mayortconectado.php` y `menortconectado.php`


```

mysql> select * from cdr;
-----
| calldate | disposition | amaflags | accountcode | userfield | uniqueid |
-----
| 2011-11-09 09:40:03 | "diana" <davelez> | 3 | | davelez | 502 | Internos | SIP/davelez-0000000 | SIP/davelez-0000001 | Dial | SIP/davelez,10,m | 6 |
4 | ANSWERED | | | | | 1320849003.0 |
| 2011-11-09 09:48:22 | "diana" <davelez> | 3 | | davelez | login | Internos | SIP/davelez-0000002 | | Hangup | | 20 |
20 | ANSWERED | | | | | 1320850102.2 |
| 2011-11-09 10:10:50 | dbanda | 3 | | dbanda | 503 | Internos | SIP/dbanda-0000003 | SIP/Landino-0000004 | Dial | SIP/Landino,10,m | 1 |
0 | NO ANSWER | | | | | 1320851450.3 |
| 2011-11-09 11:25:58 | dbanda | 3 | | dbanda | 503 | Internos | SIP/dbanda-0000005 | SIP/Landino-0000006 | Hangup | | 11 |
0 | ANSWERED | | | | | 1320855958.5 |
| 2011-11-09 11:26:29 | dbanda | 3 | | dbanda | 501 | Internos | SIP/dbanda-0000007 | SIP/jaurquiz-0000008 | Dial | SIP/jaurquiz,10,m | 5 |
0 | NO ANSWER | | | | | 1320855989.7 |
| 2011-11-09 11:26:39 | dbanda | 3 | | dbanda | 1001 | Internos | SIP/dbanda-0000009 | SIP/1001-000000a | Dial | SIP/1001 | 20 |
0 | NO ANSWER | | | | | 1320855999.9 |
| 2011-11-09 11:27:11 | dbanda | 3 | | dbanda | 54 | Internos | SIP/dbanda-000000b | | Read | AID,agent-user | 10 |
10 | ANSWERED | | | | | 1320856031.11 |
| 2011-11-09 11:27:25 | dbanda | 3 | | dbanda | login | Internos | SIP/dbanda-000000c | | Hangup | | 16 |
16 | ANSWERED | | | | | 1320856045.12 |
| 2011-11-09 12:05:41 | "diana" <davelez> | 3 | | davelez | login | Internos | SIP/davelez-000000d | | Hangup | | 15 |
15 | ANSWERED | | | | | 1320858341.13 |
| 2011-11-09 12:06:13 | "Javier Urquiza" <jaurquiz> | 3 | | jaurquiz | 2020 | Internos | SIP/jaurquiz-000000e | | Hangup | | 2 |
2 | ANSWERED | | | | | 1320858373.14 |
| 2011-11-09 12:10:59 | "Javier Urquiza" <jaurquiz> | 3 | | jaurquiz | 1001 | Internos | SIP/jaurquiz-000000f | SIP/1001-00000010 | Dial | SIP/1001 | 4 |
0 | NO ANSWER | | | | | 1320858659.15 |
| 2011-11-09 12:11:06 | "Javier Urquiza" <jaurquiz> | 3 | | jaurquiz | login | Internos | SIP/jaurquiz-00000011 | | Hangup | | 33 |
33 | ANSWERED | | | | | 1320858665.17 |
| 2011-11-09 12:17:51 | "Javier Urquiza" <jaurquiz> | 3 | | jaurquiz | 503 | Internos | SIP/jaurquiz-00000012 | SIP/Landino-00000013 | Dial | SIP/Landino,10,m | 8 |
4 | ANSWERED | | | | | 1320859071.18 |
| 2011-11-09 12:18:58 | "Javier Urquiza" <jaurquiz> | 3 | | jaurquiz | login | Internos | SIP/jaurquiz-00000014 | | Hangup | | 9 |
9 | ANSWERED | | | | | 1320859138.20 |
| 2011-11-09 12:20:13 | dbanda | 3 | | dbanda | 2020 | Internos | SIP/dbanda-00000015 | | Hangup | | 2 |
2 | ANSWERED | | | | | 1320859213.21 |
-----
15 rows in set (0.00 sec)

```

Figura 3.5 Ejemplo tabla cdr

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 (revisar la figura 3.5), 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. En la figura 3.6 se detallan los códigos de consulta, en la sección superior el código de mayor promedio y en la sección inferior el código de menor promedio.

```
$query1 = "SELECT dstchannel FROM (SELECT dstchannel,
AVG(duration) dur FROM cdr WHERE
DAYOFYEAR(calldate)=DAYOFYEAR(CURDATE()) AND dst='2020' OR
dst='2021' GROUP BY dstchannel ORDER BY dur) AS t WHERE
dstchannel IN (SELECT agent FROM (SELECT o.agent, o.time_id FROM
queue_log o WHERE o.verb = 'AGENTLOGIN' AND o.call_id NOT IN
(SELECT q.call_id FROM queue_log q WHERE q.verb = 'AGENTLOGOFF')
AND o.agent NOT IN (SELECT t.agent FROM queue_log t WHERE t.verb
= 'CONNECT' AND t.call_id NOT IN (SELECT q.call_id FROM queue_log
q WHERE q.verb = 'COMPLETECALLER' OR q.verb = 'COMPLETEAGENT'))))
AS e);";
```

```
$query1 = "SELECT dstchannel FROM (select dstchannel,
AVG(duration) dur FROM cdr WHERE
DAYOFYEAR(calldate)=DAYOFYEAR(CURDATE()) AND dst='2020' OR
dst='2021' GROUP BY dstchannel ORDER BY dur DESC) AS t WHERE
dstchannel IN (SELECT agent FROM (SELECT o.agent, o.time_id FROM
queue_log o WHERE o.verb = 'AGENTLOGIN' AND o.call_id NOT IN
(SELECT q.call_id FROM queue_log q WHERE q.verb = 'AGENTLOGOFF')
AND o.agent NOT IN (SELECT t.agent FROM queue_log t WHERE t.verb
= 'CONNECT' AND t.call_id NOT IN (SELECT q.call_id FROM queue_log
q WHERE q.verb = 'COMPLETECALLER' OR q.verb = 'COMPLETEAGENT'))))
AS e);";
```

Figura 3.6. Códigos de consulta mayorpromedio.php, menorpromedio.php

Para los revisar los archivos completos de mayorpromedio.php y menorpromedio.php dirigirse respectivamente los anexos N y O.

CAPÍTULO 4

FUNCIONAMIENTO Y PRUEBAS

Una vez realizadas las instalaciones necesarias de los programas a utilizar tales como, el servidor de PBX Asterisk, el motor de Base de Datos MySQL, PHP5, para el manejo de AGI, el servidor web APACHE, para la inclusión de las paginas web, además de las configuraciones en

los archivos de Asterisk y la creación de código en php, se presentan las pruebas y el detalle de las mismas, para el correcto funcionamiento del centro de contacto.

4.1 INICIALIZANDO ASTERISK

A continuación se mostrará un conjunto de comandos que permiten iniciar o detener los servicios de Asterisk, todos estos comandos son ingresados mediante el intérprete de líneas de comando.

<code>service asterisk start</code>	Iniciar servicio
<code>service asterisk restart</code>	Reiniciar servicio
<code>service asterisk status</code>	Consultar estado del servicio
<code>asterisk -c</code>	Iniciar Asterisk y abrir la consola remota
<code>asterisk -r</code>	Ingresar a la consola remota
<code>asterisk -rx'comando'</code>	Ejecutar comando sin ingresar a la consola

4.2 CONFIGURACIÓN DE LA BASE DE DATOS

Debido a la necesidad de manejar gran cantidad de datos para la realización de este proyecto, se decidió utilizar un motor de base de datos como MySQL, para el manejo de tablas y datos concernientes al proyecto.

Se creó una base de datos de nombre Asterisk de la cual se obtiene información que es consultada por la PBX, para definir los criterios de elección de los agentes conectados y tipos de usuario. Dicha base de datos posee 4 tablas que se describirán a continuación.

4.2.1 TABLA CDR

Esta tabla proporciona toda la información de las llamadas realizadas tales como la fecha, el origen, canal y destino de la llamada, así como también la duración de las mismas. La descripción de esta tabla se puede apreciar en la figura 4.1.

Esta tabla 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.

```
mysql> describe cdr;
```

Field	Type	Null	Key	Default	Extra
calldate	datetime	NO	MUL	0000-00-00 00:00:00	
clid	varchar(80)	NO			
src	varchar(80)	NO			
dst	varchar(80)	NO	MUL		
dcontext	varchar(80)	NO			
channel	varchar(80)	NO			
dstchannel	varchar(80)	NO			
lastapp	varchar(80)	NO			
lastdata	varchar(80)	NO			
duration	int(11)	NO		0	
billsec	int(11)	NO		0	
disposition	varchar(45)	NO			
amaflags	int(11)	NO		0	
accountcode	varchar(20)	NO	MUL		
userfield	varchar(255)	NO			
uniqueid	varchar(32)	NO			

16 rows in set (0.00 sec)

Figura 4.1 Descripción de la tabla cdr

4.2.2 TABLA QUEUE_LOG

Esta tabla 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. [11] La tabla queue_log se describe en la figura 4.2.

```
mysql> describe queue_log;
```

Field	Type	Null	Key	Default	Extra
partition	varchar(20)	NO	MUL		
time_id	int(11) unsigned	NO		0	
call_id	varchar(30)	NO			
queue	varchar(30)	NO			
agent	varchar(30)	NO			
verb	varchar(30)	NO			
data1	varchar(30)	NO			
data2	varchar(30)	NO			
data3	varchar(30)	NO			
data4	varchar(30)	NO			
data5	varchar(30)	NO			
serverid	varchar(10)	NO			
unique_row_count	int(10) unsigned	NO		NULL	auto_increment

13 rows in set (0.00 sec)

Figura 4.2 Tabla queue_log

4.2.3 TABLA VIP

La tabla vip, figura 4.3, almacena todos los usuarios que han sido registrados por el administrador de sistema y reciben un trato especial, que consiste en ser encolados con una prioridad mayor que los usuarios que no pertenecen a esta tabla. Esto permitirá a los usuarios vip ser atendidos con mayor prontitud. Estos usuarios son registrados en a través de la pagina web como se puede observar en la figura 4.4.

```
mysql> describe vip;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| callId | varchar(80)   | NO   | PRI |          |       |
| Nombre | varchar(80)   | NO   |     |          |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

Figura 4.3 Tabla vip

The screenshot shows a web browser window with the address bar displaying the file path: file:///C:/Users/Andinos/Desktop/luis/final%20de%20tesis/html/users.html. The page content includes a navigation menu with links for 'Políticas' and 'Usuarios Vip'. Below the menu, there is a login form with two input fields: 'Nombre y Apellidos' and 'User'. An 'Enviar' button is positioned below the 'User' field.

Figura 4.4 Pagina web para el ingreso de Usuarios vip

4.2.4 TABLA OPC

La tabla opc es aquella que contiene un único dato, que el sistema consulta para saber la política de elección de agente, como se puede apreciar en la figura 4.5. Ésta política será la que el administrador del sistema haya ingresado mediante la página web como se muestra en la figura 4.6.


```
mysql> describe opc;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| opid  | varchar(1)    | NO   | PRI |          |       |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Figura 4.5 Tabla opc

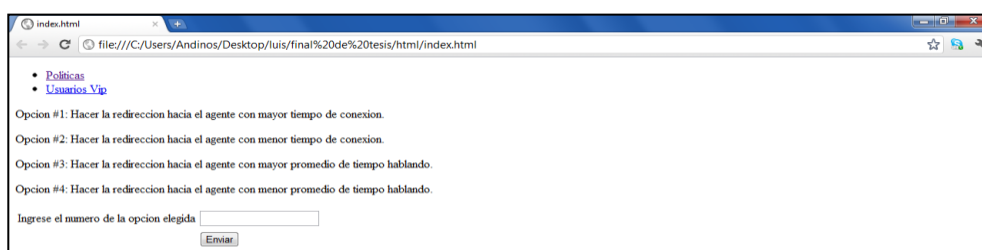


Figura 4.6 Página web para el ingreso de políticas de decisión

4.3 PRUEBA DE LLAMADA AL CALL CENTER

En la figura 4.7 se muestra todo el proceso, desde que el cliente realiza la llamada hasta que contesta un agente que ha sido seleccionado, pasando por la consulta de la política elegida, verificación del tipo de usuario es por medio de la identificación de llamada entrante, inhabilitación de todos los agentes, luego habilitación del agente que ha sido seleccionado por las políticas correspondientes.

```

== Using SIP RTP CoS mark 5
-- Executing [2020@internos:1] Answer("SIP/dbanda-00000051", "") in new stack
-- Executing [2020@internos:2] PauseQueueMember("SIP/dbanda-00000051", ".Agent/1001") in new stack
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
-- Executing [2020@internos:3] PauseQueueMember("SIP/dbanda-00000051", ".Agent/1002") in new stack
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
-- Executing [2020@internos:4] PauseQueueMember("SIP/dbanda-00000051", ".Agent/1003") in new stack
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
-- Executing [2020@internos:5] AGI("SIP/dbanda-00000051", "eleccion.php") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/eleccion.php
-- AGI Script Executing Application: (GoT0) Options: (212,1)
-- Goto (Internos,212,1)
-- <SIP/dbanda-00000051>-AGI Script eleccion.php completed, returning 0
-- Executing [212@internos:1] Answer("SIP/dbanda-00000051", "") in new stack
-- Executing [212@internos:2] AGI("SIP/dbanda-00000051", "menostconectado.php") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/menostconectado.php
-- AGI Script Executing Application: (GoT0) Options: (202,1)
-- Goto (Internos,202,1)
-- <SIP/dbanda-00000051>-AGI Script menostconectado.php completed, returning 0
-- Executing [202@internos:1] Answer("SIP/dbanda-00000051", "") in new stack
-- Executing [202@internos:2] UnpauseQueueMember("SIP/dbanda-00000051", ".Agent/1002") in new stack
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
-- Executing [202@internos:3] AGI("SIP/dbanda-00000051", "usuariovip.php") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/usuariovip.php
-- AGI Script Executing Application: (GoT0) Options: (2021,1)
-- Goto (Internos,2021,1)
-- <SIP/dbanda-00000051>-AGI Script usuariovip.php completed, returning 0
-- Executing [2021@internos:1] GetMusicOnHold("SIP/dbanda-00000051", "default") in new stack
-- Executing [2021@internos:2] Queue("SIP/dbanda-00000051", "QueueVip,r") in new stack
Jan 11 11:52:25 WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
-- Stopped music on hold on SIP/agent2-0000003f
-- agent_call, call to agent '1002' call on 'SIP/agent2-0000003f'
-- <SIP/agent2-0000003f> Playing 'beep.slin' (language 'es')
-- Called Agent/1002
-- Agent/1002 answered SIP/dbanda-00000051

```

Figura 4.7 Proceso de llamada al centro de contacto

La figura 4.8, muestra que cuando se inicia la llamada todos los agentes que se encuentran en la cola se colocan en un estado inactivo o inhabilitado, esto lo hacemos para después activar al agente dependiendo del tipo de política que haya escogido el administrador del sistema. Esta política es consultada mediante AGI y ejecuta un archivo con código PHP llamado elección.php (línea señalada entre flechas azules, figura 4.8) el cual consulta la tabla opc y, dependiendo de los datos en la tabla envía a otra extensión en el archivo extensions.conf (en el caso del ejemplo a la extensión 212).

```

== Using SIP RTP CoS mark 5
-- Executing [2020@internos:1] Answer("SIP/dbanda-00000051", "") in new stack
-- Executing [2020@internos:2] PauseQueueMember("SIP/dbanda-00000051", "Agent/1001") in new stack
[Jan 11 11:52:25] WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
[Jan 11 11:52:25] WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
[Jan 11 11:52:25] WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
-- Executing [2020@internos:3] PauseQueueMember("SIP/dbanda-00000051", "Agent/1002") in new stack
[Jan 11 11:52:25] WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
[Jan 11 11:52:25] WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
[Jan 11 11:52:25] WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
-- Executing [2020@internos:4] PauseQueueMember("SIP/dbanda-00000051", "Agent/1003") in new stack
[Jan 11 11:52:25] WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
[Jan 11 11:52:25] WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
[Jan 11 11:52:25] WARNING[22088]: : : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
-- Executing [2020@internos:5] AGI("SIP/dbanda-00000051", "eleccion.php") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/eleccion.php
-- AGI Script Executing Application: (GoTo) Options: (212,1)
-- Goto (internos,212,1)
-- <SIP/dbanda-00000051>AGI Script eleccion.php completed, returning 0

```

Figura 4.8 Inicio de la llamada

Siguiendo el ejemplo, la figura 4.9 muestra que dentro de la extensión 212 se ejecuta el siguiente AGI que es `menostconectado.php` (línea entre flechas azules, figura 4.9) que devuelve al agente disponible que menos tiempo tiene conectado, en este caso es el “Agent/1002” al cual se lo activa para que pueda recibir la llamada como se muestra en la línea entre flechas rojas.

```

-- Executing [212@internos:1] Answer("SIP/dbanda-00000051", "") in new stack
-- Executing [212@internos:2] AGI("SIP/dbanda-00000051", "menostconectado.php") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/menostconectado.php
-- AGI Script Executing Application: (GoTo) Options: (202,1)
-- Goto (internos,202,1)
-- <SIP/dbanda-00000051>AGI Script menostconectado.php completed, returning 0
-- Executing [202@internos:1] Answer("SIP/dbanda-00000051", "") in new stack
-- Executing [202@internos:2] UnpauseQueueMember("SIP/dbanda-00000051", "Agent/1002") in new stack

```

Figura 4.9 Elección de AGI

En la última parte de la llamada del centro de contacto consulta la base de datos (línea entre flechas azules) específicamente la tabla `vip`, para

consultar si la persona que realiza la llamada es un usuario vip. En el caso que el usuario esté en la tabla vip, el programa dirige la llamada hacia otra extensión donde se coloca al usuario en una cola con prioridad (QueueVip) lo que hará que las llamadas que entren en esa cola sea contestadas antes que la cola de los usuarios normales (MyQueue).

```

Executing [202@internos:3] AGI("SIP/dbanda-00000051", "usuariovip.php") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/usuariovip.php
-- AGI Script Executing Application: (GoT0) Options: (2021,1)
-- Goto (internos,2021,1)
-- <SIP/dbanda-00000051>AGI Script usuariovip.php completed, returning 0
-- Executing [2021@internos:1] SetMusicOnHold("SIP/dbanda-00000051", "default") in new stack
-- Executing [2021@internos:2] Queue("SIP/dbanda-00000051", "QueueVip,") in new stack
[Jan 11 11:52:25] WARNING[22000]: : Realtime mapping for 'queue_log' found to engine 'mysql', but the engine is not available
-- Stopped music on hold on SIP/agente2-0000003f
-- agent call, call to agent '1002' call on 'SIP/agente2-0000003f'
-- <SIP/agente2-0000003f> Playing 'beep.slin' (Language 'es')
-- Called Agent/1002
-- Agent/1002 answered SIP/dbanda-00000051

```

Figura 4.10 Tipo de usuario

4.4 ESPECIFICACIONES DE AGI

En la tabla IV están especificaciones de las opciones que puede tener la tabla opc, dependiendo del dato de opc se dirige a una extensión que contiene el AGI correspondiente a la opción especificada.

Opción (opc)	Extensiones	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	Este php devuelve el agente con mayor promedio de tiempo en llamadas
4	214	menorpromedio.php	Muestra el agente con menor promedio en sus llamadas

Tabla IV. Tabla de elección de agentes

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Se cumplió el objetivo principal de implementar el centro de llamadas utilizando programas complementarios al software para PBX Asterisk como: motores de base de datos, servidores web y un código propiedad de QUEUOMETRICS, para obtener información de las colas en tiempo real. Estos datos fueron necesarios para la asignación de los agentes para las llamadas recibidas y el filtrado de llamadas entrantes para clientes con privilegios por medio de AGI en complemento con códigos en PHP.

2. La implementación de este proyecto, cubre una necesidad importante en el mundo ya que puede ser utilizado en varios sistemas de consulta de información y atención al cliente y acoplarse a sus políticas internas. Según las pruebas realizadas durante la realización del proyecto, la información recopilada en las bases de datos sobre el detalle de las llamadas, abarca un campo de parámetros muy diverso. Teniendo ésta cantidad de parámetros, es posible implementar nuevas políticas para redirección de las llamadas, cambiando únicamente el código de consulta

de la política, lo que permitiría cumplir con necesidades específicas de diferentes empresas.

3. La transición de la red pública a la red sobre IP en general, se realizará en poco tiempo debido a que el uso de una PC emulando a un teléfono por medio de un software en una red TCP/IP es de fácil aprendizaje, en el mundo de hoy. En este trabajo se implementó un sistema de amigable para los agentes, gracias a que la interacción con el software le indica verbalmente los pasos a seguir para iniciar sesión, advertencia ante errores, etc. Incluso el sistema para el administrador es de fácil uso al momento de elegir la política correspondiente a la empresa.

4. Las tablas en la base de datos llenadas por medio de Asterisk, da los resultados de las llamadas e información de los agentes. Este trabajo mediante líneas de consultas en la base de Datos y empleando AGI de Asterisk para controlar los resultados, se logró consultar la información necesaria para las diferentes políticas que la empresa desea y aplicarlas correctamente. La eficiencia de la atención de la empresa será basada en la política que se elija. Dichas políticas de redirección podrán ser cambiadas por medio de una página web, que la misma empresa manejará a su conveniencia.

RECOMENDACIONES

1. Usar programas que emulen un teléfono IP (SoftPhone), lo que permitiría ahorrar en costos de adquisición y mantener un buen ambiente en el centro de llamada y no perturbar o molestar las demás llamadas con el timbrar de los teléfonos, mediante el uso de diademas.
2. Usar aplicaciones de código abierto como QUEUETRICS, que ayudan al manejo de la base de datos con información de Asterisk y que tienen interfaces amigables y de fácil manejo. Además el uso de aplicaciones de código abierto abaratan costos de implementación.
3. Reducir el estrés producido durante el tiempo de espera del cliente, reproduciendo música menos monótona y anunciando que sus inquietudes serán atendidas en brevedad.
4. Revisar el uso de las herramientas para el manejo de la información en tiempo real con bases de datos, que posee Asterisk. Estas herramientas dieron problemas en la implementación por lo que se optó por código adicional como el qloader, para cumplir el mismo objetivo.

5. Se recomienda el uso de la interfaz AML de Asterisk, como una solución alternativa para el manejo de estados de agentes (Activo o Inactivo), colas y asignación de llamadas.

ANEXO A

En el anexo A encontramos el código correspondiente al archivo sip.conf.

```
[general]
context=noautenticadas
allowguest=no
srvlookup=yes
udpbindaddr=0.0.0.0
tcpenable=no
qualify=yes
language=es
callcounter=yes
```

```
[jaurquiz]
type=friend
context=internos
host=dynamic
nat=yes
secret=jaug60ie
dtmfmode=auto
disallow=all
allow=ulaw
```

```
[dbanda]
type=friend
context=internos
host=dynamic
nat=yes
secret=dgbb46ie
dtmfmode=auto
disallow=all
allow=ulaw
```

```
[davelez]
type=friend
context=internos
host=dynamic
nat=yes
secret=secretito17
dtmfmode=auto
disallow=all
allow=ulaw
```

```
[aurquizo]
type=friend
context=internos
host=dynamic
nat=yes
secret=20071124
dtmfmode=auto
```

disallow=all
allow=ulaw

[landino]
type=friend
context=internos
host=dynamic
nat=yes
secret=luis
dtmfmode=auto
disallow=all
allow=ulaw

[bevelez]
type=friend
context=internos
host=dynamic
nat=yes
secret=pepito
dtmfmode=auto
disallow=all
allow=ulaw

[pepito]
type=friend
context=internos
host=dynamic
nat=yes
secret=pepito
dtmfmode=auto
disallow=all
allow=ulaw

[agente1]
type=friend
context=internos
host=dynamic
nat=yes
secret=agente
dtmfmode=auto
disallow=all
allow=ulaw

[agente2]
type=friend
context=internos
host=dynamic
nat=yes
secret=agente
dtmfmode=auto
disallow=all
allow=ulaw

[agente3]

```
type=friend
context=internos
host=dynamic
nat=yes
secret=agente
dtmfmode=auto
disallow=all
allow=ulaw
```

ANEXO B

En el anexo B encontramos el código correspondiente al archivo
extensions.conf

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

[globals]
AGENT_CONTEXT=agents
USER1=SIP/jaurquiz
USER2=SIP/davelez
USER3=SIP/landino
USER4=SIP/dbanda
USER5=SIP/aurquizo
USER6=SIP/pepito

[internos]
exten => 1001,1,AgentLogin()

exten => 2020,1,Answer()
exten => 2020,2,PauseQueueMember(,Agent/1001)
exten => 2020,3,PauseQueueMember(,Agent/1002)
exten => 2020,4,PauseQueueMember(,Agent/1003)
exten => 2020,5,AGI(eleccion.php)
exten => 2020,n,Hangup()

exten => 2345,1,Answer()
exten => 2345,n,UnpauseQueueMember(,Agent/1001)
exten => 2345,n,UnpauseQueueMember(,Agent/1002)
exten => 2345,n,UnpauseQueueMember(,Agent/1003)
exten => 2345,n,AGI(usuariovip.php)

exten => 2021,1,SetMusicOnHold(default)
exten => 2021,n,Queue(QueueVip,r)
exten => 2021,n,Hangup()

exten => 2022,1,SetMusicOnHold(default)
exten => 2022,n,Queue(MyQueue,r)
exten => 2022,n,Hangup()

exten => 222,1,answer()
exten => 222,n,PauseQueueMember(,Agent/1001)
exten => 222,n,Hangup()

exten => 501,1,Macro(extensiones,${USER1})

exten => 502,1,Macro(extensiones,${USER2})
```

```

exten => 503,1,Macro(extensiones,${USER3})

exten => 504,1,Macro(extensiones,${USER4})

exten => 505,1,Macro(extensiones,${USER5})

exten => 506,1,Macro(extensiones,${USER6})

exten => 200,1,answer()
exten => 200,n,UnpauseQueueMember(,Agent/1001)
exten => 200,n,UnpauseQueueMember(,Agent/1002)
exten => 200,n,UnpauseQueueMember(,Agent/1003)

exten => 201,1,answer()
exten => 201,n,UnpauseQueueMember(,Agent/1001)
exten => 201,n,AGI(usuariovip.php)
exten => 201,n,wait(5)
exten => 201,n,Hangup()

exten => 202,1,answer()
exten => 202,n,UnpauseQueueMember(,Agent/1002)
exten => 202,n,AGI(usuariovip.php)
exten => 202,n,Hangup()

exten => 203,1,answer()
exten => 203,n,UnpauseQueueMember(,Agent/1003)
exten => 203,n,AGI(usuariovip.php)
exten => 203,n,Hangup()

exten => 211,1,answer()
exten => 211,n,AGI(mayortconectado.php)
exten => 211,n,Hangup()

exten => 212,1,answer()
exten => 212,n,AGI(menortconectado.php)
exten => 212,n,Hangup()

exten => 213,1,answer()
exten => 213,n,AGI(mayorpromedio.php)
exten => 213,n,Hangup()

exten => 214,1,answer()
exten => 214,n,AGI(menorpromedio.php)
exten => 214,n,Hangup()

[macro-extensiones]
exten => s,1,Dial(${ARG1},10,m)
same => n,GotoIf("${DIALSTATUS}" = "BUSY"?busy:unavail)
same => n(unavail),VoiceMail(${MACRO_EXTEN}@default,u)
same => n,Hangup()
same => n(busy),VoiceMail(${MACRO_EXTEN}@default,b)
same => n,Hangup()

```

ANEXO C

En el anexo C encontramos el código correspondiente al archivo `agents.conf`.

```
[general]
multiplelogin=yes

[agents]
maxlogintries=3
autologoff=15
autologoffunavail=no
ackcall=yes
acceptdtmf=#
endcall=yes
enddtmf=*
musiconhold => music_class
goodbye => goodbye_file
updatecdr=no
recordagentcalls=yes
savecallsin=/var/calls
agent => 1001,12345,Javier Urquizo
agent => 1002,54321,Luis Andino
agent => 1003,67890,Diana Velez
```

ANEXO D

En el anexo D encontramos el código correspondiente al archivo `queues.conf`.

```
[general]
autofill=yes
updatecdr=yes
```

```
[sales]
strategy=rrmemory
ringinuse=no
joinempty=no
strategy=ringall
```

```
[MyQueue] ; nombre de la cola
music=default ; música que coloca antes que lo atiendan
strategy=ringall ; estrategia de ring
eventwhencalled=yes
timeout=15 ; tiempo máximo de ring a un miembro
retry=5 ; tiempo de espera hasta reintentar con otro miembro
wrapuptime=0 ; tiempo de espera luego de intentar con todos los
miembros
maxlen = 0 ;
announce-frequency = 0 ; cantidad de veces que sale el anuncio
announce-holdtime = no ; si dice el tpo de espera aproximado
```

```
member => Agent/1001,1
member => Agent/1002,1
member => Agent/1003,1
```


ANEXO E

En el anexo E encontramos el código correspondiente al archivo `manager.conf`.

```
[general]
enabled = yes
port = 5038
bindaddr = 0.0.0.0

[mdm]
secret = pass
read = call,log,verbose,agent,dialplan
write = system,call,log,verbose,command,agent,user,originate
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.255
```

ANEXO F

En el anexo F encontramos el código correspondiente al archivo index.html.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
  </head>
  <body>
    <ul>
      <li><a href="index.html">Políticas</a></li>
      <li><a href="users.html">Usuarios Vip</a></li>
    </ul>
    <p>Opcion #1: Hacer la redireccion hacia el agente con mayor
tiempo de conexion.</p>
    <p>Opcion #2: Hacer la redireccion hacia el agente con menor
tiempo de conexion.</p>
    <p>Opcion #3: Hacer la redireccion hacia el agente con mayor
promedio de tiempo hablando.</p>
    <p>Opcion #4: Hacer la redireccion hacia el agente con menor
promedio de tiempo hablando.</p>
    <div>
      <form method="post" action="alta2.php">
        <table>
          <tr>
            <td>Ingrese el numero de la opcion elegida</td>
            <td><input name='Opid' type='text' maxlength='1'
/></td>
          </tr>
          <tr>
            <td>&nbsp;</td>
            <td><input name='submit' type='submit'
value='Enviar' /></td>
          </tr>
        </table>
      </div>
    </body>
  </html>
```

ANEXO G

En el anexo G encontramos el código correspondiente al archivo users.html.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
  </head>
  <body>
    <ul>
      <li><a href="index.html">Políticas</a></li>
      <li><a href="users.html">Usuarios Vip</a></li>
    </ul>
    <form method="post" name="frm" action="alta.php">
      <table>
        <tr>
          <td>Nombre y Apellidos</td>
          <td><input name='Nombre' type='text'
maxlength='20' /></td>
        </tr>
        <tr>
          <td>User</td>
          <td><input name='User' type='text'
maxlength='20' /></td>
        </tr>
        <tr>
          <td>&nbsp;</td>
          <td><input name='submit' type='submit'
value='Enviar' /></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

ANEXO H

En el anexo H encontramos el código correspondiente al archivo alta.php.

```
<?php
ob_implicit_flush(true);
set_time_limit(6);
function connect_db()
{
$db_connection = mysql_connect ('localhost', 'root',
'labtelecom10') or die (mysql_error());
$db_select = mysql_select_db('asterisk') or die (mysql_error());
}
connect_db();

$Nombre=$_POST['Nombre'];
$user=$_POST['User'];

$query = "INSERT INTO vip values('".$user."','".$Nombre."')";

if (!mysql_query($query)) {

    echo "Se produjo un error al ingresar los datos, "
.mysql_error();
}
else{
echo "Los datos han sido ingresaron de forma satisfactoria";
}
?>
```

ANEXO I

En el anexo I encontramos el código correspondiente al archivo alta2.php.

```
<?php
ob_implicit_flush(true);
set_time_limit(6);
function connect_db()
{
$db_connection = mysql_connect ('localhost', 'root',
'labtelecom10') or die (mysql_error());
$db_select = mysql_select_db('asterisk') or die (mysql_error());
}
connect_db();

$Opid=$_POST['Opid'];
$query = "UPDATE opc SET opid = ('".$Opid."');";

if (!mysql_query($query)) {

    echo "Se produjo un error al ingresar los datos, "
.mysql_error();
}
else{
echo "Los datos han sido ingresaron de forma satisfactoria";
}
?>
```

ANEXO J

En el anexo J encontramos el código correspondiente al archivo `eleccion.php`.

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

<?php
ob_implicit_flush(true);
set_time_limit(6);
$in = fopen("php://stdin","r");
$stdlog = fopen("/var/log/asterisk/acadi.log", "w");

function read()
{
global $in, $debug;
$input = str_replace("\n", "", fgets($in, 4096));
return $input;
}
function errlog($line) {
global $err;
echo "VERBOSE \"$line\"\n";
}
function write($line) {
global $debug;
echo $line."\n";
}
function connect_db()
{
$db_connection = mysql_connect ('localhost', 'root',
'labtelecom10') or die (mysql_error());
$db_select = mysql_select_db('asterisk') or die (mysql_error());
}
while ($env=read())
{
$env = str_replace("\"", "", $env);
$s = split(":", $env);
$agi[str_replace("agi_", "", $s[0])] = trim($s[1]);
if ($env == "")
{
break;
}
}

// programa inicial
$cli = $agi[callerid];
$exten= $agi[extension];

connect_db();
```

```
$query1 = "Select * from opc";

$query_result1 = @mysql_query($query1);
$row_count = mysql_num_rows($query_result1);
$row1 = @mysql_fetch_array ($query_result1);

If ($row_count != 0 ) { //registro ya existe
    $tipo=$row1["opid"];
    write ("SET CONTEXT internos");

    If ($tipo == "1" ) {
        write ("EXEC GoTO 211,1");
    }
    If ($tipo == "2" ) {
        write ("EXEC GoTO 212,1");
    }
    If ($tipo == "3" ) {
        write ("EXEC GoTO 213,1");
    }
    If ($tipo == "4" ) {
        write ("EXEC GoTO 214,1");
    }
}
Else {
    write ("SET CONTEXT internos");
    write ("EXEC GoTO 1001,1"); //extension para clientes nuevos
}

fclose($in);
fclose($stdlog);
exit;
?>
```

ANEXO K

En el anexo K encontramos el código correspondiente al archivo `usuariovip.php`.

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

<?php
ob_implicit_flush(true);
set_time_limit(6);
$in = fopen("php://stdin","r");

$stdlog = fopen("/var/log/asterisk/acadi.log", "w");

function read()
{
    global $in, $debug;
    $input = str_replace("\n", "", fgets($in, 4096));
    return $input;
}
function errlog($line) {
    global $err;
    echo "VERBOSE \"$line\"\n";
}
function write($line) {
    global $debug;
    echo $line."\n";
}
function connect_db()
{
    $db_connection = mysql_connect ('localhost', 'root',
    'labtelecom10') or die (mysql_error());
    $db_select = mysql_select_db('asterisk') or die (mysql_error());
}
while ($env=read())
{
    $env = str_replace("\",\"",", $env);
    $s = split(":", $env);
    $agi[str_replace("agi_", "", $s[0])] = trim($s[1]);
    if ($env == "")
    {
        break;
    }
}

// programa inicial

$cli = $agi[callerid];
echo $cli;
```



```
connect_db();

$query1 = "select callId from vip where callId='".$cli.'"";

$query_result1 = @mysql_query($query1);
$row_count = mysql_num_rows($query_result1);
$row1 = @mysql_fetch_array ($query_result1);

write ("SET CONTEXT internos");

If ($row_count > 0 ) { //registro ya existe
    write ("EXEC GoTO 2021,1");
}
Else { // registro no existe en nuestra base, entonces es un
cliente nuevo
    write ("EXEC GoTO 2022,1");
}

fclose($in);
fclose($stdlog);
exit;
?>
```

ANEXO L

En el anexo L encontramos el código correspondiente al archivo `mayortconectado.php`.

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

<?php
ob_implicit_flush(true);
set_time_limit(6);
$in = fopen("php://stdin","r");

$stdlog = fopen("/var/log/asterisk/acadi.log", "w");

function read()
{
    global $in, $debug;
    $input = str_replace("\n", "", fgets($in, 4096));
    return $input;
}
function errlog($line) {
    global $err;
    echo "VERBOSE \"$line\"\n";
}
function write($line) {
    global $debug;
    echo $line."\n";
}
function connect_db()
{
    $db_connection = mysql_connect ('localhost', 'root',
    'labtelecom10') or die (mysql_error());
    $db_select = mysql_select_db('asterisk') or die (mysql_error());
}
while ($env=read())
{
    $env = str_replace("\",\"",\"", $env);
    $s = split(":", $env);
    $agi[str_replace("agi_", "", $s[0])] = trim($s[1]);
    if ($env == "")
    {
        break;
    }
}

// programa inicial

$cli = $agi[callerid];
$exten= $agi[extension];
```

```

connect_db();

$query1 = "Select agent from (select o.agent, o.time_id from
queue_log o where o.verb = 'AGENTLOGIN' and o.call_id not in
(select q.call_id from queue_log q where q.verb = 'AGENTLOGOFF')
and o.agent not in (select t.agent from queue_log t where t.verb
= 'CONNECT' and t.call_id not in (select q.call_id from queue_log
q where q.verb = 'COMPLETECALLER' or q.verb = 'COMPLETEAGENT'))
ORDER BY o.time_id) AS e;";
$query_result1 = @mysql_query($query1);
$row_count = mysql_num_rows($query_result1);
$row1 = @mysql_fetch_array ($query_result1);

If ($row_count != 0 ) { //registro ya existe
    $tipo=$row1["agent"];
    write ("SET CONTEXT internos");
    If ($tipo == "Agent/1001" ) {
        write ("EXEC GoTO 201,1");
    }
    If ($tipo == "Agent/1002" ) {
        write ("EXEC GoTO 202,1");
    }
    If ($tipo == "Agent/1003" ) {
        write ("EXEC GoTO 203,1");
    }
}
Else { // registro no existe en nuestra base, entonces es un
cliente nuevo
    write ("SET CONTEXT internos");
    write ("EXEC GoTO 2345,1"); //extension para clientes nuevos
}

fclose($in);
fclose($stdlog);
exit;
?>

```

ANEXO M

En el anexo M encontramos el código correspondiente al archivo `menortconectado.php`.

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

<?php
ob_implicit_flush(true);
set_time_limit(6);
$in = fopen("php://stdin","r");

$stdlog = fopen("/var/log/asterisk/acadi.log", "w");

function read()
{
    global $in, $debug;
    $input = str_replace("\n", "", fgets($in, 4096));
    return $input;
}
function errlog($line) {
    global $err;
    echo "VERBOSE \"$line\"\n";
}
function write($line) {
    global $debug;
    echo $line."\n";
}
function connect_db()
{
    $db_connection = mysql_connect ('localhost', 'root',
    'labtelecom10') or die (mysql_error());
    $db_select = mysql_select_db('asterisk') or die (mysql_error());
}
while ($env=read())
{
    $env = str_replace("\",\"", $env);
    $s = split(":", $env);
    $agi[str_replace("agi_", "", $s[0])] = trim($s[1]);
    if ($env == "")
    {
        break;
    }
}

// programa inicial

$cli = $agi[callerid];
$exten= $agi[extension];
```

```

connect_db();

$query1 = "Select agent from (select o.agent, o.time_id from
queue_log o where o.verb = 'AGENTLOGIN' and o.call_id not in
(select q.call_id from queue_log q where q.verb = 'AGENTLOGOFF')
and o.agent not in (select t.agent from queue_log t where t.verb
= 'CONNECT' and t.call_id not in (select q.call_id from queue_log
q where q.verb = 'COMPLETECALLER' or q.verb = 'COMPLETEAGENT'))
ORDER BY o.time_id DESC) AS e;";
$query_result1 = @mysql_query($query1);
$row_count = mysql_num_rows($query_result1);
$row1 = @mysql_fetch_array ($query_result1);

If ($row_count != 0 ) { //registro ya existe
    $tipo=$row1["agent"];
    write ("SET CONTEXT internos");
    If ($tipo == "Agent/1001" ) {
        write ("EXEC GoTO 201,1");
    }
    If ($tipo == "Agent/1002" ) {
        write ("EXEC GoTO 202,1");
    }
    If ($tipo == "Agent/1003" ) {
        write ("EXEC GoTO 203,1");
    }
}
Else { // registro no existe en nuestra base, entonces es un
cliente nuevo
    write ("SET CONTEXT internos");
    write ("EXEC GoTO 2345,1"); //extension para clientes nuevos
}

fclose($in);
fclose($stdlog);
exit;
?>

```

ANEXO N

En el anexo N encontramos el código correspondiente al archivo mayorpromedio.php.

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

<?php
ob_implicit_flush(true);
set_time_limit(6);
$in = fopen("php://stdin","r");

$stdlog = fopen("/var/log/asterisk/acadi.log", "w");

function read()
{
    global $in, $debug;
    $input = str_replace("\n", "", fgets($in, 4096));
    return $input;
}
function errlog($line) {
    global $err;
    echo "VERBOSE \"$line\"\n";
}
function write($line) {
    global $debug;
    echo $line."\n";
}
function connect_db()
{
    $db_connection = mysql_connect ('localhost', 'root',
    'labtelecom10') or die (mysql_error());
    $db_select = mysql_select_db('asterisk') or die (mysql_error());
}
while ($env=read())
{
    $env = str_replace("\",\"", $env);
    $s = split(":", $env);
    $agi[str_replace("agi_", "", $s[0])] = trim($s[1]);
    if ($env == "")
    {
        break;
    }
}

// programa inicial

$cli = $agi[callerid];
$exten= $agi[extension];
```

```

connect_db();

$query1 = "select dstchannel from (select dstchannel,
avg(duration) dur from cdr where
dayofyear(calldate)=DAYOFYEAR(CURDATE()) AND dst='2020' or
dst='2021' group by dstchannel order by dur) as t where
dstchannel in (Select agent from (select o.agent, o.time_id from
queue_log o where o.verb = 'AGENTLOGIN' and o.call_id not in
(select q.call_id from queue_log q where q.verb = 'AGENTLOGOFF')
and o.agent not in (select t.agent from queue_log t where t.verb
= 'CONNECT' and t.call_id not in (select q.call_id from queue_log
q where q.verb = 'COMPLETECALLER' or q.verb = 'COMPLETEAGENT'))
AS e);";

$query_result1 = @mysql_query($query1);
$row_count = mysql_num_rows($query_result1);
$row1 = @mysql_fetch_array ($query_result1);

If ($row_count != 0 ) { //registro ya existe
    $tipo=$row1["dstchannel"];
    write ("SET CONTEXT internos");

    If ($tipo == "Agent/1001" ) {
        write ("EXEC GoTO 201,1");
    }
    If ($tipo == "Agent/1002" ) {
        write ("EXEC GoTO 202,1");
    }
    If ($tipo == "Agent/1003" ) {
        write ("EXEC GoTO 203,1");
    }
}
Else { // registro no existe en nuestra base, entonces es un
cliente nuevo

    $query2 = "Select agent from (select o.agent, o.time_id from
queue_log o where o.verb = 'AGENTLOGIN' and o.call_id not in
(select q.call_id from queue_log q where q.verb =
'AGENTLOGOFF') and o.agent not in (select t.agent from queue_log
t where t.verb = 'CONNECT' and t.call_id not in (select
q.call_id from queue_log q where q.verb = 'COMPLETECALLER' or
q.verb = 'COMPLETEAGENT')) ORDER BY o.time_id) AS
e";
    $query_result2 = @mysql_query($query2);
    $row_count1 = mysql_num_rows($query_result2);
    $row2 = @mysql_fetch_array ($query_result2);

    If ($row_count1 != 0 ) { //registro ya existe
        $tipo=$row2["agent"];
        write ("SET CONTEXT internos");
        If ($tipo == "Agent/1001" ) {
            write ("EXEC GoTO 201,1");
        }
    }
}

```

```
        If ($tipo == "Agent/1002" ) {
            write ("EXEC GoTO 202,1");
        }
        If ($tipo == "Agent/1003" ) {
            write ("EXEC GoTO 203,1");
        }
    }
    Else { // registro no existe en nuestra base, entonces es un
cliente nuevo
        write ("SET CONTEXT internos");
        write ("EXEC GoTO 2345,1"); //extension para clientes
nuevos
    }
}

fclose($in);
fclose($stdlog);
exit;
?>
```


ANEXO O

En el anexo O encontramos el código correspondiente al archivo menorpromedio.php.

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

<?php
ob_implicit_flush(true);
set_time_limit(6);
$in = fopen("php://stdin","r");

$stdlog = fopen("/var/log/asterisk/acadi.log", "w");

function read()
{
    global $in, $debug;
    $input = str_replace("\n", "", fgets($in, 4096));
    return $input;
}
function errlog($line) {
    global $err;
    echo "VERBOSE \"$line\"\n";
}
function write($line) {
    global $debug;
    echo $line."\n";
}
function connect_db()
{
    $db_connection = mysql_connect ('localhost', 'root',
    'labtelecom10') or die (mysql_error());
    $db_select = mysql_select_db('asterisk') or die (mysql_error());
}
while ($env=read())
{
    $env = str_replace("\",\"", $env);
    $s = split(":", $env);
    $agi[str_replace("agi_", "", $s[0])] = trim($s[1]);
    if ($env == "")
    {
        break;
    }
}

// programa inicial

$cli = $agi[callerid];
$exten= $agi[extension];
```

```

connect_db();

$query1 = "select dstchannel from (select dstchannel,
avg(duration) dur from cdr where
dayofyear(calldate)=DAYOFYEAR(CURDATE()) AND dst='2020' or
dst='2021' group by dstchannel order by dur desc) as t where
dstchannel in (Select agent from (select o.agent, o.time_id from
queue_log o where o.verb = 'AGENTLOGIN' and o.call_id not in
(select q.call_id from queue_log q where q.verb = 'AGENTLOGOFF')
and o.agent not in (select t.agent from queue_log t where t.verb
= 'CONNECT' and t.call_id not in (select q.call_id from queue_log
q where q.verb = 'COMPLETECALLER' or q.verb = 'COMPLETEAGENT'))
AS e);";
$query_result1 = @mysql_query($query1);
$row_count = mysql_num_rows($query_result1);
$row1 = @mysql_fetch_array ($query_result1);

If ($row_count != 0 ) { //registro ya existe
    $tipo=$row1["dstchannel"];
    write ("SET CONTEXT internos");
    If ($tipo == "Agent/1001" ) {
        write ("EXEC GoTO 201,1");
    }
    If ($tipo == "Agent/1002" ) {
        write ("EXEC GoTO 202,1");
    }
    If ($tipo == "Agent/1003" ) {
        write ("EXEC GoTO 203,1");
    }
}
Else { // registro no existe en nuestra base, entonces es un
cliente nuevo

    $query2 = "Select agent from (select o.agent, o.time_id from
queue_log o where o.verb = 'AGENTLOGIN' and o.call_id not in
(select q.call_id from queue_log q where q.verb =
'AGENTLOGOFF') and o.agent not in (select t.agent from queue_log
t where t.verb = 'CONNECT' and t.call_id not in (select
q.call_id from queue_log q where q.verb = 'COMPLETECALLER' or
q.verb = 'COMPLETEAGENT')) ORDER BY o.time_id) AS
e";
    $query_result2 = @mysql_query($query2);
    $row_count1 = mysql_num_rows($query_result2);
    $row2 = @mysql_fetch_array ($query_result2);

    If ($row_count1 != 0 ) { //registro ya existe
        $tipo=$row2["agent"];
        write ("SET CONTEXT internos");

        If ($tipo == "Agent/1001" ) {
            write ("EXEC GoTO 201,1");
        }
        If ($tipo == "Agent/1002" ) {

```

```
        write ("EXEC GoTO 202,1");
    }
    If ($tipo == "Agent/1003" ) {
        write ("EXEC GoTO 203,1");
    }
}
Else { // registro no existe en nuestra base, entonces es un
cliente nuevo

        write ("SET CONTEXT internos");
        write ("EXEC GoTO 2345,1"); //extension para clientes
nuevos
    }
}

fclose($in);
fclose($stdlog);
exit;
?>
```

BIBLIOGRAFÍA

[1] Wikipedia, Definición de Asterisk, <http://es.wikipedia.org/wiki/Asterisk>, consultado en Julio del 2011.

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

[3] Wikipedia, Definición de ACD,

http://en.wikipedia.org/wiki/Automatic_call_distributor, consultado en Julio del 2011.

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

[5] Queumetrics, Definición Qloader,

<http://queumetrics.com/manuals/QLoader-chunked/>, consultado en Diciembre de 2011.

[6] Queumetrics, Descarga del script Qloader,

<http://queumetrics.com/download/qloaderd-1.27.tar.gz>, consultado en Diciembre 2011.

[7] Asterisk, Descarga de los componentes de Asterisk,

<http://www.asterisk.org/downloads>, consultado en Octubre del 2011.

[8] Flavio E. Gonçalves, Configuration Guide for Asterisk™ PBX, V. Office Networks Ltda., 2007, pp. 29-30, 68-71, 170, 274-288.

[9] Voip-Info.org, Información agents.conf, <http://www.voip->

[info.org/wiki/view/Asterisk+Agents](http://www.voip-info.org/wiki/view/Asterisk+Agents), consultado en Noviembre del 2011.

[10] Voip-Info.org, Información colas, <http://www.voip-info.org/wiki/view/Asterisk+call+queues>, consultado en Noviembre del 2011.

[11] Voip-Info.org, Información queue_log, http://www.voip-info.org/wiki/view/Asterisk+log+queue_log, consultado en Noviembre del 2011.