



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

INFORME DE MATERIA DE GRADUACIÓN

**“IMPLEMENTACIÓN DE UN MÓDULO QUE PERMITA ELABORAR
REPORTES PERSONALIZADOS DEL CDR DE ASTERISK SOBRE
PLATAFORMA LAMPA”**

Previa a la obtención del Título de:

INGENIERO EN TELEMÁTICA

Presentada por:

JOSÉ LUIS CASTRO NAULA

HENRY PAÚL BARRERA CRIOLLO

GUAYAQUIL – ECUADOR

AÑO

2009

AGRADECIMIENTO

A Dios nuestro Padre.

A la Escuela Superior Politécnica del Litoral y sus Profesores por su valioso aporte en nuestro crecimiento académico.

Con un profundo sentimiento de gratitud a la Ing. Rebeca Estrada, Directora del presente Informe de la Materia de Graduación, por el tiempo y esfuerzo invertido.

A todos quienes de una u otra manera aportaron en nuestro trabajo, en especial al Ing. Gabriel Astudillo, por su colaboración en la realización de nuestro informe final.

DEDICATORIA

A Dios.

A mis Padres.

A mi enamorada Mónica Rojas.

José L. Castro Naula

A Dios mi creador por su infinita bondad y fidelidad.

A Carmen Barrera, mi señora madre, quien con su esfuerzo y amor, ha sido mi guía y motivación.

A la Fundación San Luis, al Grupo Pronaca, a mis grandes y apreciados amigos.

Henry P. Barrera Criollo

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis 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)



Handwritten signature of José Luis Castro Naula, consisting of a large, stylized 'J' and 'C' followed by 'Luis Castro Naula' in cursive.

José Luis Castro Naula



Handwritten signature of Henry Paul Barrera Criollo, featuring a large, stylized 'H' and 'B' with the number '3' integrated into the signature.

Henry Paul Barrera Criollo

TRIBUNAL DE GRADUACIÓN



Ing. Rebeca Estrada

PROFESOR DIRECTOR DE LA MATERIA DE GRADUACIÓN



Ing. Patricia Chavez

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

R E S U M E N

El presente trabajo tiene como objetivo principal, implementar un módulo que permita elaborar reportes de registros de llamadas de Asterisk sobre la plataforma LAMPA.

Aquello permitirá proporcionar un ambiente útil y fácil de manejar los reportes del CDR, poder realizar futuras configuraciones de acuerdo a la necesidad del usuario, presentar la información de los registros de llamadas de manera efectiva y actualizada, permitiendo además, el acceso a la aplicación de manera confiable y segura desde cualquier sitio.

ÍNDICE GENERAL

RESUMEN

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

INTRODUCCIÓN

CAPÍTULO I

ANTECEDENTES Y JUSTIFICACIÓN.....	12
1.1 Antecedentes.....	13
1.2 Justificación.....	15
1.3 Descripción del Proyecto.....	16
1.4 Objetivos.....	18
1.4.1 Objetivos Generales.....	18
1.4.2 Objetivos Específicos.....	18
1.5 Metodología.....	19

CAPÍTULO II

REGISTRO DE DETALLE DE LLAMADAS.....	20
2.1 CDR (Call Detail Records).....	21
2.1.1 Campos del CDR.....	21

2.2 Almacenamiento del CDR en una Base de Datos.....	23
2.2.1 Conectores de Módulos de Base de Datos.....	24
2.3 Archivos Comunes de Configuración del CDR.....	26
2.4 Registros Adicionales del CDR.....	31

CAPÍTULO III

IMPLEMENTACIÓN DEL PROYECTO.....	34
3.1 LAMPA y su Relación con XAMPP.....	35
3.1.1 XAMPP para Linux.....	36
3.1.2 XAMPP para Windows.....	36
3.1.3 XAMPP para Mac OS X.....	36
3.1.4 XAMPP para Solaris.....	37
3.2 Parámetros Avanzados para Empezar o Parar XAMPP.....	39
3.3 Lista de Carpetas y Archivos Importantes de XAMPP.....	42

CAPÍTULO IV

FUNCIONAMIENTO Y PRUEBAS DEL PROYECTO.....	44
4.1 Archivos de Configuración de Asterisk.....	45
4.1.1 Descripción del Archivo de Conexión CDR MySQL.....	45
4.1.2 Descripción del Archivo Protocolo de Iniciación de Sesión.....	45
4.1.3 Descripción del Archivo Protocolo de Inter Asterisk.....	50

4.1.4 Descripción del Archivo de Extensiones del CDR.....	51
4.1.5 Descripción del Archivo de Conferencia.....	54
4.1.6 Descripción del Archivo de Correo de Voz.....	54
4.2 Creación de la Estructura de la Base de Datos.....	55
4.3 Clase Conexión PHP.....	58

CONCLUSIONES

RECOMENDACIONES

BIBLIOGRAFÍA

ÍNDICE DE FIGURAS

Fig. 1.1 Diagrama General del Proyecto... ..	17
Fig. 1.2 Estructura LAMPA... ..	17
Fig. 3.1 Interfaz Principal de XAMPP... ..	41
Fig. 4.1 Creación de Tablas del CDR... ..	56
Fig. 4.2 Campos de la Tabla de Usuarios	57
Fig. 4.3 Módulo Principal del CDR... ..	60
Fig. 4.4 Reporte de los Campos del CDR	61
Fig. 4.5 Gráficos Estadísticos del CDR.....	62

INTRODUCCIÓN

El presente documento tiene como objetivo brindar las bases teóricas y conceptuales para el desarrollo e implementación de un módulo que permita elaborar reportes del CDR (Call Detail Records) de Asterisk sobre la plataforma LAMPA, con lo cual se podrá visualizar, analizar y manipular la información de los registros de las llamadas sobre una interfaz mucho más amigable para el usuario.

La tesis está dividida en cuatro capítulos, los cuales contienen los elementos que fueron utilizados para el análisis, diseño e implementación del módulo antes mencionado, útil para cualquier PBX u empresa que tenga instalada Voz sobre IP.

CAPÍTULO I

ANTECEDENTES Y JUSTIFICACIÓN

1.1 ANTECEDENTES

En el amplio mundo de las comunicaciones y en específico, la telefonía de voz sobre IP, podemos encontrar algunas aplicaciones open source y de propietario que permiten gestionar el CDR, desarrollar una propia será nuestra tarea a cumplir.

Tradicionalmente, la generación y manejo de los CDR ha sido conocido en los EE.UU. como de contabilidad automática de mensajes o la AMA, un sistema que se remonta a la década de 1940.

Hoy en día, PBX y conmutadores de software en su mayoría generan los CDR, ya sea en formatos delimitados por comas o que se escriben directamente en una base de datos.

Entre las aplicaciones de software libre que tenemos podemos mencionar:

Astbill: es una de las mejores aplicaciones opensource para tarificación, control de cuentas y llamadas.

Areski Stat v2: se trata de una aplicación para listar y realizar estadísticas de las llamadas realizadas o enviadas.

A2Billing: es un completo sistema de tarificación.

Y entre las de propietario tenemos:

VoIPManager: es una interfaz de administración gráfica para Asterisk, que permite configurar y administrar sólo como un PBX convencional, pero no va a interferir con las personalizaciones avanzadas de VoIP que puede que desee aplicar.

Voipswitch es una plataforma de software que permite a los servicios de VoIP rápido despliegue. Contiene todos los elementos necesarios para una implementación exitosa de los diversos servicios de VoIP.

1.2 JUSTIFICACIÓN

Ante la necesidad de visualizar la información generada por el CDR, resulta importante implementar un módulo que permita al usuario revisar los registros detallados de llamadas, tanto locales y de larga distancia, puede ser utilizado para la verificación de uso, la reconciliación de facturación, gestión de red y para supervisar el uso de teléfonos, para determinar el volumen de uso del teléfono, así como también el abuso del sistema telefónico de su empresa u organización.

Su implementación también permitirá la planificación de las necesidades de telecomunicaciones del futuro.

1.3 DESCRIPCIÓN DEL PROYECTO

La implementación del Módulo del CDR bajo la plataforma LAMPA, cuenta de tres etapas. En la primera parte, se registran las correspondientes llamadas en un archivo específico de Asterisk en un formato particular, sean estas dentro o fuera de la red de nuestra compañía.

En la segunda parte, tenemos el almacenamiento de los registros en una base de datos, que para nuestro proyecto será MySQL. Finalmente en la última parte, tenemos la implementación de nuestra interfaz de visualización, desarrollada bajo el código en PHP, para las respectivas consultas o análisis de la información de los registros de las llamadas.

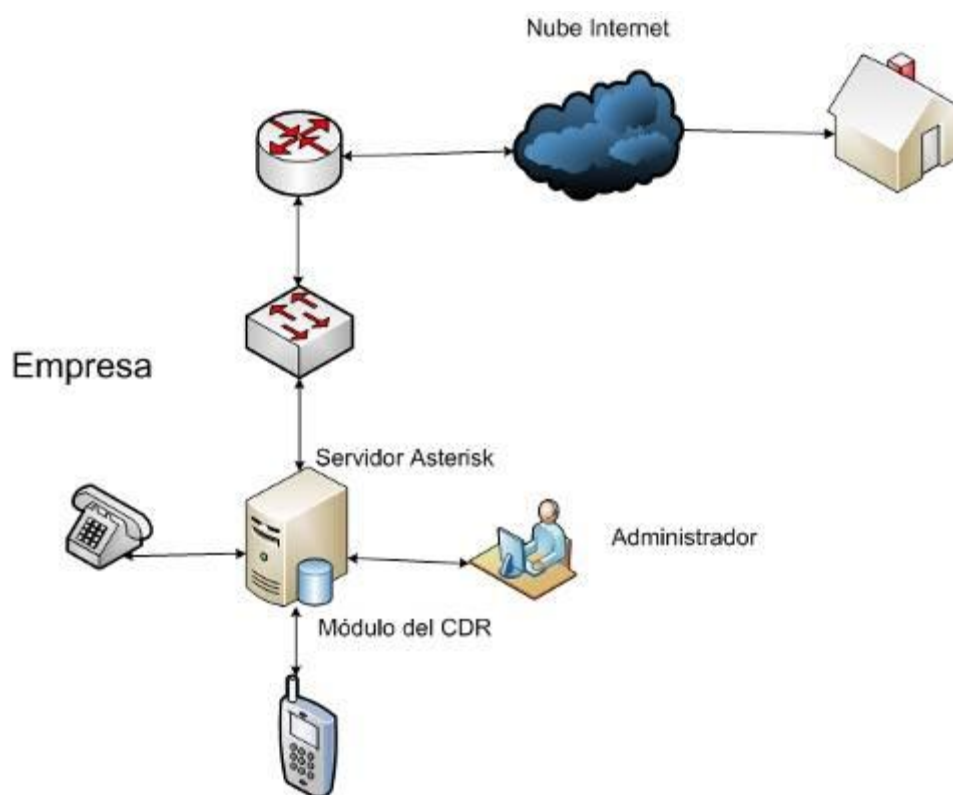


Fig.1.1 Diagrama General del Proyecto

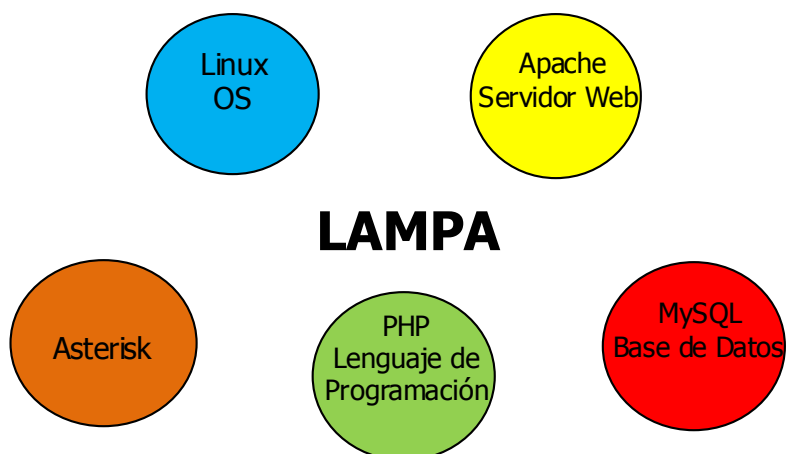


Fig.1.2 Estructura LAMPA

1.4 OBJETIVOS

1.3.1 Objetivos Generales

- Implementar un módulo que permita elaborar reportes del CDR de Asterisk sobre la plataforma LAMPA.

1.3.2 Objetivos Específicos

- Proporcionar un ambiente útil y fácil de manejar los reportes del CDR para el usuario final.
- Poder realizar configuraciones de acuerdo a la necesidad del usuario.
- Presentar la información de los registros de llamadas de manera efectiva y actualizada.
- Permitir el acceso a la aplicación con seguridad y confiabilidad desde cualquier sitio.

1.5 METODOLOGÍA

En nuestro desarrollo del módulo para la elaboración de reporte CDR (Call Detail Records) de Asterisk, tendremos que implementarlo bajo el sistema operativo de Linux, instalar Apache como nuestro servidor web, MySQL como gestor de base de datos, PHP nuestro lenguaje de programación y Asterisk como nuestro PBX.

Posteriormente realizar las configuraciones necesarias en los archivos CDR de asterisk para su conexión con la base de datos de MySQL.

La creación de nuestra tabla con los respectivos campos y parámetros requeridos y finalmente iniciar el código de programación en PHP para obtener al final nuestro módulo de administración de los reportes del CDR.

CAPÍTULO II
REGISTRO DE DETALLE DE LLAMADAS

2.1 CDR (CALL DETAIL RECORDS)

Un registro de detalle de llamadas (CDR) es el registro informático producido por una central telefónica con los detalles de una llamada originada, interna o externa del lugar de operaciones del PBX.

Asterisk normalmente almacena el Registro de Detalles de Llamadas (CDR) en valores separados por coma en el archivo (CSV). Si desea que el CDR sea almacenado en una base de datos, tendrá que cargar el módulo apropiado y definir el archivo en `cuestión.conf`.

2.1.1 CAMPOS DEL CDR

1. **accountcode**: ¿Qué número de cuenta de usar: cuenta de facturación de Asterisk,
2. **src**: Caller * número de identificación.
3. **dst**: Extensión de destino.
4. **dcontext**: Contexto de destino.
5. **clid**: Caller * ID con el texto.

6. **channel:** Canal utilizado.
7. **dstchannel:** Canal de destino, si procede.
8. **lastapp:** La última aplicación en su caso.
9. **lastdata:** La última solicitud de datos.
10. **start:** Inicio de la llamada (fecha/hora).
11. **answer:** Respuesta de la llamada (fecha/hora).
12. **end:** Fin de la llamada (fecha/hora).
13. **duration:** Tiempo total en el sistema, en segundos (entero), de marcar para colgar.
14. **billsec:** Llamada en tiempo total es, en cuestión de segundos (entero), de respuesta a colgar
15. **disposition:** ¿Qué pasó con la llamada?
RESPONDIDAS, NO RESPONDIDAS, OCUPADA, FALLIDA (en algunos backends CDR, por ejemplo, ODBC, estos pueden ser números enteros; en cuenta que información más detallada se puede encontrar en la variable \$ dialplan HANGUPCAUSE).

16. **amaflags:** ¿Qué parámetros de uso?
FACTURACIÓN, IGNORAR, etc. Especificado en una base por canal como accountcode.

17. **user field:** Un usuario definido por el campo.

2.2 ALMACENAMIENTO DEI CDR EN UNA BASE DE DATOS

El CDR también puede ser almacenado en una base de datos. Asterisk soporta actualmente SQLite, PostgreSQL, entre otros.

Los detalles de configuración de estas bases de datos no serán cubiertos en este proyecto, sino que se describen en el código fuente de Asterisk, en el marco del doc / subdirectorío. Por razones de licencia, cdr_mysql en asterisk-addons.), muchas personas prefieren almacenar sus CDR en una base de datos porque esto hace que sea más fácil la consulta específica para ellos.

Para cada módulo, nos indicará el tipo de base de datos que soporta, y especificar la configuración del archivo, si es necesario.

2.2.1 CONECTORES DE MÓDULOS DE BASE DE DATOS DEL CDR

cdr_csv.so cdr_csv.so

Proporciona: CSV CDR backend

cdr_custom.so cdr_custom.so

Fichero de configuración: cdr_custom.conf

Proporciona: Personalizable CSV CDR backend

cdr_manager.so cdr_manager.so

Fichero de configuración: cdr_manager.conf

Proporciona: Asterisk Call Manager CDR backend

cdr_odbc.so

Fichero de configuración: cdr_odbc.conf

Proporciona: ODBC CDR backend

cdr_pgsql.so cdr_pgsql.so

Fichero de configuración: cdr_pgsql.conf

Proporciona: PostgreSQL CDR backend

El código de cuenta puede ser definida en función de cada usuario. Si se define esta cuenta será asignado a un registro de llamadas. Cuando no hay código de cuenta específica de usuario, se establece. El nombre accountcode configurado será utilizado como nombreDeArchivo.csv en /var / asterisk / log / cdr-csv / directorio para almacenar el registro de llamadas (CDR) para el user / peer/ friend.

El conector cdr_odbc teóricamente podría sustituir todos los conectores de base de datos específica-sin embargo, la gente puede preferir el uso de conectores específicos debido a las diferencias de rendimiento, problemas de estabilidad, la preferencia personal, compatibilidad hacia

atrás, y así sucesivamente. Muchas opciones están disponibles. Si usted está familiarizado con bases de datos, Asterisk te da muchas opciones.

2.3 ARCHIVOS COMUNES DE CONFIGURACIÓN DEL CDR

cdr.conf

El archivo de cdr.conf se utiliza para habilitar el registro de llamadas de detalle a una base de datos. El almacenamiento de la llamada es útil para todo tipo de propósitos, incluyendo la facturación, la prevención del fraude, calidad de servicio, evaluaciones, y más. El archivo cdr.conf contiene algunos parámetros generales que no son específicos a cualquier base de datos en particular, sino que más bien indican como Asterisk debe manejar el paso de información a la base de datos.

Todas las opciones están en el marco del [general] de la partida del archivo cdr.conf, veamos sus campos:

Batch

Acepta los argumentos sí y no. Asterisk permite escribir datos en un buffer en lugar de la escritura a la base de datos al final de cada llamada, para reducir la carga en el sistema.

Tenga en cuenta que si el sistema muere inesperadamente cuando esta opción está establecida en yes, la pérdida de datos puede ocurrir.

Enable

Acepta los argumentos sí y no. Especifica si utilizar o no el registro CDR. Si establece en no, esto anula cualquier módulo CDR explícitamente cargados. El valor por defecto es sí.

Safeshutdown

Acepta los argumentos sí y no. La configuración safeshutdown a sí impedirá que Asterisk se apague por completo hasta que el buffer se vacíe y toda la información se escriba en la base de datos. Si este parámetro se establece en No, la información es probable que se pierda.

Scheduleroonly

Acepta los argumentos sí y no. Si está generando un gran volumen de registros de llamadas en un sistema que está empujando a una base de datos remota, el establecimiento de sí puede ser de beneficio. Dado que el programador no se puede iniciar una nueva tarea hasta que se termine el actual, escribir puede afectar negativamente a otros procesos que requiera el planificador.

Size

Acepta un entero como argumento. Define el número de CDRs para acumular en el búfer antes de escribir en la base de datos. El valor por defecto es 100.

Time

Acepta un número entero (en segundos) como argumento. Establece el número de segundos antes de que Asterisk vacíe el buffer y escribe los CDR para la base de datos, independientemente del número de registros.

cdr_manager.conf

El archivo `cdr_manager.conf` sólo contiene un `[general]` partida y una única opción activado, que puede utilizar para especificar si la API de Asterisk Manager genera eventos. Si desea que los eventos del CDR que se genere, tendrá las siguientes líneas en el archivo `cdr_manager.conf`:

```
[general] [general]
```

```
enabled=yes Enabled = sí
```

cdr_odbc.conf

Asterisk puede almacenar los datos de CDR en una base de datos local o remota a través de la interfaz ODBC. Contiene el archivo

cdr_odbc.conf. Asterisk necesita de información para conectarse a la base de datos. El módulo de cdr_odbc.so intentará cargar el archivo cdr_odbc.conf, y si la información se encuentra para conectar a una base de datos, los datos del CDR se registrarán allí.

Si se va a utilizar una base de datos para almacenar datos de CDR, que tendrá que seleccionar una de las muchas que están disponibles. Asterisk no le gusta tener CDR múltiples bases de datos para conectarse, por lo que no tienen cdr_.conf extra para colgar archivos de tu directorio de configuración de Asterisk.

cdr_pgsql.conf

Asterisk puede almacenar los datos de CDR en una base de datos PostgreSQL a través del módulo cdr_pgsql.so.

Cuando el módulo se carga la información necesaria se puede leer en el archivo cdr.pgsql.conf y Asterisk se conectará a la base de datos PostgreSQL para escribir y almacenar los datos del CDR.

cdr_tds.conf

Asterisk puede también almacenar los datos de los CDR a una base de datos FreeTDS (incluidos los de MS SQL) con el uso del módulo de cdr_tds.so. El archivo cdr_tds.conf de configuración se lee una vez que el módulo de está cargado. Tras una conexión exitosa, los datos de CDR se escribirán en la base de datos.

2.4 REGISTROS ADICIONALES DEL CDR

ForkCDR () Crea un CDR adicional de la llamada actual.

Esta aplicación se utiliza a menudo en aplicaciones de tarjeta de llamadas para distinguir las llamadas entrantes (el original CDR) desde el momento llamada "facturable" (segundo de los CDR).

```
exten => 123,1,Answer( )
```

```
exten => 123,2,ForkCDR( )
```

```
exten => 123,3,Playback( )
```

NoCDR () Desactiva el CDR para la llamada actual.

Ejemplo, no registrar las llamadas al 555-1212

```
exten => 5551212,1,Answer( )
```

```
exten => 5551212,2,NoCDR( )
```

```
exten => 5551212,3,Dial(Zap/4/5551212)
```

ResetCDR () Restablece el Registro de Detalle de Llamadas.

Hace que el registro de detalle de llamadas se coloque a cero. Si la opción-w no se especifica, una copia de la actual, el CDR será almacenado antes que la actual sea puesta a cero.

Escribir una copia del registro actual del CDR:

```
exten => 123,1,Answer( )
```

```
exten => 123,2,Playback(tt-monkeys)
```

```
exten => 123,3,ResetCDR(w)
```

```
exten => 123,4,Playback(tt-monkeys)
```


SetCDRUserField () Establece el campo de detalle de llamadas.

Establece el campo de usuario CDR en el valor especificado. El campo de usuario CDR es un campo adicional que usted puede utilizar para que los datos no se almacenen en otro sitio del registro. El CDR se puede utilizar para la facturación, almacenamiento de otros datos de carácter arbitrario de una llamada en particular.

```
exten => 123,1,SetCDRUserField(testing)
```

```
exten => 123,2,Playback(tt-monkeys)
```

CAPÍTULO III

IMPLEMENTACIÓN DEL PROYECTO

3.1 LAMPA Y SU RELACIÓN CON XAMPP

LAMPA (Linux, Apache, MySQL, PHP / Perl, Asterisk) resulta ser un software que se está convirtiendo en una popular plataforma de desarrollo Web. Esto es comprensible dado que cuenta con un sistema operativo libre, servidor web, base de datos y lenguaje de scripting. Sin embargo, un problema de tener un número de separado de componentes de código abierto es la integración. El proyecto de XAMPP aspira eliminar este problema. Con XAMPP, no hay ninguna razón para que los desarrolladores alcancen una solución de código abierto al considerar las plataformas de desarrollo Web.

La descarga de XAMPP incluye más que sólo Apache, MySQL, PHP y Perl. También incluye una serie de paquetes relacionados con código abierto, tales como: OpenSSL, PEAR, phpMyAdmin, ProFTPD, y Webalizer.

XAMPP solamente requiere descargar y ejecutar un archivo zip, tar, o exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl.

3.1.1 XAMPP para Linux

La versión para sistemas Linux (testeados para SuSE, RedHat, Mandrake y Debian) contiene: Apache, MySQL, PHP & PEAR, Perl, ProFTPD, phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, gdbm, zlib, expat, Sablotron, libxml, Ming, Webalizer, pdf class, ncurses, mod_perl, FreeTDS, gettext, mcrypt, mhash, eAccelerator, SQLite e IMAP C-Client.

3.1.2 XAMPP para Windows

Versión para Windows 98, NT, 2000, XP y Vista. Esta versión contiene: Apache, MySQL, PHP + PEAR, Perl, mod_php, mod_perl, mod_ssl, OpenSSL, phpMyAdmin, Webalizer, Mercury Mail Transport System para Win32 y NetWare Systems v3.32, JpGraph, FileZilla FTP Server, mcrypt, eAccelerator, SQLite, y WEB-DAV + mod_auth_mysql.

3.1.3 XAMPP para Mac OS X

La versión para Mac OS X contiene: Apache, MySQL, PHP & PEAR, SQLite, Perl, ProFTPD, phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, zlib, Ming, Webalizer, mod_perl, eAccelerator.

3.1.4 XAMPP para Solaris

La versión para Solaris (desarrollada y probada con Solaris 8, probada con Solaris 9) contiene: Apache, MySQL, PHP & PEAR, Perl, ProFTPD, phpMyAdmin, OpenSSL, Freetype2, libjpeg, libpng, zlib, expat, Ming, Webalizer, pdf class.

Una vez instalado, XAMPP coloca un script de inicio con el nombre de lampp en el `<nobr> <wbr> </ nobr> / opt / lampp`. Esta secuencia de comandos se inicia, se detiene, recarga, y comprueba el estado de XAMPP. Por ejemplo, considérese el siguiente comando:

```
<nobr> <wbr></nobr> /opt/lampp/lampp start <nobr> <wbr> </ nobr> /opt/lampp/lampp start
```

Esto inicia los servicios básicos XAMPP de Apache (con soporte para PHP), MySQL y ProFTPD. Para ver las otras opciones que puede pasar a XAMPP ejecutar el script con el `-- help` como opción. La mayoría de las opciones se explica por sí mismo, pero vamos a echar un vistazo a dos que pueden afectar el rendimiento de su sitio web y la seguridad.

El acceso al servidor en `http://localhost`, trae la página de administración de XAMPP. El marco en el lado izquierdo de la página se divide en tres

secciones. La primera es XAMPP específicos. Desde aquí, se puede comprobar el estado de los servicios que los controles de XAMPP, verificar su configuración de seguridad, leer la documentación de XAMPP, y ver una lista de todos los programas instalados por XAMPP.

La sección de demo contiene siete programas de ejemplo que utilizan componentes de XAMPP. Estos programas simples que no están destinadas a mostrar la potencia de una plataforma LAMP. Su propósito es simplemente para comprobar que los diversos componentes están trabajando juntos correctamente. Cuatro de las demos tienen vínculos código fuente que proporcionan el código PHP que utilizan. Se puede dar a cualquier persona con experiencia mínima de PHP una visión rápida de su sintaxis.

En la sección de herramientas, puede administrar MySQL y SQLite con phpMyAdmin y phpSQLiteAdmin respectivamente, basadas en interfaces Web para la gestión de base de datos. Su integración hace que XAMPP sea competitiva y exclusiva para entornos de desarrollo Web, ya que eliminan la necesidad de un experto en SQL dedicado a realizar comandos de administración de bases de datos en línea.

La última herramienta que se incluye es Webalizer, un rápido registro de servidor Web de análisis de archivo de programa escrito en C. Se puede personalizar su producción mediante la edición del `webalizer.conf`, archivo ubicado en el directorio `etc` de la instalación de XAMPP.

3.2 Parámetros Avanzados para Empezar o Parar XAMPP

Start:	Inicia XAMPP.
Stop:	Detiene XAMPP.
Restart:	Reinicia XAMPP.
Startapache:	Inicia solamente Apache
Startssl:	Inicia el soporte de SSL de Apache.
Startmysql:	Inicia únicamente la base de datos MySQL
Startftp:	Inicia el ProFTPD server, puedes subir tus archivos vía ftp.

Stopapache:	Detiene Apache.
Stopssl:	Detiene el soporte de SSL de Apache.
Stopmysql:	Detiene MySQL.
Stopftp:	Detiene el ProFTPD server.
Security:	Inicia el programa anteriormente mencionado para establecer una mejor seguridad.

Para correr la aplicación, desde la consola escribimos el siguiente comando:

```
/opt/lampp/lampp start
```


Vista Principal de XAMPP

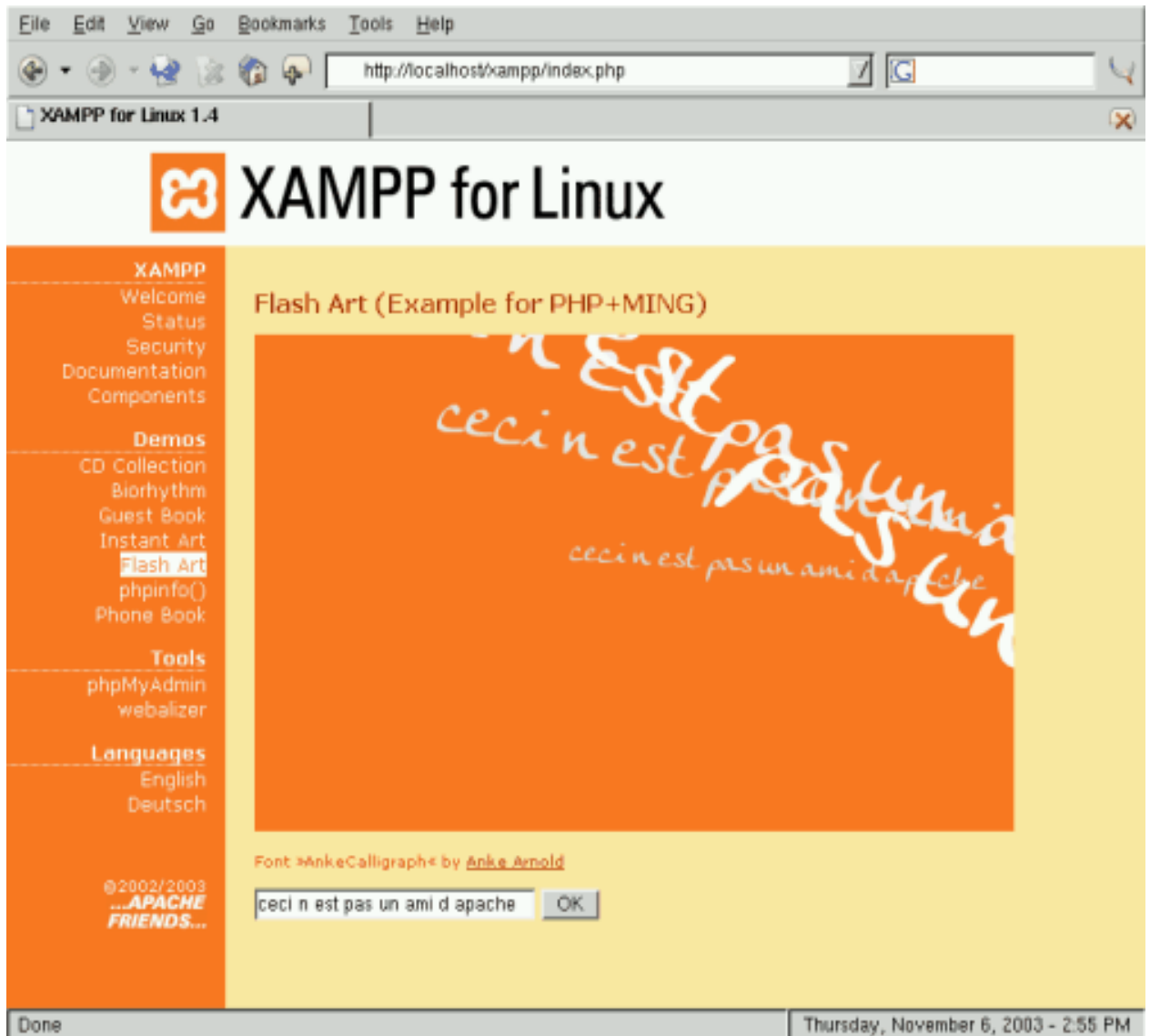


Fig. 3.1 Vista Principal de XAMPP

3.3 Lista de Carpetas y Archivos Importantes de XAMPP

`/opt/lampp/bin/`

Es la carpeta en donde se encuentran los comandos de XAMPP. Por ejemplo el código `/opt/lampp/bin/mysql` invoca el monitor de MySQL.

`/opt/lampp/htdocs/`

Es el directorio donde se encuentran los archivos root de Apache.

`/opt/lampp/etc/httpd.conf`

Archivo de configuración de Apache.

`/opt/lampp/etc/my.cnf`

Archivo de configuración de MySQL.

`/opt/lampp/etc/php.ini`

Archivo de configuración de PHP.

`/opt/lampp/etc/proftpd.conf` Archivo de configuración de PROFTPD.

`/opt/lampp/phpmyadmin/config.inc.php` Archivo de configuración de phpMyAdmin.

Oficialmente, los diseñadores de XAMPP sólo pretendían su uso como una herramienta de desarrollo para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet.

En la práctica, sin embargo, XAMPP es utilizado actualmente para servidor de sitios Web y, con algunas modificaciones, es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes. Es básico para un programador web.

CAPÍTULO IV
FUNCIONAMIENTO Y PRUEBAS DEL
PROYECTO

4.1 ARCHIVOS DE CONFIGURACIÓN DE ASTERISK

4.1.1 Descripción del Archivo de Conexión CDR MySQL

Para iniciar nuestro proyecto debemos configurar el fichero `cdr_mysql.conf` para que apunte a nuestra base de datos. A continuación el detalle del mismo:

```
[global]
hostname=localhost
dbname=asterisk
table=cdr
password=henry
user=henry
port=3306
sock= /opt/lampp/var/mysql/mysql.sock
;userfield=1
```

4.1.2 Descripción del Archivo Protocolo de Iniciación de Sesión

Archivo correspondiente del CDR, sip.conf.

[general]

context=default

srvlookup=yes ; habilita la busqueda en el DNS para

language=es

register => pbx1:welcome@200.126.13.221/pbx2

register => pbx1:welcome@200.126.13.214/PBX3

[1001] ;cliente SIP 201 acepta valores alfanumericos

type=friend ; opciones : friend(llamar y recibir), user(recibir
llamadas),peer(make call)

secret=1001 ; password que va a tener el cliente

qualify=yes ; Qualify peer is no more than 2000 ms away

nat=no

host=dynamic ; el ususario se registrara con nosotros

canreinvite=no ;

context=internal ;

[1002] ;cliente SIP 201 acepta valores alfanumericos
type=friend ; opciones : friend(llamar y recibir), user(recibir
llamadas),peer(make call)
secret=1002 ; password que va a tener el cliente
qualify=yes ; Qualify peer is no more than 2000 ms away
nat=no ;
host=dynamic ; el usuario se registrara con nosotros
canreinvite=no ;
context=internal ;

[1003] ; cliente SIP 201 acepta valores alfanumericos
type=friend ; opciones : friend(llamar y recibir), user(recibir
llamadas),peer(make call)
secret=1003 ; password que va a tener el cliente
qualify=yes ; Qualify peer is no more than 2000 ms away
nat=no ;
host=dynamic ; el usuario se registrara con nosotros
canreinvite=no ;
context=internal ;

[1004] ;cliente SIP 201 acepta valores alfanumericos

type=friend ; opciones : friend(llamar y recibir), user(recibir llamadas),peer(make call)

secret=1004 ; password que va a tener el cliente

qualify=yes ; Qualify peer is no more than 2000 ms away

nat=no ;

host=dynamic ; el ususario se registrara con nosotros

canreinvite=no ;

context=internal ;

[1005] ;cliente SIP 201 acepta valores alfanumericos

type=friend ; opciones : friend(llamar y recibir), user(recibir llamadas),peer(make call)

secret=1005 ; password que va a tener el cliente

qualify=yes ; Qualify peer is no more than 2000 ms away

nat=no ;

host=dynamic ; el ususario se registrara con nosotros

canreinvite=no ;

context=internal ;

[1006] ; cliente SIP 201 acepta valores alfanumericos

type=friend ; opciones : friend(llamar y recibir), user(recibir llamadas),peer(make call)


```
secret=1006 ; password que va a tener el cliente
qualify=yes ; Qualify peer is no more than 2000 ms away
nat=no ;
host=dynamic ; el usuario se registrara con nosotros
canreinvite=no ;
context=internal ;
```

Creando un contexto para la pbx 2

```
[pbx2]
type=friend
secret=welcome
context=from_pbx2
host=dynamic
disallow=all
allow=ulaw
;incominglimit=1
```

```
[PBX3]
type=friend
secret=welcome
```

```
context=from_PBX3  
    host=dynamic  
    disallow=all  
    allow=ulaw  
;incominglimit=1
```

4.1.3 Descripción del Archivo Protocolo de Inter Asterisk

Archivo correspondiente del CDR, iax.conf.

```
[general]  
bindport=4569  
srvlookup=yes  
autokill=yes  
language=es  
  
register => pbx1:welcome@200.126.13.221  
register => pbx1:welcome@200.126.13.214  
  
[1501]  
type=friend
```

secret=1501

qualify=yes

nat=no

host=dynamic

context=internal

[pbx2]

type=friend

secret=welcome

trunk=yes

context=from_pbx2

host=dynamic

deny=0.0.0.0/0.0.0.0

permit=200.126.13.221/255.255.255.128

[PBX3]

type=friend

secret=welcome

trunk=yes

context=from_PBX3

host=dynamic

deny=0.0.0.0/0.0.0.0

permit=200.126.13.214/255.255.255.128

4.1.4 Descripción del Archivo de Extensiones del CDR

Archivo correspondiente del CDR, extensions.conf.

[general]

language=es

[globals]

[macro-extensiones]

exten => s,1,Dial(\${ARG1},10,r)

exten => s,2,VoiceMail(u\${MACRO_EXTEN}@default)

exten => s,102,VoiceMail(b\${MACRO_EXTEN}@default)

[internal]

exten => _1[0-4]XX,1,Macro(extensiones,SIP/\${EXTEN})

exten => _1[5-9]XX,1,Macro(extensiones,IAX2/\${EXTEN})

exten => 500,1,VoiceMailMain()

exten => *100,1,Directory(default,internal,f) ; NOS PERMITE

BUSCAR A LOS USUARIOS

```
exten => *101,1,Directory(default,internal) ; Directorio
```

```
exten => 601,1,MeetMe(601,iMp,54321) ; CONFERENCIAS
```

```
include => remote
```

```
[remote]
```

```
exten => _2XXX,1,Dial(SIP/pbx2/${EXTEN},30)
```

```
exten => _2XXX,102,Dial(IAX2/pbx2/${EXTEN},30)
```

```
exten => _2XXX,n,Playback(hello)
```

```
exten => _2XXX,n,Hangup()
```

```
exten => _3XXX,1,Dial(SIP/PBX3/${EXTEN},30)
```

```
exten => _3XXX,102,Dial(IAX2/PBX3/${EXTEN},30)
```

```
exten => _3XXX,n,Playback(hello)
```

```
exten => _3XXX,n,Hangup()
```

```
exten => _5XXX,1,Dial(SIP/PBX3/${EXTEN},30)
```

```
exten => _5XXX,102,Dial(IAX2/PBX3/${EXTEN},30)
```

```
exten => _5XXX,n,Playback(hello)
```

```
exten => _5XXX,n,Hangup()
```

```
exten => _4XXX,1,Dial(SIP/pbx2/${EXTEN},30)
```

```
exten => _4XXX,102,Dial(IAX2/pbx2/${EXTEN},30)
```

```
exten => _4XXX,n,Playback(hello)
```

```
exten => _4XXX,n,Hangup()
```

```
exten => 600,1,Dial(SIP/pbx2/${EXTEN})
```

```
[from_pbx2]
```

```
include => internal
```

```
[from_PBX3]
```

```
include => internal; ir a asterisk y dar un module reload
```

```
pbx_config.so
```

4.1.5 Descripción del Archivo de Conferencias

Archivo correspondiente del CDR, meetme.conf.

```
[rooms]
```

```
conf => 600 ; Número de marcación para ingresar.
```

4.1.6 Descripción del Archivo de Correo de Voz

Archivo correspondiente del CDR, voicemail.conf.

[default]

language=es

1001=>1001, HenryBarrera, hbarrera@espol.edu.ec, 9652306@im.movistar.com.ec, tz=central|attach=yes

1004=>1004, LuisCastro, hbarrera@espol.edu.ec, 84009435@im.movistar.com.ec, tz=central|attach=yes

4.2 Creación de la estructura de la base de datos

La estructura contiene dos tablas, la primera contiene los campos propios del CDR, donde se almacenará la información del registro de las llamadas. Fig 4.1

La segunda tabla contiene los campos de registro de los usuarios que tendrán acceso a la aplicación. Fig 4.2

XAMPP para Linux 1.6.5a localhost / localhost / asterisk... Servidor: localhost Base de datos: asterisk

Estructura SQL Eliminar Buscar Generar una consulta Exportar Importar Diseñador Operaciones Privilegios

Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> cdr		15	MyISAM	latin1_swedish_ci	6.0 KB	-
<input type="checkbox"/> cdr_s_usuario		2	MyISAM	latin1_swedish_ci	1.0 KB	-
2 tabla(s)		Número de filas	MyISAM	latin1_swedish_ci	7.1 KB	0 Bytes

← Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Vista de impresión Diccionario de datos

⚠ Crear nueva tabla en la base de datos asterisk

Nombre: Número de campos:

Fig.4.1 Creación de Tablas del CDR

phpMyAdmin

Base de datos: asterisk (2)

cdtr_s_usuario

Servidor: localhost ▶ Base de datos: asterisk ▶ Tabla: cdtr_s_usuario

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id	smallint(6)		No			
<input type="checkbox"/>	login	varchar(20)	latin1_swedish_ci	No			
<input type="checkbox"/>	clave	varchar(10)	latin1_swedish_ci	No			
<input type="checkbox"/>	estado	varchar(1)	latin1_swedish_ci	No			

← Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Vista de impresión Vista de relaciones Planteamiento de la estructura de tabla

Añadir 1 campo(s) Al final de la tabla Después de id

Índices: ?

¡No se ha definido el índice!

Crear un índice en 1 columna(s)

Espacio utilizado		Estadísticas de la fila	
Tipo	Uso	Enunciado	Valor
Datos	40 Bytes	Formato	dinámico/a
Índice	1,024 Bytes	Cotejamiento	latin1_swedish_ci
Total	1,064 Bytes	Filas	2
Longitud de la fila 0		20	
Tamaño de la fila 0		532 Bytes	
Creación		26-08-2009 a las 00:21:02	
Última actualización		29-08-2009 a las 13:26:53	

Fig. 4.2 Campos de la Tabla de Usuarios

4.3 Clase Conexión PHP

La clase conexión implementada en PHP permite conectarnos con la base creada del CDR de MySQL.

Permite acceder a todos los campos de la base para posteriormente realizar las respectivas consultas del registro de llamadas. Cabe mencionar que su conexión no es con el archivo CDR de Asterisk, sino con la base de MySQL. Su conexión previa, ya fue realizada con la modificación del archivo `crd_mysql.conf`.

A continuación el código de la clase conexión de PHP:

```
<?php
class Conexion{
    var $host;
    var $user;
    var $password;
    var $database;
    var $error;

    function Conexion($host="", $user="", $password="",
$database="")
    {
        $this->host = "localhost";
        $this->user = "root";
        $this->password = "";
        $this->database = "asterisk";
    }

    function conectar()
    {
        $conexionID = @mysql_connect($this->host,$this-
>user,$this->password);
        if ($conexionID)
        {
            if (@mysql_select_db($this-
>database,$conexionID))
            {
                return $conexionID;
            }
            else
            {
                $this->error = @mysql_errno().":
".@mysql_error();

                echo $this->error;
                return 0;
            }
        }
        $this->error = @mysql_errno().": ".@mysql_error();
        echo $this->error;
        return 0;
    }
}
}}?>
```



Registro de detalle de llamadas

Bienvenido, luis



AÑO Mes : Agosto Todos
 FECHA Desde : Hasta :
DESTINO Exacto Empezar con
ORIGEN Exacto Empezar con
CANAL Elija el canal
ESTADO Elija el estado
 DURACIÓN Llamada con mayor duración Llamada con menor duración

[Logout](#)

Soluciones 2009
Guayaquil - Ecuador

Fig. 4.3 Módulo Principal del CDR



Registro de detalle de llamadas

[Ver Reporte](#)

Listado de llamadas desde 2009/01/01 hasta 2009/12/31

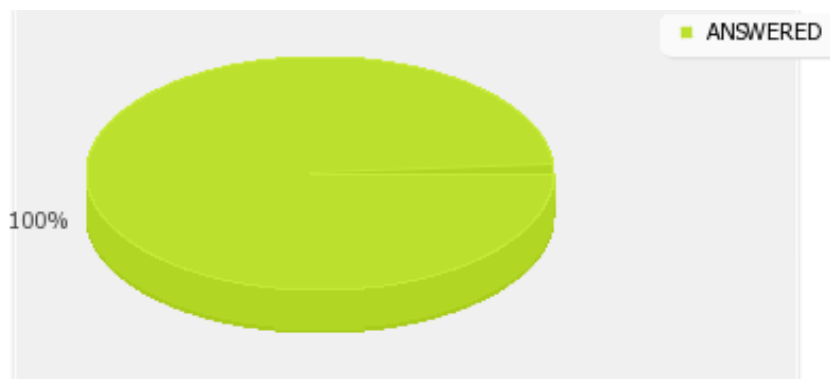
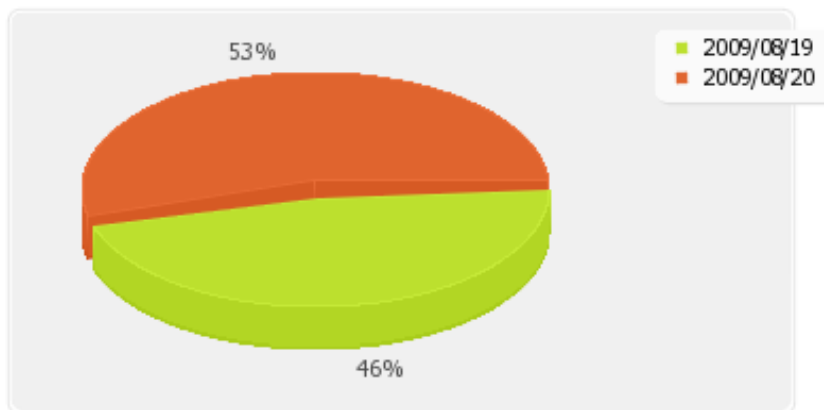
No.	Fecha	Canal	Origen	Destino	Duración	Estado
1	2009-08-19 16:51:26	SIP	1003	1002	00:00:11	CONTESTADO
2	2009-08-19 16:52:16	SIP	1002	1003	00:00:06	CONTESTADO
3	2009-08-19 17:01:28	SIP	1002	1003	00:02:06	CONTESTADO
4	2009-08-19 17:07:36	SIP	1003	1002	00:00:05	CONTESTADO
5	2009-08-19 17:19:10	SIP	1002	1003	00:05:34	CONTESTADO
6	2009-08-19 17:24:52	SIP	1003	1002	00:02:19	CONTESTADO
7	2009-08-20 16:41:36	SIP	1002	1003	00:00:14	CONTESTADO
8	2009-08-20 16:42:36	SIP	1003	1002	00:00:18	CONTESTADO
9	2009-08-20 16:45:46	SIP	1002	1003	00:00:10	CONTESTADO
10	2009-08-20 16:46:10	SIP	1003	1002	00:00:09	CONTESTADO
11	2009-08-20 16:46:27	SIP	1003	1002	00:00:10	CONTESTADO
12	2009-08-20 16:47:56	SIP	1003	1002	00:00:10	CONTESTADO
13	2009-08-20 16:49:36	SIP	1002	1003	00:00:10	NO CONTESTADO
14	2009-08-20 16:59:23	SIP	1002	1003	00:00:12	CONTESTADO
15	2009-09-20 17:07:36	SIP	1002	1003	00:00:10	NO CONTESTADO

Fig.4.4 Reporte de los Campos del CDR

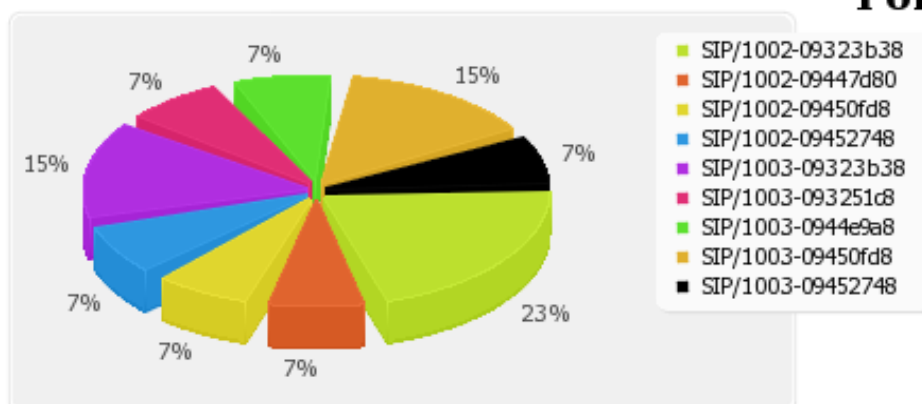
Graficos estadisticos

Desde 2009/08/01 hasta 2009/08/31

Por fecha:



Por estado



Por canal

Fig. 4.5 Gráficos Estadísticos del CDR

CONCLUSIONES

Al término de este trabajo se puede concluir:

1. Este trabajo ha permitido implementar el módulo para la elaboración de reportes del CDR de Asterisk.
2. Se ha logrado obtener una base de datos del CDR, en la cual podemos almacenar la información generada por el registro de llamadas para su presentación final de una manera ágil y útil al usuario.
3. Haber considerado a XAMPP para nuestro proyecto Web de desarrollo. Al ser la distribución independiente, asegura que todos sus servidores funcionen con la misma distribución de versiones de lámpara de pila. XAMPP facilita el uso de la lámpara de la producción o de la plataforma de desarrollo.
4. Se puede considerar de gran utilidad para el uso de empresas o instituciones que tengan instalado un PBX, ya que cumple con los requerimientos básicos para el análisis de la información.

5. Por ser opensource, permite futuras configuraciones y actualizaciones en su código fuente, logrando una mejor efectividad y alcance acorde a las necesidades de su entorno.

6. Permite además visualizar la información inclusive gráficamente.

RECOMENDACIONES

Al término de este trabajo se puede recomendar lo siguiente:

1. Por el hecho de ser opensource no garantiza el soporte del mismo, lo cual hay que tener en cuenta a nivel empresarial.
2. Tener en cuenta los requerimientos de asterisk, tales como los paquetes Addons, que contienen entre ellos MySQL, útil para el manejo de la base de datos.
3. Establecer correctamente los parámetros de la configuración del archivo cdr de asterisk, para que pueda existir la conexión con la base de datos.
4. Configurar adecuadamente los teléfonos softphones, verificando su aplicación en el protocolo adecuado, útiles para las pruebas de recepción y emisión de llamadas.
5. Aplicar una política de seguridad para el ingreso a la aplicación y su correcta administración.

BIBLIOGRAFÍA

1. Meggelen J, Smith J, y Madsen L, El Futuro de la Telefonía, O'Reilly Media, Inc. 2005.
2. Voip-Info.Org, http://www.voip-info.org/wiki/view/Asterisk+config+cdr_mysql.conf, 2009.
3. Slide Share, Introducción a Asterisk, <http://www.slideshare.net/saghul/introduccion-a-asterisk-297883>, 2009.
4. Asterisk, <http://www.asterisk.org/>, 2009.
5. Apache Friends, XAMPP, <http://www.apachefriends.org/es/xampp.html>, 2009