

# Diseño e Implementación de un Sistema de Gestión Telefónica Automática para Negocios Hoteleros.

Escobar Lenin, Garnica Freddy, Astudillo Gabriel Ing.  
Facultad de Ingeniería Eléctrica y Computación.  
Escuela Superior Politécnica del Litoral (ESPOL)  
Campus Gustavo Galindo, Km 30.5 vía Perimetral  
Apartado 09-01-5863. Guayaquil-Ecuador  
{ leniesco, frejagar , gastudil }@espol.edu.ec

## Resumen

*Este proyecto consistió en la implementación de una pequeña central telefónica basada en Asterisk capaz de gestionar a través de llamadas telefónicas las actividades relacionadas al sector hotelero. El cual permitió hacer reservaciones telefónicas, programación de despertadores a las habitaciones y consultar el estado del clima en tiempo real. Con el desarrollo de este proyecto se buscó administrar de una manera más eficiente los medios tecnológicos con los que consta el sector hotelero. Con la realización de este proyecto se buscó alcanzar los siguientes objetivos: gestionar de manera ágil las reservaciones, control sobre el flujo de llamadas, brindar un servicio automatizado de reservas, eliminando el costo un operador, brindar servicios diferentes como son despertadores y consulta del clima.*

**Palabras Claves:** Central Telefónica, Asterisk.

## Abstract

*This project involved the implementation of a small Asterisk-based PBX can handle phone calls through activities related to the hotel sector. Which allowed telephone reservations, scheduling of alarm to the rooms and check the status of real-time weather. With the development of this project sought to manage more efficiently the technological means with which has the hotel industry. With the completion of this project sought to achieve the following objectives: to manage reservations agile, control over the flow of calls, providing an automated reservation, eliminating the cost an operator to provide different services such as alarm clocks and weather query .*

**Keywords:** Telephone exchange, Asterisk.

## 1. Introducción

En el mundo actual el constante movimiento de las personas entre puntos geográficamente lejanos hace la necesidad del uso de hoteles para poder hospedarse.

Estos hoteles necesitan sistemas autónomos cada vez mejores para brindar un servicio de calidad a sus usuarios desde el momento de la reserva hasta cuando se realiza el registro de salida del mismo.

Uno de los sistemas a mejorarse es el sistema de reservas telefónicas mediante la implementación de centralitas telefónicas VoIP.

No obstante, un gran obstáculo para la implementación de estas soluciones ha sido su costo económico ya que la mayoría de estas son propietarias.

La centralita telefónica Open Source Asterisk, proporciona un método eficaz para combatir problemas de costos de implementación a la vez que nos permite un control del flujo de llamadas.

Además posee una alta escalabilidad, ya que puede ofrecer servicios para pequeñas empresas con pocos usuarios, hasta empresas que tengan diferentes sedes.

Asterisk incorpora la mayoría de los estándares telefónicos, lo que le permite conectarse sin ningún problema a la red pública de telefonía y se complementa con los diferentes lenguajes de programación (PHP, C, Pascal) para realizar tareas de más alta complejidad.

## 2. Objetivos

### 2.1. Objetivo Principal

Implementar un sistema que permita gestionar a través de llamadas telefónicas las actividades relacionadas al sector hotelero.

### 2.2. Objetivos Específicos

- Crear un sistema de reservas telefónicas automatizado para usuarios registrados previamente en la base de datos del hotel.

- Establecer un sistema que pueda realizar la programación de alarmas en cada una de las habitaciones del hotel.
- Optimizar los servicios que puedan brindarse a través de un medio telefónico.

### 3. Metodología

Se va a instalar Asterisk sobre un servidor el cual tendrá como sistema operativo CENTOS (Linux), también lo utilizaremos como servidor web para el cual se instalará Apache y PHP los cuales harán intercambio de información con una base de datos que estará corriendo en MySQL.

El servidor también contará con una tarjeta con puertos FXS y FXO para poder recibir y conmutar las llamadas que provengan desde la red PSTN.

### 4. Asterisk

Asterisk es un programa de software libre (bajo licencia GPL) que proporciona funcionalidades de una central telefónica (PBX). Como cualquier PBX, se puede conectar a un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una RDSI tanto básicos como primarios como se menciona en [1].

Asterisk incluye muchas características anteriormente sólo disponibles en costosos sistemas propietarios PBX como buzón de voz, conferencias, IVR, distribución automática de llamadas, y otras muchas más. Los usuarios pueden crear nuevas funcionalidades escribiendo un dialplan en el lenguaje de script de Asterisk o añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación reconocido por Linux.

Para conectar teléfonos estándar analógicos son necesarias tarjetas electrónicas telefónicas FXS o FXO fabricadas por Digium u otros proveedores, ya que para conectar el servidor a una línea externa no basta con un simple módem. La conexión entre el servidor y las líneas analógicas o digitales están de acuerdo al diseño de la figura 1.

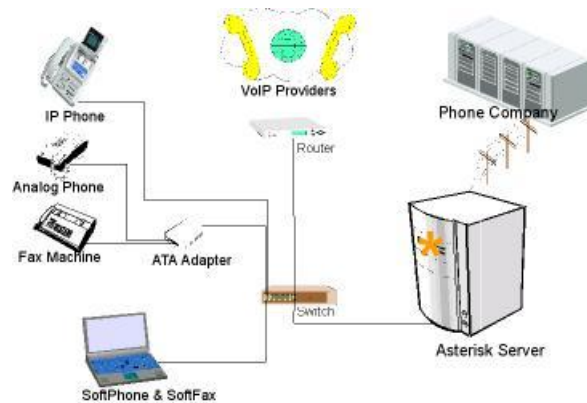


FIGURA 1. Esquema conceptual de una red basada en Asterisk.

Una de las características más interesantes de Asterisk es que reconoce muchos protocolos VoIP como pueden ser SIP, H.323, IAX y MGCP. Asterisk puede interoperar con terminales IP actuando como un registrador y como puerta de enlace entre ambos.

Asterisk puede integrarse como un puente transparente hacia la tecnología Voz sobre IP sin necesidad de modificar o actuar en la infraestructura telefónica ya desplegada y en producción.

Gracias a su condición de software libre, Asterisk está al alcance de pequeñas, medianas y grandes empresas, ya que su implementación es de muy bajo costo con respecto a las PBX's propietarias, ofreciendo además una gama de funcionalidades que se ajustan a los requerimientos de cualquier empresa.

Cabe recalcar que Asterisk podrá ajustarse a las redes telefónicas públicas a través de unas tarjetas que permiten la comunicación entre estas dos tecnologías como se ilustra en la figura 2. Las tarjetas que se van a utilizar son las FXS/FXO, que son muy accesibles para cualquier tipo de empresa de acuerdo a los requerimientos necesarios. En el mercado existen diferentes tipos de tarjetas que van a ir variando de acuerdo a los servicios que se vayan a brindar.

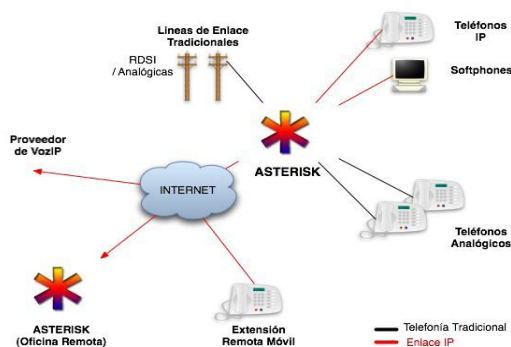


Figura 2. Teléfono IP GPX2000

## 5. Protocolo SIP

SIP es un protocolo desarrollado por el grupo de trabajo MMUSIC del IETF con la intención de ser el estándar para la iniciación, modificación y finalización de sesiones interactivas de usuario donde intervienen elementos multimedia como el video, voz, mensajería instantánea, juegos en línea y realidad virtual como se menciona en [3].

La sintaxis de sus operaciones se asemeja a las de HTTP y SMTP, los protocolos utilizados en los servicios de páginas Web y de distribución de e-mails respectivamente. Esta similitud es natural ya que SIP fue diseñado para que la telefonía se vuelva un servicio más en Internet.

### 5.1 Relación Entre SIP y Asterisk

El módulo de canal SIP de Asterisk permite comunicarse a través de VOIP con SIP y teléfonos de los intercambios. Asterisk es capaz de actuar como

- Un cliente SIP: Esto significa que los registros de Asterisk como un cliente a otro servidor SIP y recibe y realiza llamadas a este servidor. Las llamadas entrantes se encaminan a una extensión de Asterisk.

- Un servidor SIP: Asterisk puede ser configurado para que los clientes SIP (teléfonos, los clientes de software) el registro en el servidor Asterisk y establecer sesiones con el servidor SIP, es decir, las llamadas y responde a las llamadas entrantes. Dicho esto, Asterisk no es una completa función de servidor SIP como SIP Express Router u OpenSER. Si usted va a tener miles de teléfonos SIP, usted debe utilizar SER o OpenSER y desviar llamadas a Asterisk de voz o el acceso PSTN.

- Una puerta de enlace SIP: Actos de Asterisk como puerta de enlace de los medios de comunicación entre la SIP, IAX, MGCP, H.323 y PSTN conexiones. A modo de ejemplo, un servidor de Asterisk se puede conectar a la RDSI para dar a su conectividad de clientes SIP a la red telefónica conmutada

### 5.2 Características principales de SIP

Aspectos importantes referentes al protocolo SIP:

- El control de llamadas es sin estado (stateless), y proporciona escalabilidad entre los dispositivos telefónicos y los servidores.

- SIP necesita menos ciclos de CPU para generar mensajes de señalización de forma que un servidor podrá manejar más transacciones.

- Una llamada SIP es independiente de la existencia de una conexión en la capa de transporte.

- SIP soporta autenticación de llamante y llamado mediante mecanismos HTTP.

- Autenticación, criptográfica y cifrado son soportados salto a salto por SSL/TSL pero SIP puede usar cualquier capa de transporte o cualquier mecanismo de seguridad de HTTP, como SSH o S-HTTP.

- Un proxy SIP puede controlar la señalización de la llamada y puede bifurcar a cualquier número de dispositivos simultáneamente.

## 6. Implementación

La finalidad de este proyecto se basa en brindar soluciones telefónicas a muy bajo costo, haciendo uso del software libre como herramienta primordial para la elaboración del mismo.

### 6.1. Introducción

La finalidad de este proyecto se basa en brindar soluciones telefónicas a muy bajo costo, haciendo uso del software libre como herramienta primordial para la elaboración del mismo.

Para esta solución utilizamos como software principal Asterisk, que permite implementar centrales telefónicas a pequeña y gran escala dando muy buenos resultados, sin necesidad de recurrir a otros tipos de software propietarios que demandarían una mayor inversión.

Asterisk aumenta notablemente sus alcances al trabajar conjuntamente con otras herramientas como son: MySQL, PHP entre otras. Las antes mencionadas son que ayudan a cumplir con todas las funcionalidades propuestas en el proyecto.

Las centrales telefónicas han sido reemplazadas por computadoras que conjuntamente con programas simulan su funcionamiento, dando pie a que este tipo de servicios sean cada vez más accesibles para las personas y así poder mejorar los servicios que ofrecen sus empresas.

### 6.2. Hardware utilizado

El proceso para la selección del hardware a utilizar para implementar las soluciones es muy delicado y en el cual se deben tener en cuenta muchos factores que pueden influir en el funcionamiento de la central telefónica.

Se debe precisar los tipos de servicios a brindar y los datos que estos van a procesar, y con esto definir las características del hardware apropiado.

### 6.3. Servidor

Se ha trabajado con tres servidores: un servidor web, un servidor de bases de datos y el servidor que funciona como central telefónica, todos estos instalados sobre una misma computadora. Cuyas características se detallan en la tabla I.

|            |                        |
|------------|------------------------|
| CPU        | AMD SEMPRON de 2.1 GHz |
| RAM        | 1 GB.                  |
| Disco Duro | 20 GB.                 |

**Tabla I.** Características del servidor

### 6.4. Teléfono IP

Para nuestras pruebas realizadas en el laboratorio utilizamos un teléfono IP de la marca GrandStream modelo GXP2000 el cual se muestra en la figura 1.



**Figura 2.** Teléfono IP GPX2000

## 6.5. Software

### 6.5.1. Servidor PBX

En la tabla II se detalla los componentes utilizados en nuestro servidor.

|                   |                      |
|-------------------|----------------------|
| Sistema Operativo | Centos 5.5           |
| Software PBX      | Asterisk versión 1.8 |

**Tabla II.** Características del servidor

### 6.5.2. Servidor Web

La tabla III muestra los detalles del servidor apache.

|                       |             |
|-----------------------|-------------|
| Sistema Operativo     | Centos 5.5  |
| Software Servidor Web | Apache2.2.3 |
| Add-On                | PHPmyadmin  |

**Tabla III.** Servidor Web

### 6.5.3. Base de datos

Se utilizó MySQL como servidor de base de datos el cual almacena los datos necesarios para el desarrollo del proyecto como se muestra en la tabla IV.

|                        |                     |
|------------------------|---------------------|
| Sistema Operativo      | Centos 5.5          |
| Motor de base de datos | MySql server 5.0.77 |

**Tabla IV.** Servidor Base de datos

### 6.5.4. Softphones

Los softphones son simuladores de teléfonos IP los cuales trabajan con la dirección IP de la máquina sobre la cual está instalado y nos han sido de mucha ayuda para nuestras pruebas ya que son aplicaciones que se las pueden descargar gratuitamente desde internet mientras que los teléfonos IP reales son relativamente costosos.

Para este caso utilizamos el softphone X-Lite.

## 6.6 Configuración de archivos principales

### 6.6.1 Configuración Del Archivo “Sip.Conf”

El sip.conf es el archivo de configuración de canal correspondiente al protocolo SIP (Protocolo de Inicio de Sesión/Session Initiation Protocol) el cual es el protocolo de señalización más popular en la actualidad como se menciona en [5]. Este archivo contiene todas las configuraciones acerca de los usuarios que van a utilizar el protocolo SIP y además van a estar registrados dentro de nuestra PBX, y se encuentra en el directorio “etc/asterisk/sip.conf”.

## 6.6.1 Configuración Del Archivo “Extensions.Conf”

El principal archivo de configuración del Asterisk es el extensions.conf. En él se plasma toda la lógica de funcionamiento del sistema mejor conocida como dialplan o plan de discado. En el extensions.conf se controlan todas las conexiones al Asterisk, tanto entrantes como salientes, por tanto en él definimos todo el comportamiento de nuestra central telefónica IP [6].

## 7. Pruebas

### 7.1 Introducción

Después de haber concluido satisfactoriamente con la instalación de Asterisk, nuestro motor de base de datos MySQL, el servidor web Apache y todas las librerías necesarias para el correcto funcionamiento de nuestra PBX, además de haber modificado nuestros archivos de configuración y habiendo creado nuestros propios scripts para la realización de las consultas nos disponemos a presentar detalladamente cada una de las pruebas realizadas.

### 7.1 Consulta del clima en tiempo real

Procederemos a mostrar el proceso para la consulta del clima.

Marcamos las extensión “0” desde nuestro softphone, luego del cual se reproduce un mensaje de bienvenida ofreciéndonos dos opciones, como se muestra en la figura 3.

```

root@localhost:~
File Edit View Terminal Tabs Help
root@localhost:~ x root@localhost:~ x root@localhost:~
localhost*CLI>
localhost*CLI>
localhost*CLI>
localhost*CLI>
localhost*CLI>
localhost*CLI>
== Using SIP RTP CoS mark 5
-- Executing [0@internos:1] Goto("SIP/fgarnik-00000001", "MenuHotel_Menu,1") in new stack
-- Goto (MenuHotel_Menu,1)
-- Executing [Menu@MenuHotel:1] Answer("SIP/fgarnik-00000001", "") in new stack
-- Executing [Menu@MenuHotel:2] Background("SIP/fgarnik-00000001", "bienvenida1") in new stack
-- <SIP/fgarnik-00000001> Playing 'bienvenida1.gsm' (Language 'es')

```

FIGURA 3. Ingreso al menú del hotel

Luego de esto al usuario se le permite elegir entre dos opciones, para esta prueba se ingresará la opción dos, la cual reproduce la temperatura actual junto con la temperatura máxima y mínima para ese mismo día.

Una vez ingresado el número dos, Asterisk ejecuta el archivo “Wer.agi” que contiene las instrucciones para poder conectarse a la página web y mediante los comandos de PHP poder obtener los datos de la temperatura. Los archivos de audio son reproducidos

uno a la vez junto con los valores obtenidos desde la web como se puede apreciar en la figura 4.

```

-----
-- Executing [2@MenuHotel:1] Goto("SIP/fgarnik-00000001", "internos,clima,1") in new stack
-- Goto (internos,clima,1)
-- Executing [clima@internos:1] Answer("SIP/fgarnik-00000001", "") in new stack
-- Executing [clima@internos:2] AGI("SIP/fgarnik-00000001", "Wer.agi") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/Wer.agi
-- Playing 'temperature' (escape_digits=) (sample_offset 0)
-- Playing 'currently' (escape_digits=) (sample_offset 0)
-- <SIP/fgarnik-00000001> Playing 'digits/27.ulaw' (language 'es')
-- Playing 'degrees' (escape_digits=) (sample_offset 0)
-- Playing 'celsius' (escape_digits=) (sample_offset 0)
-- Playing 'temperature' (escape_digits=) (sample_offset 0)
-- Playing 'minimum' (escape_digits=) (sample_offset 0)
-- <SIP/fgarnik-00000001> Playing 'digits/21.ulaw' (language 'es')
-- Playing 'degrees' (escape_digits=) (sample_offset 0)
-- Playing 'temperature' (escape_digits=) (sample_offset 0)
-- Playing 'maximum' (escape_digits=) (sample_offset 0)
-- <SIP/fgarnik-00000001> Playing 'digits/28.ulaw' (language 'es')
-- Playing 'degrees' (escape_digits=) (sample_offset 0)
-- Playing 'vm-goodbye' (escape_digits=) (sample_offset 0)
[Aug 19 13:52:05] ERROR[4926]: : write() returned error: Broken pip
-- <SIP/fgarnik-00000001>AGI Script Wer.agi completed, returning 0
-- Executing [clima@internos:3] Hangup("SIP/fgarnik-00000001", "") in new stack
== Spawn extension (internos, clima, 3) exited non-zero on 'SIP/fgarnik-00000001'
localhost*CLI> █

```

FIGURA 4. Ejecución del script “Wer.agi”.

Para comprobar los datos, se consulta instantáneamente los valores que se encuentran en la página web como se muestra en la figura 5, y se verifica que son los mismos que han sido reproducidos por Asterisk.

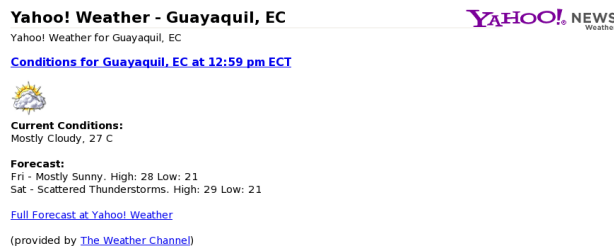


FIGURA 5. Condiciones del clima.

### 7.2 Ingreso de un nuevo cliente a la base de datos del hotel.

En esta parte se realiza el ingreso de un nuevo cliente en el hotel.

Una vez comprobado que la base de datos está corriendo se procede a ingresar al cliente, abrir el navegador e ingresar la página del hotel cuya dirección es <http://localhost/registro.php>, como resultado se obtiene lo que se muestra en la figura 6.



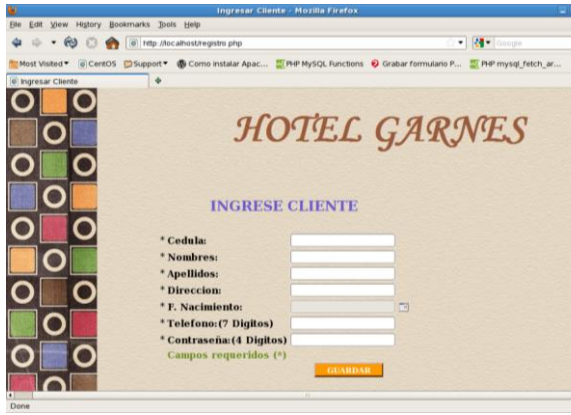


FIGURA 6. Página web del hotel.

Se ingresan los datos del nuevo cliente como se aprecia en la figura 7, en este caso todos los datos son obligatorios y todos los campos tienen las validaciones correspondientes para que el usuario no ingrese datos erróneos.

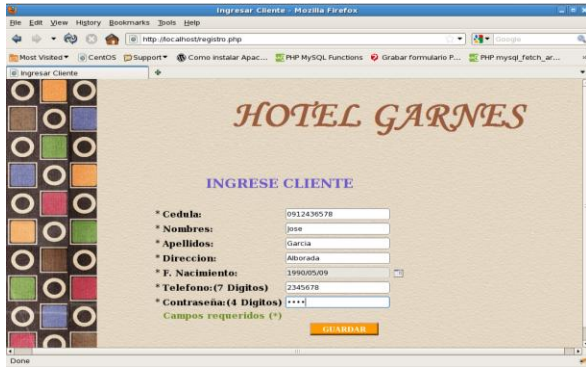


FIGURA 7. Ingreso de los datos.

Si el cliente ingresa algún dato que no es correcto, o datos incompletos el sistema no dejara que sean ingresados, caso contrario los datos son ingresados de forma satisfactoria como se muestra en la figura 8.

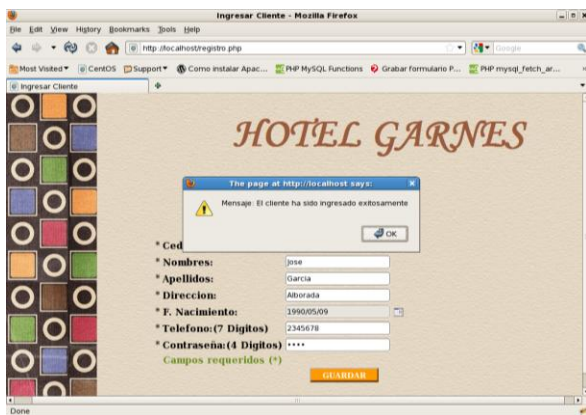


FIGURA 8. Datos ingresados correctamente.

Como se puede apreciar en la figura 9 el cliente no se encuentra registrado previamente en la base de datos.

| id_Cliente | Cedula     | Clave | Nombre        | Apellido        | F. Nacimiento | Telefono | Direccion    |
|------------|------------|-------|---------------|-----------------|---------------|----------|--------------|
| 1          | 0920603759 | 1234  | Freddy        | Garcia          | 1987-12-18    | 2498732  | Sur          |
| 2          | 0921974622 | 4321  | Lenin         | Escobar         | 1988-01-03    | 2498989  | Santa Monica |
| 12         | 0909842940 | 8581  | Gabriel       | Astudillo       | 1981-05-08    | 2397815  | Cda Fae      |
| 20         | 0902345222 | 1234  | Lilian Yamani | Mendoza Mendoza | 2011-08-02    | 2343434  | Sur          |
| 21         | 0909999976 | 4321  | Jose          | Escobar         | 2011-08-07    | 2343536  | Centro       |

FIGURA 9 Base de datos sin el cliente.

Después de haber hecho el ingreso satisfactorio de los datos, el cliente ya aparece en la base de datos de acuerdo a la figura 10, con toda la información descrita previamente.

```
mysql> select * from Cliente;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'select * from Cliente' at line 1
mysql> select * from Cliente;
```

| id_Cliente | Cedula     | Clave | Nombre        | Apellido        | F. Nacimiento | Telefono | Direccion    |
|------------|------------|-------|---------------|-----------------|---------------|----------|--------------|
| 1          | 0920603759 | 1234  | Freddy        | Garcia          | 1987-12-18    | 2498732  | Sur          |
| 2          | 0921974622 | 4321  | Lenin         | Escobar         | 1988-01-03    | 2498989  | Santa Monica |
| 12         | 0909842940 | 8581  | Gabriel       | Astudillo       | 1981-05-08    | 2397815  | Cda Fae      |
| 20         | 0902345222 | 1234  | Lilian Yamani | Mendoza Mendoza | 2011-08-02    | 2343434  | Sur          |
| 21         | 0909999976 | 4321  | Jose          | Escobar         | 2011-08-07    | 2343536  | Centro       |
| 22         | 0912436578 | 1234  | Jose          | Garcia          | 1990-05-09    | 2345678  | Alborada     |

FIGURA 10. Base de datos ya con el cliente.

### 7.3 Programación de despertador

En este punto se revisa el proceso para la programación de un despertador desde una extensión la cual se encuentra dentro del hotel.

Marcar la extensión 0 para escuchar el menú principal, luego escoger la opción 1 que es la opción para la programación de despertadores, se ingresa la hora deseada para el despertador y en ese momento el despertador ya se encuentra programado como lo muestra la figura 11.

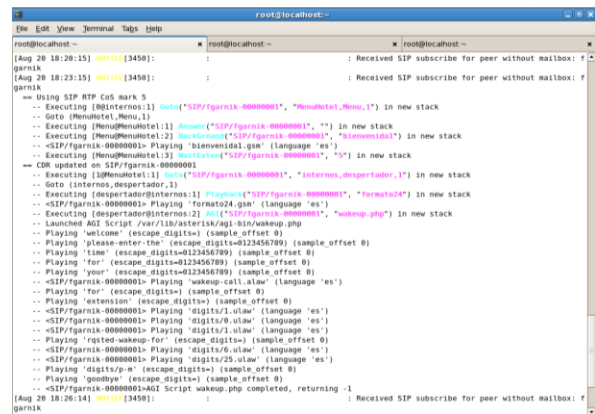


FIGURA 11. Ingreso del despertador para las 18:25.

Cuando sea la hora a la que se programó el despertador, el servidor genera una llamada hacia la extensión como se muestra en la figura 12a, y al alzar el auricular se reproduce una melodía similar al de un tono de alarma como se muestra en la figura 12b.

```
[Aug 20 18:26:34] NOTICE [3450]: : Received SIP subscribe for peer without mailbox: fgarnik
-- Attempting call on Local/180@internos for application MusicOnHold() (Retry 1)
-- Executing [180@internos:1] Dial("Local/180@internos-e214:2", "SIP/fgarnik") in new stack
-- Using SIP RTP CoS mark 5
-- Called fgarnik
-- SIP/fgarnik-00000002 is ringing
-- SIP/fgarnik-00000002 answered Local/180@internos-e214:2
-- Started music on hold, class 'default', on Local/180@internos-e214:1
-- Span extension (internos, 301, 1) exited non-zero on 'Local/180@internos-e214:2'
-- Stopped music on hold on SIP/fgarnik-00000002.
[Aug 20 18:27:18] NOTICE [4179]: : Call completed to Local/180@internos
localhost@cli>
```

FIGURA 12a. Consola de Asterisk, generación de la llamada



FIGURA 12b. Llamada entrante del despertador.

### 7.4 Programación de una reserva

En este punto se analiza la programación de una reserva para un cliente el cual ya debe estar previamente ingresado en la base de datos del hotel con su respectivo usuario y contraseña.

Debe llamar a la extensión indicada para realizar la reserva, luego ingresar los datos de usuario (cedula) y contraseña (ver figura 13), luego de ser validados los datos se le indica que ingrese el tipo de habitación que desea reservar 1 “sencilla”, 2 “doble” o 3 “triple”, después debe ingresar las fechas tanto de llegada al hotel como de salida del mismo, si existen habitaciones disponibles para esas fechas se le indica los datos ingresados por el cliente y el costo total de la reserva, si el cliente está de acuerdo presionará 1 caso contrario 2; Si el cliente elije la opción 1 la reservación se realiza.

```
-- Using SIP RTP CoS mark 5
-- Executing [555@internos:1] Playback("SIP/fgarnik-00000000", "bienvenida2") in new stack
-- <SIP/fgarnik-00000000> Playing 'bienvenida2.gsm' (language 'es')
-- Executing [555@internos:2] Read("SIP/fgarnik-00000000", "cedCliente") in new stack
-- User entered '0926035759'
-- Executing [555@internos:3] Playback("SIP/fgarnik-00000000", "contrasena") in new stack
-- <SIP/fgarnik-00000000> Playing 'contrasena.gsm' (language 'es')
-- Executing [555@internos:4] Read("SIP/fgarnik-00000000", "passcliente") in new stack
[Aug 20 18:17:15] NOTICE [3450]: : Received SIP subscribe for peer without mailbox: fgarnik
-- User entered '1234'
-- Executing [555@internos:5] AGI("SIP/fgarnik-00000000", "Consulta-1.agi,0926035759,3254") in new stack
-- Launched AGI Script: /var/lib/asterisk/agi-bin/Consulta-1.agi
```

FIGURA 13 Ingreso de usuario y contraseña.

## 8. Conclusiones

1. El desarrollo de este tipo de aplicaciones automatizadas está creciendo a un ritmo cada vez mayor, y el uso de herramientas de este tipo son de mucha utilidad en estos días, donde las empresas están destinadas a estar a la vanguardia de la tecnología, en nuestro caso al implementar un sistema de reservas telefónico que si bien es cierto, es básico a la hora de realizar la reserva pero con posteriores modificaciones llegaría a ser un software muy útil para cualquier tipo de hotel.
2. El uso del software ha sido muy beneficioso para nosotros, porque podemos demostrar que mediante el uso del mismo se pueden lograr aplicaciones muy novedosas sin necesidad de recurrir a software privativos que pueden alcanzar grandes valores monetarios, y además de eso, en el internet se puede encontrar un sinnúmero de tutoriales y foros en los cuales encontrar información acerca de Asterisk y sus aplicaciones que en otros países se han desarrollado con anterioridad.
3. Mediante el uso de lenguaje de programación PHP, y el motor de base de datos MySQL logramos interactuar entre nuestra PBX y los datos almacenados. Utilizando estas herramientas conseguimos la información con respecto a las condiciones actuales del clima, realizar nuevas reservas para clientes registrados previamente en nuestra base de datos y la programación automática de despertadores de una manera sencilla, con esto llegamos a la conclusión que es de gran importancia aprender un poco más acerca de estos lenguajes, ya que Asterisk se complementa con los mismos.
4. Pudimos darnos cuenta del alcance de Asterisk y así entender por qué esta aplicación ha tomado tanta fuerza y su uso va en aumento cada día más. Asterisk es una herramienta muy poderosa la cual se puede aprovechar conforme se conozca más de ella.
5. Se logró crear un sistema de reserva automatizado y de fácil acceso para todos los usuarios que ya se encuentran en los registros del hotel.
6. Se estableció un sistema de despertador capaz de realizar llamadas a todas las

extensiones del hotel de acuerdo a la hora ingresada por los diferentes usuarios.

7. Se optimizo el servicio de reservas, en su nivel más básico, al no necesitar de una persona que se encuentre tomando los datos del cliente.

## 9. Agradecimiento

Agradecemos al Ing. Gabriel Astudillo por haber sido nuestro director de tesis y estar atento a responder nuestras inquietudes para así poder resolver los inconvenientes y dudas que se presentaron durante el desarrollo del proyecto.

## 10. Referencias

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

[2] 3CX Ltda., Definición de FXS/FXO, <http://www.3cx.es/voip-sip/fxs-fxo.php>, revisado en Junio del 2011.

[3] Wikipedia, Definición del protocolo SIP, [http://es.wikipedia.org/wiki/Session\\_Initiation\\_Protocol](http://es.wikipedia.org/wiki/Session_Initiation_Protocol), revisado en Junio del 2011.

[4] Ramón Jiménez, Definición del archivo SIP.conf, <http://jimenezra.blogspot.com/2011/02/telefonía-ip-asterisk-sipconf.html>, revisado en Julio del 2011.

[5] Ramón Jiménez, Definición del archivo EXTENSIONS.conf, <http://jimenezra.blogspot.com/2011/02/telefonía-ip-asterisk-extensionsconf.html>, revisado en Agosto del 2011.

[6] VoIp-info.org, Descarga del archivo "Wakeup.php", <http://www.voip-info.org/wiki/view/Asterisk+tips+Wakeup+Call+PHP>, revisado en Junio del 2011.

[7] VoIp-info.org, Definiciones varias, <http://www.voip-info.org>, revisado en Agosto del 2011.