



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



“Diseño e Implementación de un Sistema Telefónico Interactivo que Permita Realizar Consultas de Calificaciones para la Academia CISCO – ESPOL”

Sandra Pineda Obando (1), Ángel Aguirre Sanabria (2), Gabriel Astudillo Brocel (3)

Facultad de Ingeniería en Electricidad y Computación (FIEC)

Escuela Superior Politécnica del Litoral (ESPOL)

Campus Gustavo Galindo, Km 30.5 vía Perimetral

Apartado 09-01-5863. Guayaquil, Ecuador

SAKAPINE@espol.edu.ec (1), aaguirre@fiec.espol.edu.ec (2)

Escuela Superior Politécnica del Litoral (ESPOL) (3), Ingeniero en Telecomunicaciones(3), gastudil@espol.edu.ec (3)

Resumen

El presente trabajo consistió en el diseño e implementación de un sistema IVR basado en Asterisk para los estudiantes de la Academia Cisco – ESPOL.

Con la finalidad que mediante la programación de un script se pueda acceder a una base de datos y obtener así la información deseada toda la interacción se realizara por AGI, el mismo que permite el funcionamiento de Asterisk con programas externos en diferentes lenguajes de programación.

Con este proyecto conseguimos que los estudiantes por medio de una llamada puedan acceder a sus notas de los diferentes cursos tomados en Cisco-ESPOL, las 24 horas al día, ingresando su número de cédula, además de información de posibles cursos a dictarse. Además de una página web creada para la parte administrativa que le facilite el ingreso de calificaciones y consultas en la base de datos.

Palabras Claves: Asterisk, IVR, AGI.

Abstract

This paper involved the design and implementation of an IVR system based on Asterisk for Cisco Academy students – ESPOL.

With the purpose that by scheduling a script to access a database and obtain the desired information the whole interaction is performed by AGI, the same that enables the operation of Asterisk with external programs in different programming languages.

With this project, students get through a call to access their notes of the different courses taken at Cisco-ESPOL, 24 hours a day, entering your ID number, as well as information of possible courses to be issued. In addition to a website created for the administrative part that facilitates the entry of qualifications and query the database

1. Introducción

Debido a la evolución de los sistemas telefónicos en los últimos años, tanto en el avance tecnológico así como de los servicios y aplicaciones telefónicas que aparecen con el objetivo de mejorar la interacción con el usuario.

Los IVR son aplicaciones de voz interactivas que aceptan como entrada tanto tonos marcados por el usuario como la voz del mismo, ofreciendo distintos tipos de respuesta según la programación del propio sistema.

Las tecnologías IVR han tenido un gran apogeo es así como en un inicio solo era usado en bancos y grandes empresas, llegando ahora hasta las tiendas que ofrecen interfaces de compra vía telefónica, así como en las universidades en donde se las utiliza para hacer consultas o brindar información.

2. Metodología

En este proyecto con el fin de realizar nuestros objetivos usaremos las siguientes herramientas; un servidor con el sistema operativo Centos sobre el cual realizaremos la instalación de Asterisk con sus respectivas librerías.

También haremos uso de la herramienta de gestión de base de datos MySQL en la cual crearemos una base de datos y realizaremos la conexión con Asterisk en el servidor por medio de AGI, el lenguaje que vamos a utilizar para programar AGI será PHP.

Finalmente usaremos el lenguaje de programación PHP para la creación de una página web dinámica que se conecte a la base de datos.

También usamos la librería de phpagi el cual posee varias funciones muy útiles a la hora de utilizar PHP como lenguaje de programación dentro de Asterisk.

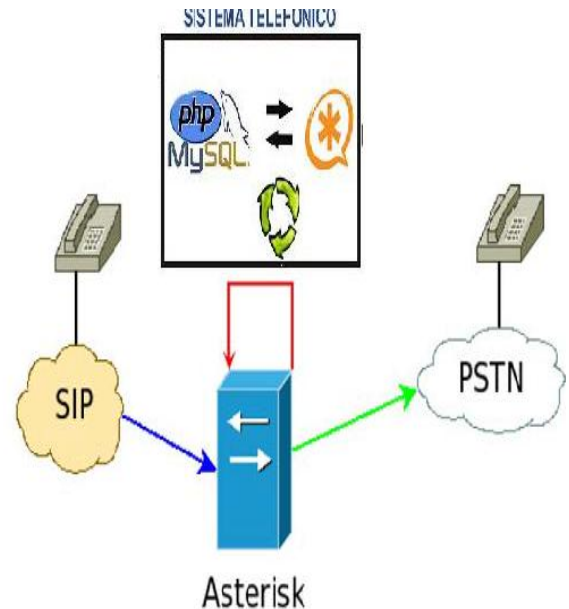


Figura 1. Interacción de nuestro Sistema a Implementar.

3. Asterisk y AGI

Asterisk es un Multiprotocol Modular PBX, cumple todas las funciones de una centralita, que integra las funcionalidades de telefonía clásica con nuevas capacidades derivadas de su flexible y potente arquitectura.

Funciona para múltiples sistemas operativos tales como Linux, BSD, MacOSX y Windows, donde las llamadas en el sistema disparan funciones a través de patrones de dígitos (mejor conocidos como extensiones), ofreciendo un completo control sobre el enrutamiento de las mismas con facilidad.

Proporciona todas las funcionalidades de las grandes centralitas propietarias (buzones de voz, IVR, etc.) y ofrece algunas posibilidades y servicios no disponibles en la mayoría de ellos (grabación de llamadas, extensiones remotas).

Además, por su arquitectura abierta y por su tecnología VoIP aporta importantes ventajas en costes y capacidades frente a los sistemas de telefonía convencionales.

Asterisk soporta gran variedad de protocolos de comunicaciones VoIP y es compatible con la mayor parte de fabricantes del hardware empleado para telefonía IP.

AGI provee una interfaz estándar para que programas externos puedan controlar el plan de marcación, permite a Asterisk enviar parámetros a un programa externo, ejecutar el programa y luego regresar a Asterisk e incluso recibir el resultado de dicho programa.

Al momento de la invocación de un programa AGI se le pasan ciertos parámetros y este a su vez responde con comandos AGI que son entendidos por Asterisk como su nombre mismo lo indica es una puerta de enlace hacia otras aplicaciones informáticas con las que se logra la integración entre la telefonía y la computación.

Los scripts AGI se comunican con Asterisk a través de STDIN, STDOUT y STDERR, esto quiere decir que deberían poderse ejecutar desde la misma línea de comandos.

4. Componentes para la implementación.

La elección del servidor es la parte más importante para poder desarrollar el presente proyecto, ya que será éste el que recopile y procese todos los datos que se generen en cada llamada.

Es por eso que hemos elegido un equipo con plataforma x86 con las siguientes características que vemos en las tablas 1 y 2:

Componentes de Hardware

Componente	Característica
Procesador	Intel Pentium Dual Core de 1.72GHz
Memoria RAM	1GB
Disco Duro	120 GB
Tarjeta de Red	10/100 Mbps

Tabla 1. Especificaciones Técnicas.

Componentes de software

Componente	Característica
Sistema Operativo	Linux
Distribución	Centos 5.5
Software IP PBX	Asterisk 1.6.1.9
Base de Datos	Mysql 5.0
Servidor Web	Apache 2.2.17

Tabla 2. Componentes de Software.

5. IVR

Permite atender llamadas telefónicas de manera automática y a consultar bases de datos. El IVR provee el acceso a la información que figura en sus bases de datos, posibilitando que se encuentre disponible para el usuario que realiza la llamada a un número de teléfono, el sistema de audiorespuesta contesta la llamada y le presenta al usuario una serie de acciones a realizar, esto se hace mediante mensajes (menús de opciones o reconocimiento de voz).

El usuario elige la opción a realizar introduciendo un número en el teclado del teléfono y navega por los diferentes menús hasta encontrar la información solicitada, todo esto es configurado en el plan de marcado que se explica más adelante.

Entre sus principales ventajas tenemos:

- Mejor servicio de atención al cliente.
- Reduce el porcentaje de abandono de llamadas.
- Reduce las colas de espera y tiempos de respuesta. Con los sistemas de IVR, las llamadas se pueden transferir a los departamentos apropiados con el proceso de selección del menú.
- Aumenta el número de llamadas atendidas.
- Acceso inmediato y sin esperas.
- Menor costo de atención por llamada.
- Reducción de costos de operación. Los sistemas integrados de respuesta de la voz

permiten que las compañías amplíen fácilmente sus horas de trabajo, sin la necesidad de operadores. Esto permite que los clientes llamen según su conveniencia y que puedan realizar múltiples tareas automáticamente.

6. Plan de Marcado

Asterisk utiliza el plan de marcado con un propósito general: completa el proceso de llamada en dos escenarios, es decir, tanto cuando el otro extremo se encuentre presente una persona o como cuando no haya nadie.

El plan de marcado de Asterisk incluye reglas que especifican qué hacer cuando:

- Una llamada se recibe en un canal particular o es realizada por un determinado usuario.
- Una llamada se recibe a una determinada hora del día, de la semana, etc.
- El extremo receptor de la llamada no contesta en un determinado intervalo de tiempo.
- La persona que realiza la llamada presiona ciertos dígitos tras escuchar un menú.
- La persona que realiza la llamada es dejada en espera o necesita entrar en una cola de espera, etc; durante la espera el usuario puede escuchar música o un mensaje; el usuario puede estar en espera indefinidamente o durante un tiempo limitado, tras el cual se llevarán a cabo otras acciones sobre la llamada.
- La persona que realiza la llamada establece una multiconferencia o transfiere la llamada telefónica a otra extensión.

El plan de marcado de Asterisk, es especificado en el archivo de configuración `extensions.conf`. En este archivo podemos distinguir tres secciones, cada una encabezada por una palabra entre corchetes que define el nombre de la sección.

La primera sección, llamada `[general]`, te permite establecer el valor de dos opciones usadas para controlar que el plan de marcado pueda o no ser modificado en tiempo real, desde la líneas de comando de Asterisk.

La segunda sección, llamada `[globals]`, se utiliza para definir variables cuyos valores podrán ser leídos y modificados en el plan de marcado, y que no modifican el comportamiento normal de Asterisk, sino simplemente almacenan un valor.

La tercera sección de este archivo de configuración, son los llamados contextos. Mientras que solamente pueden existir una sección llamada "general" y otra "globals", en el caso de los contextos pueden existir tantos como se quiera.

Un contexto, define diferentes modos de operación de Asterisk, se trata de un conjunto de extensiones que podrán ser ejecutadas según determinados criterios o a las que se le asocian un conjunto de permisos para realizar ciertas acciones, estas acciones se pueden también realizar en el AGI que detallamos su explicación a continuación.

7. Configuración de Asterisk

La configuración de Asterisk requerida para nuestro proyecto se realiza básicamente en tres archivos de la carpeta `/etc/asterisk/`: `sip.conf`, `extensions.conf` y en `usr/src/agi-bin`.

7.1 Archivo `sip.conf`

En este archivo se indica al servidor las extensiones locales que tendrán acceso a nuestro IVR.

A continuación detallamos la configuración del mismo:

```
[general]
context=default
srvlookup=yes
language=yes 49
disallow=all allow=ulaw,alaw,gsm,wav
```

```
[501]
callerid=softphone
type=friend
secret=1234
qualify=yes
nat=yes
host=dynamic
canreinvite=no
context=interno
```



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



7.2 Archivo extension.conf

Este archivo define el comportamiento de la central telefónica frente al manejo de llamadas, establece el plan de marcado (dialplan). A continuación detallamos la configuración del mismo.

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

```
[interno]
exten => _50X,1,Answer()
exten => _50X,n,Background(Bienvenida)
exten => _50X,n,Background(Pedido)
exten => _50X,n,WaitExten()
exten => 1,1,Goto(notas,s,1)
exten => 2,1,Goto(informacion,s,1)
exten => 3,1,Hangup()
exten => i,1,Background(invalido)
exten => i,n,Goto(interno,s,1)
exten => t,1,Playback(goodbye)
exten => t,n,Hangup()
exten => h,1,Hangup()
```

```
[notas]
exten => s,1,Background(cedula)
exten => s,2,Read(CI,,10)
exten => s,3,SayDigits(${CI})
exten => s,4,Playback(validar)
exten => s,5,WaitExten()
exten => 1,1,Agi(notas,agi,${CI})
exten => 2,1,Goto(s,1)
exten => t,1,Playback(goodbye)
exten => t,2,Hangup()
```

```
[informacion]
exten => s,1,Background(info)
exten => s,2,Background(sugerencia)
exten => 1,1,Goto(informacion,s,1)
exten => 2,1,Goto(notas,s,1)
exten => 3,1,Hangup()
exten => t,1,Playback(goodbye)
exten => t,2,Hangup()
```

```
[grabar]
exten => 100,1,Answer()
exten => 100,2,Record(informacion%d.wav)
exten => 100,3,Hangup()
```

7.3 Archivo AGI.

Este archivo define la interacción de Asterisk con programas externos como MySQL en nuestro caso y está escrito en PHP.

```
#!/usr/bin/php -q
<?php
ob_implicit_flush(false);
set_time_limit(50);
require "/var/lib/asterisk/agi-bin/phpagi-2.20/phpagi.php";

//Se abren los diferentes archivos
$in = fopen('php://stdin','r');
$stdlog = fopen('agi.log','w');

//Si debug es true, escribe en el archivo de log
$debug = true;
$agi = new AGI();

//Asignamos el valor de la cedula ingresada a $identificador
//write("GET VARIABLE CI");
//read();

$ident = $agi->get_variable('CI');
$identificador = $ident['data'];

//Hacemos la conexión a MySQL
$conectar =
mysql_connect("localhost","root","asterisk") or
die ("Error: El servidor no puede conectar con la
base de datos");
mysql_select_db("notas",$conectar) or
die(mysql_error());

//Seleccionamos datos de la tabla academia
$query = "SELECT *FROM academia WHERE
cedula = $identificador";
$resultado = mysql_query($query,$conectar);
$filas = mysql_num_fields($resultado);
$contentido = mysql_fetch_row($resultado);
$agi->text2wav("Nombres $contentido[1]");
$agi->text2wav("Apellidos $contentido[2]");
$campos = array(0 => 'cedula', 1 => 'nombres', 2
=> 'apellidos', 3 => 'ccna1', 4 => 'ccna2', 5 =>
'ccna3', 6 => 'ccna4', 7 => 'ccnp1', 8 => 'it1', 9
=> 'cableado estructurado', 10 => 'redes
inalambricas');
for($indice=3;$indice<$filas;$indice++){
do{
$agi->text2wav("$campos[$indice]");
$agi->text2wav("$contentido[$indice]");
$agi->text2wav("Si desea escuchar de nuevo
```



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



```
la nota presione 1");
$repetir = $sagi->get_data('beep',3000,1);
}while($repetir['result'] == 1);
}

$sagi->exec('Background', 'goodbye');
fclose($sin);
fclose($stdlog);

mysql_close($conectar);
$sagi->hangup();

?>
```

8. Conclusiones.

- 1) Gracias al continuo desarrollo y crecimiento de herramientas como PHP y MySQL permite encontrar la solución a problemas de filtrado de información dentro de proyectos de Voz sobre IP.
- 2) Con el presente proyecto pudimos observar como las tecnologías IVR se están destacando en el mercado debido a la optimización que brindan en el intercambio de información, reduciendo costos de operación y mantenimiento.
- 3) Hemos podido darnos cuenta de como con herramientas de software libre podemos construir sistemas confiables y de buen rendimiento sin tener que envidiar nada al software privativo y lo mejor sin costo alguno.
- 4) La modularización del sistema brinda la oportunidad de en cualquier momento incluir más aplicaciones para otorgar una mayor capacidad o incluir mejoras.
- 5) Debido a la clase PHPAGI desarrollada bajo PHP se pudo establecer una mejor interacción entre el script desarrollado bajo AGI con Asterisk.
- 6) En el sistema se ha logrado que haya una facilidad de uso para el usuario y que este tenga una respuesta inmediata.

9. Recomendaciones.

- 1) Se recomienda el uso del protocolo SIP, ya que es estandarizado por la IETF y es ampliamente implementado por todos los fabricantes de equipos y software. IAX2 está aun siendo estandarizado y es por ello que no se encuentra en muchos dispositivos existentes en el mercado, sin embargo también es viable el uso de IAX2 que consume menos ancho de banda y es más robusto.
- 2) Se recomienda la implementación de nuevas opciones en el IVR, para en caso de dudas la llamada se dirija a una secretaria de la Academia CISCO.
- 3) Implementar una nueva base de datos, donde se almacenen las llamadas que se han hecho durante el día, con el fin de conocer en que hora hay mayor flujo de tráfico y administrar mejor el ancho de banda.
- 4) Se recomienda que a partir de este proyecto se implemente, ya sea en otro proyecto de tesis, un IVR más avanzado usando VXML permitiendo así la creación de IVRs de nueva generación como son los NLU (Natural Languages Understanding), que permite que el IVR entienda la voz del usuario como comando valido para navegar entre los diferentes menús que lo conforman.

10 Bibliografía

- [1]. Luis Torres, “VOIP - Voz sobre IP (Voice over Internet Protocol)”, <http://www.monografias.com/trabajos3/voip/voip.shtml>, Diciembre del 2010.
- [2]. Diego Pasquarelli, “Introducción a la Telefonía IP (VoIP)”, <http://infovoiplc.blogspot.com/2009/02/caracteristicas-y-funcionalidad-del.html>, Octubre del 2007.
- [3]. Informática Hoy, “Entender VoIP (Voz sobre IP)”, <http://www.alegsa.com.ar/Notas/185.php>, Enero 2010.



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



- [4]. Leidy Contreras, “Características y Funcionalidades de la VoIP”, <http://infovoiplc.blogspot.com/2009/02/caracteristicas-y-funcionalidad-del.html>, Febrero del 2009.
- [5]. Soluctec Perú, “¿Qué es Asterisk?”, <http://www.soluctecperu.com/spsac/que-es-asterisk>, Enero 2010.]
- [6]. Gorka Gorrotxategi – Iñaki Baz, “Voz sobre IP y Asterisk”, <http://documentacion.irontec.com/cursoAsteriskVozIP-3-introduccionAsterisk.pdf>, Marzo del 2009.
- [7]. Tanke, “Interactive Voice Response” <http://aweps.industriaks3.com/html/ivr.html>, Marzo 2009.
- [8]. “IVR - Sistema de Respuesta Interactiva de voz”, http://www.classworx.com.ar/appdesign_ivr.html, Abril del 2009.
- [9]. “El Plan de Mercado”, <http://bibing.us.es/proyectos/abreproy/11379/fichero/memoria%252F4.pdf>, Febrero 2010.
- [10]. Edgar Landivar, “Comunicaciones Unificadas con Elastix” (2° edición), <https://www.camundanet.com/attachments/article/86/unificadas1.pdf>, 2008-2009.
- [11]. “Lista de Bases de datos populares con Servidor”, <http://www.entrebites.cl/foros/temas-generales/13922-lista-de-bases-de-datos-populares-con-servidor.html>, Agosto del 2008.
- [12]. Q-Forja, “Proyecto Hispavoces”, http://forja.guadalinux.org/frs/?group_id=21&release_id=120, Febrero 2011.
- [13]. “Descarga del programa Alien”, http://ftp.de.debian.org/debian/pool/main/a/alien/alien_8.72.tar.gz, Febrero 2011.