

ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

INFORME DE MATERIA DE GRADUACION

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

Previa a la obtención del TITULO de:

LICENCIADO EN REDES Y SISTEMAS OPERATIVOS

Presentada por:

**SANDRA KATHERINE PINEDA OBANDO
ANGEL PATRICIO AGUIRRE SANABRIA**

GUAYAQUIL – ECUADOR

AÑO
2011

AGRADECIMIENTO

A DIOS, quien nunca nos ha abandonado, por darnos fuerzas, valor y guiar nuestros pasos hacia el fin de esta jornada.

A nuestros padres, quienes nos han dado su apoyo tanto económico como moral, quienes nunca han dudado de nuestra capacidad y siempre estuvieron allí velando por nosotros, aconsejándonos y amándonos a pesar de haber cometido cualquier error.

A nuestros maestros, que con sus sabias enseñanzas y consejos nos han dado nuestras primeras herramientas para defendernos en mundo laboral.

DEDICATORIA

Agradezco a Dios, por permitirme la construcción de este sueño, a mis padres que con gran esfuerzo y dedicación me apoyaron para llegar a ser lo que hoy soy. A mis demás familiares y amigos que estuvieron allí, siempre con una palabra de aliento.

Sandra K. Pineda Obando

A Dios, por haberme guiado todos estos años, a mis padres y hermanos que han confiado en mí y me han apoyado.

Ángel P. Aguirre Sanabria

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Trabajo 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)

Sandra Katherine Pineda Obando

Ángel Patricio Aguirre Sanabria

TRIBUNAL DE SUSTENTACIÓN

Ing. Gabriel Astudillo

PROFESOR DE LA MATERIA DE GRADUACIÓN

Ing. Giuseppe Blacio

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

RESUMEN

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

Para el desarrollo de este proyecto utilizamos herramientas de software libre como lo son Asterisk, MySQL, PHP; permitiendo así que mediante la programación de un script en PHP se pueda acceder a una base de datos y obtener así la información deseada.

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.

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, ingresando su número de cédula, además de información de posibles cursos.

INDICE DE CONTENIDO

CAPITULO 1 ANTECEDENTES Y JUSTIFICACION	1
1.1 ANTECEDENTES	2
1.2 JUSTIFICACIÓN	3
1.3 DESCRIPCIÓN DEL PROYECTO	4
1.3.1 Objetivo General	4
1.3.2 Objetivos Específicos	4
1.3.3 Alcance	5
1.4 METODOLOGIA	6
1.5 PERFIL DE LA TESIS	7
CAPITULO 2 MARCO TEORICO	8
2.1 Voz sobre IP	9
2.1.1 Características principales	10
2.1.2 Ventajas de la VoIP.	10
2.1.2 Funcionalidad	11
2.2 ASTERISK	12
2.2.1 Funcionalidades de Asterisk	13
2.3 IVR (Interactive Voice Response)	17
2.3.1 ¿Cómo funciona un IVR?	19
2.3.2 Funcionalidades.	20
2.3.3 Principales Servicios de un IVR	20
2.3.4 Ventajas de un IVR	20
2.4 PLAN DE MERCADO.	21

2.5	AGI (Asterisk Gateway Interface)	24	
CAPITULO 3 HERRAMIENTAS PARA LA SOLUCION E			
IMPLEMENTACION			26
3.1	ESPECIFICACION TECNICAS	27	
3.2	HERRAMIENTAS BASADAS EN SOFTWARE.	29	
3.2.1	Softphone.....	29	
3.2.2	Php	30	
3.2.3	MySql.....	31	
3.3	INSTALACION	31	
3.3.1	Instalación de Asterisk	33	
3.3.2	Instalación de Apache, MySql Y PHP	35	
3.3.3	Instalación de voces en español.....	39	
3.3.4	Creación de la Base de Datos	42	
3.4	CONFIGURACION DE LOS ARCHIVOS DE ASTERISK.	46	
3.4.1	Configuración del archivo sip.conf.....	46	
CAPITULO 4 FUNCIONAMIENTO Y PRUEBAS			56
4.1	FUNCIONAMIENTO Y PRUEBAS	57	

INDICE DE FIGURAS

FIGURA 1: INTERACCIÓN DE NUESTRO SISTEMA A IMPLEMENTAR.	5
FIGURA 2: CONVERSIÓN DE LA SEÑAL ANALÓGICA A DIGITAL.	9
FIGURA 3: ASTERISK Y SUS FUNCIONALIDADES.....	12
FIGURA 4: SISTEMA IVR.....	17
FIGURA 5: DIAGRAMA DE INTERACCIÓN ENTRE ASTERISK, DIALPLAN Y AGI.	25
FIGURA 6: VISTA DEL SOFTPHONE X-LITE.	29
FIGURA 7: VISTA DE APACHE INSTALADO EXITOSAMENTE.....	36
FIGURA 8: VISTA DE PHP INSTALADO EXITOSAMENTE.....	37
FIGURA 9: VISTA DE LA INSTALACIÓN DE MySQL EXITOSAMENTE.....	38
FIGURA 10: VISTA DEL COMANDO #MYSQL -P.	42
FIGURA 11: SALIDA DE PANTALLA DE UN SHOW DATABASE.	43
FIGURA 12: MUESTRA DE PANTALLA DE UNA BASE DE DATOS CREADA.....	44
FIGURA 13: VISTA DE UNA BASE DE DATOS CON SUS CAMPOS LLENOS.	45
FIGURA 14: CONSOLA DE INICIO DE ASTERISK.....	57
FIGURA 15: CONFIGURACIÓN DEL SOFTPHONE X-LITE.....	58
FIGURA 16: REGISTRO DE SOFTPHONE X-LITE.....	59
FIGURA 17: SALIDA DEL COMANDO SIP SHOW PEERS.	59
FIGURA 18: VISTA DE UN PLAN DE MARCADO FUNCIONANDO.....	60

INDICE DE TABLAS

TABLA 1: ESPECIFICACIONES TÉCNICAS.	27
TABLA 2: COMPONENTES DE SOFTWARE.	27
TABLA 3: COSTOS DE IMPLEMENTAR VOIP CON SOFTWARE LIBRE.	28

INTRODUCCION

Desde principios la humanidad ha presentado cierta inquietud por cómo poder comunicarnos, lo que nos ha llevado a desarrollar una serie de tecnologías, las cuales a medida que el tiempo iba transcurriendo éstas iban mejorando. Es así como podemos destacar las señales de humo, que luego evolucionó y dio origen al envío de información de un lugar a otro a través de una persona.

Luego hubo un momento en que empezamos a desarrollar un tipo de tecnología llamada electrónica que facilitó el nacimiento del telégrafo; y debido a la imperante necesidad del hombre de buscar nuevas formas de comunicación se llegó a la invención de un aparato más tecnológico y eficiente que revolucionó al mundo entero, estamos hablando del teléfono cuyo inventor fue Antonio Meucci.

Con el nacimiento del teléfono se pone en funcionamiento la Red Telefónica Básica la cual es una red de comunicación diseñada primordialmente para la transmisión de voz. Se trata de la red telefónica clásica en la que los terminales telefónicos (teléfonos) se comunican con una central a través de un canal compartido por la señal del micrófono y del auricular.

El rápido crecimiento y la fuerte implantación de las redes IP, junto al desarrollo de técnicas avanzadas de digitalización de voz, priorización de tráfico así como el estudio de nuevos estándares que permitan la calidad de servicio en redes IP, han creado un entorno donde es posible transmitir la Voz sobre IP.

La Voz sobre IP (VoIP) está abaratando las comunicaciones internacionales y mejorando por tanto la comunicación entre proveedores y clientes. El ancho de banda creciente a nivel mundial y la optimización de los equipos de capa 2 y 3 hacen que el futuro de la Voz sobre IP sea muy prometedor.

CAPITULO 1

ANTECEDENTES Y JUSTIFICACION

1.1 ANTECEDENTES

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.

Es así como la Voz sobre IP ha alcanzado niveles de aceptación muy grandes gracias a la popularidad de Asterisk, el cual es un programa de software libre que proporciona las funcionalidades de una central telefónica. Entre las características de Asterisk tenemos buzón de voz, conferencias, IVR (Interactive Voice Response) y muchas más.

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.

En nuestro proyecto haremos el Diseño y la Implementación de un sistema IVR para realizar consultas de calificaciones en la Academia Cisco – ESPOL.

Con esto buscaremos que la interacción con los estudiantes de la Academia sea de una forma más automatizada.

1.2 JUSTIFICACIÓN

Debido a la gran necesidad de poder tener acceso a un servicio las 24 horas del día en un ambiente académico, donde se necesitan las calificaciones en un tiempo prudencial y que el estudiante no siempre tiene acceso a la web, es que se necesita usar otras alternativas que se ajusten a lo que necesita el usuario, como la herramienta con que la mayoría cuenta, el celular.

Esto en un entorno convencional de telefonía local se logra teniendo personal las 24 horas del día, además del alto costo de las llamadas si se hacen por medio del celular, pero usando un Software Libre e implementando el estándar de comunicación VoIP y un sistema IVR sería más económico para ambas partes.

1.3 DESCRIPCIÓN DEL PROYECTO

Para el desarrollo e implementación de nuestro proyecto nos hemos propuesto alcanzar los siguientes objetivos:

1.3.1 Objetivo General

- Implementar una solución de un Sistema telefónico interactivo que satisfaga las necesidades de consulta de calificaciones de los usuarios de la Academia Cisco – ESPOL que sea económicamente rentable.

1.3.2 Objetivos Específicos

- Determinar el software y la respectiva configuración que brinde los requerimientos para obtener la mejor compatibilidad, desempeño y resultados del proyecto.
- Demostrar la viabilidad técnica del uso de tecnología VoIP así como también del uso de software libre.
- Analizar los diversos protocolos de señalización y hardware necesarios para la implementación del sistema telefónico interactivo.
- Crear un asistente de llamadas (IVR), que interactúe con una base de datos MySql donde se guardaran los registros de los alumnos,

además de la creación de un formulario en PHP para facilitar su ingreso en la misma como se muestra en la Figura 1.

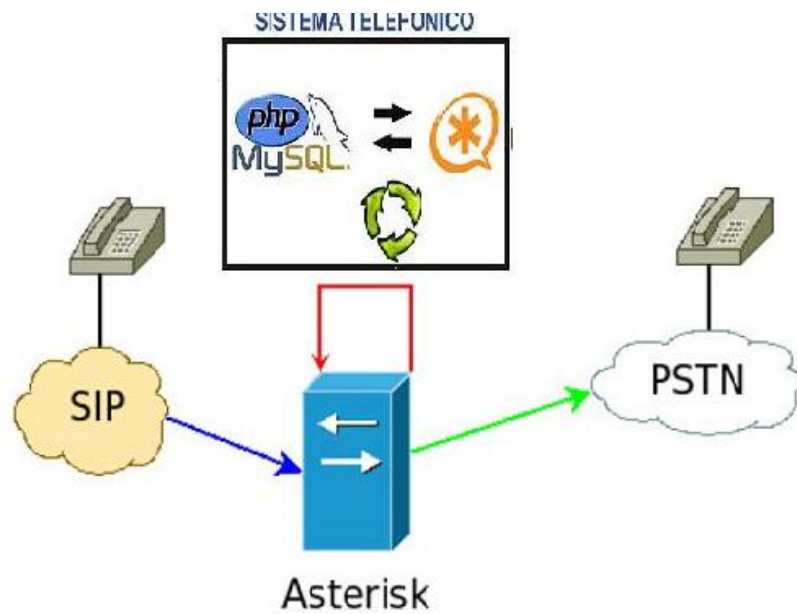


Figura 1: Interacción de nuestro Sistema a Implementar.

1.3.3 Alcance

- Permitir que las consultas de calificaciones por parte de los estudiantes de la Academia Cisco – ESPOL sea de una manera dinámica, es decir que se consulte directamente a la base de datos de la Academia por medio del uso del sistema telefónico.
- El sistema telefónico se hará con Asterisk porque funciona como una centralita, además de ser capaz de trabajar con casi todos los estándares de telefonía tradicional.

- Se usará un sistema IVR para que los estudiantes puedan acceder al servicio las 24 horas del día, ya que con esto no se necesitará operadores para atender las llamadas.
- El sistema almacenará estas calificaciones en un servidor, mediante una base de datos que será MySql, y serán ingresadas a la base de datos por medio de PHP porque es un lenguaje multiplataforma y completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.

1.4 METODOLOGIA

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 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.

1.5 PERFIL DE LA TESIS

El objetivo principal de esta tesis es desarrollar un sistema telefónico interactivo para la Academia Cisco – ESPOL el cual permita a sus estudiantes realizar consultas de sus calificaciones obtenidas en los cursos que hayan tomado en la Academia, para esto vamos a hacer uso de un sistema IVR.

En el capítulo 1 se destacan los Antecedentes y Justificación del porque de esta tesis, además se incluyen los objetivos que se tienen para el desarrollo de la misma así como la Metodología a implementarse.

En el capítulo 2 tenemos la base teórica respecto a lo que es Voz sobre IP, el software que vamos a usar, en este caso Asterisk, lo que son los IVR y el plan de marcado.

En el capítulo 3, se habla de su implementación, todo lo que se necesita tanto de hardware como de software para llevar a cabo este proyecto.

Posteriormente, en el capítulo 4, se harán las pruebas de funcionalidad ayudándonos de un softphone.

CAPITULO 2

MARCO TEORICO

2.1 Voz sobre IP

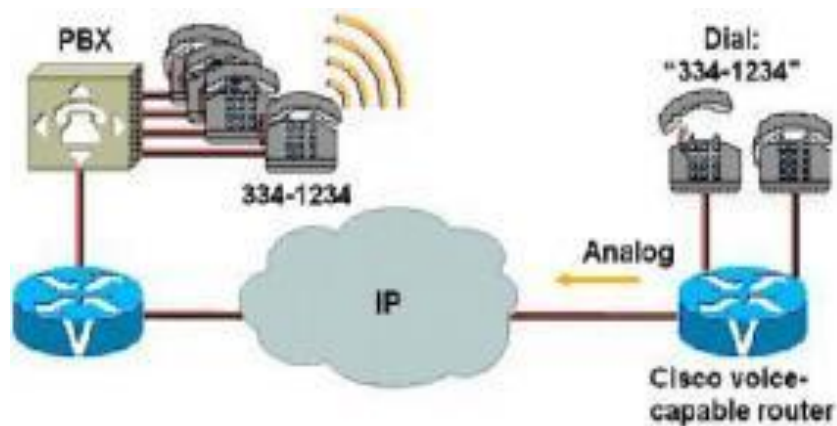


Figura 2: Conversión de la señal analógica a digital.

VoIP se puede definir como una aplicación de telefonía que puede ser habilitada a través de una red de datos de conmutación de paquetes por medio del Protocolo IP.

La ventaja de esta es la transmisión de voz como datos, ya que se mejora la eficiencia del ancho de banda para transmisión de voz en tiempo real.

Es la tecnología que posibilita el uso de redes IPs como medio de transmisión de voz que consiste en convertir los paquetes de voz, analógicos, en paquetes digitales y hacerlos transitar por internet como se muestra en la Figura 2. [1]

2.1.1 Características principales.

Permite controlar el tráfico de la red, por lo que se disminuyen las posibilidades de que se produzcan caídas importantes en el rendimiento.

Las redes soportadas en IP presentan las siguientes ventajas: [2]

- Es independiente del tipo de red física que lo soporta. Permite la integración con las grandes redes de IP actuales.
- Es independiente del hardware utilizado.
- Permite ser implementado tanto en software como en hardware, con la particularidad de que el hardware supondría eliminar el impacto inicial para el usuario común.
- Permite la integración de Video.

2.1.2 Ventajas de la VoIP.

- Se reducen los costos frente a los de la telefonía tradicional, especialmente cuando realizamos llamadas de larga distancia, en algunos casos las llamadas a teléfonos celulares suelen ser más caras que si se realizan por medio de la telefonía convencional y las llamadas entre las distintas delegaciones de la empresa saldrían gratis.
- Se pueden recibir llamadas estando en cualquier lugar geográfico del mundo mientras se tenga una conexión a Internet, el router y un teléfono convencional.

- Herencia de la red IP, que es la característica conocida por conmutación de paquetes. Esta característica permite que los paquetes busquen automáticamente el mejor camino disponible para que la comunicación sea establecida, aprovechando de manera más inteligente los recursos disponibles.[3][4]

2.1.2 Funcionalidad

VoIP puede facilitar tareas que serían más difíciles de realizar usando las redes telefónicas comunes: [4]

- Las llamadas telefónicas locales pueden ser automáticamente enrutadas a un teléfono VoIP, sin importar dónde se esté conectado a la red. Uno podría llevar consigo un teléfono VoIP en un viaje, y en cualquier sitio conectado a Internet, se podría recibir llamadas.
- Los agentes de Call center usando teléfonos VoIP pueden trabajar en cualquier lugar con conexión a Internet lo suficientemente rápida.
- Algunos paquetes de VoIP incluyen los servicios extra por los que PSTN normalmente cobra un cargo extra, o que no se encuentran disponibles en algunos países, como son las llamadas tripartitas, retorno de llamada, remarcación automática, o identificación de llamadas.

2.2 ASTERISK

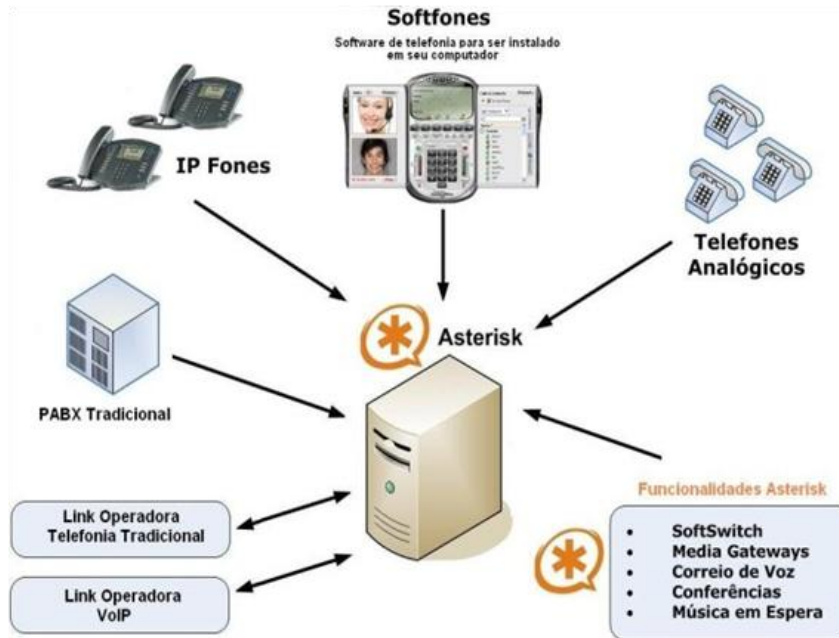


Figura 3: Asterisk y sus funcionalidades.

Asterisk es un OpenSource 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) como se muestra en la Figura 3.

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. [5]

2.2.1 Funcionalidades de Asterisk

- Como PBX soporta: buzones de voz, conferencias, transferencias, llamadas en espera, música en espera, IVR (secretaria virtual), grupos, colas, grabación de llamadas, Caller ID, registro de llamadas, intercomunicador, sistema de anuncios, envío de mensajes, envío y recepción de FAX, etc.
- Asterisk es capaz de trabajar con prácticamente todos los estándares de telefonía tradicional:
 - Líneas analógicas
 - Líneas digitales: E1, T1, accesos básicos.

- Soporta casi todos los protocolos de VoIP:
 - SIP
 - IAX/IAX2
 - MGCP
 - Cisco Skinny
- Conexión con líneas de telefonía tradicional, mediante interfaces tipo analógico (FXO) para líneas de teléfono fijo o bien móvil y RDSI (BRI o PRI).
- Soporte de extensiones analógicas, bien para terminales telefónicos analógicos, o equipos de fax.
- Soporte a múltiples códecs.
- Música en Espera basada en archivos MP3 y similar.
- Funciones básicas de usuario:
 - Transferencias (directa o consultiva)
 - Desvíos
 - Capturas (de grupo o de extensión)
 - Conferencia múltiple
 - Aparcamiento de llamadas (Call parking)
 - Llamada directa a extensión
 - Retrollamada – Callback (llamada automática cuando disponible).

- **Buzón de Voz:** sistema de contestador automático personalizado por usuario. Se integra con el sistema de directorio (LDAP) y con el email.
- **Sistema de Audioconferencias:** Sistema que permite la conexión remota de diferentes usuarios que quieren mantener una reunión virtual y suministra la correcta gestión y control de los usuarios que se incorporan a ella.
- **IVR Operadora Automática:** Sistema automatizado de respuesta que permite redirigir las llamadas entrantes en función a las opciones seleccionadas por el llamante.
- **Informes detallados de llamadas (CDR):** Detalle de llamadas realizadas/recibidas por extensión, para imputación de costes departamentales, por cliente o incluso para facturación.
- **ACD:** Sistema Automático de Distribución de Llamadas entrantes. Pensado para Centros de Llamadas, para atención comercial o soporte técnico.
- **CTI:** Integración con sistemas de gestión comercial o de atención al cliente.
- **IPCC (IP Contact Center):** Integración con sistemas avanzados de gestión de centros de llamadas, vía soluciones abiertas o propietarias.

- **LCR (Least Cost Routing):** Encaminamiento de llamadas por el proveedor VoIP más económico.
- **AGI (Asterisk Gateway Interface):** Interfaz para comunicar Asterisk con programas externos en diferentes lenguajes de programación.
- **AMI (Asterisk Management Interface):** Gestión y control remoto de Asterisk con esto se permite:
 - Crear llamadas
 - Monitorizar llamadas
 - Monitorizar canales y colas
 - Ejecutar comandos
- Configuración en base de datos: usuarios, extensiones, proveedores. [5][6]

2.3 IVR (Interactive Voice Response).

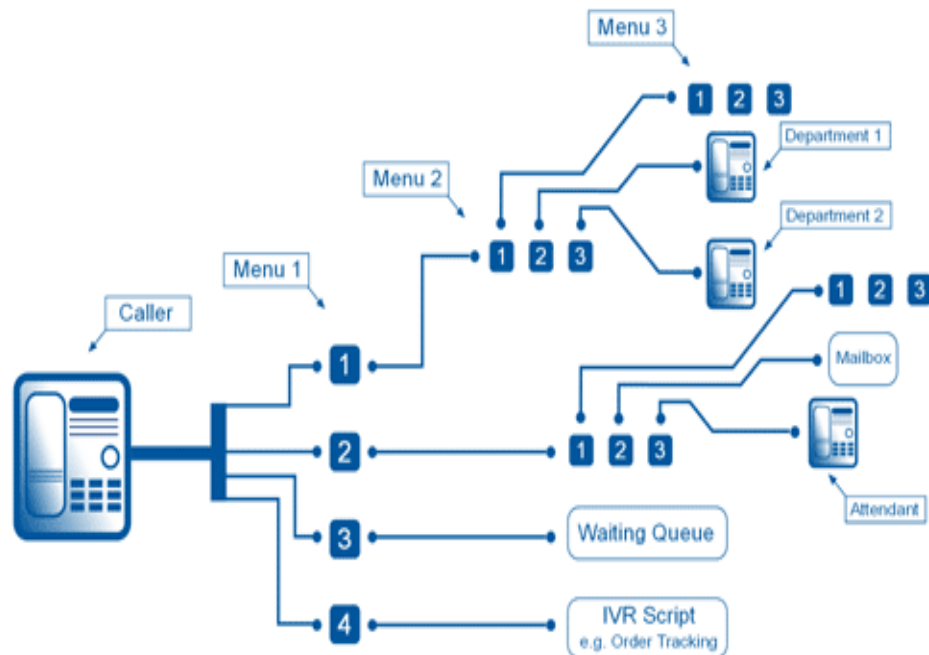


Figura 4: Sistema IVR.

IVR es la sigla para “Interactive Voice Response”, cuya traducción al español es “Respuesta de Voz Interactiva”, también conocido como VRU (Voice Response Unit).

IVR es una poderosa plataforma de desarrollo de aplicaciones telefónicas, que permite diseñar, integrar, implementar y administrar sistemas de respuesta interactiva de voz, utilizando un amigable lenguaje gráfico y en muy corto tiempo.

Es comúnmente implementado en empresas o entidades que reciben grandes cantidades de llamadas, a fin de reducir la necesidad de personal y los costos que el servicio ofrecido representen para dicha entidad.

IVR tiene capacidad para atender miles de llamadas al día, permitiendo a sus clientes recibir información, consultar y modificar bases de datos, vía telefónica y transferirse con una persona, cuando así lo requieran por medio de varios menús con múltiples opciones de marcado para satisfacer las necesidades del cliente como se muestra en la Figura 4.

Sus clientes podrán llamar las 24 horas del día, los 365 días del año, desde cualquier teléfono celular o fijo, y ser atendidos de inmediato; efectuar transacciones, realizar consultas, conferencias, levantar pedidos, recibir o enviar faxes, y muchas otras operaciones de manera automática y amigable.

Con ello, sus costos de atención de llamadas disminuirán drásticamente; así mismo los tiempos de espera y la calidad en el servicio, mejorarán notablemente.

2.3.1 ¿Cómo funciona un 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. [7]

Registra cada llamada recibida, con fecha, hora, duración y actividad de la llamada, en una base de datos abierta. Esta información permite generar reportes de tráfico de llamadas por hora, por día, por puerto, así como estadísticas de acceso a los distintos menús del sistema.

Los IVRs responden a tonos o comandos de voz, obtiene la información de su base de datos y la proporciona al usuario en forma de voz.

2.3.2 Funcionalidades.

Llamadas encaminadas para áreas internas especializadas.

- Previsibilidad: medición del volumen de llamadas y opciones más buscadas.
- Generación de informes.
- Sintetizador de voz (TTS - Text to Speech).
- Envío y recibo de fax.

2.3.3 Principales Servicios de un IVR

- Pago de cuentas.
- Identificación de clientes a través de documentos o protocolos.
- Divulgación de nuevos negocios, promociones, campañas de fidelidad y productos/ servicios.
- Divulgación de direcciones en función de la proximidad con una determinado dirección postal.
- Consulta de facturas, saldos e inversiones.

2.3.4 Ventajas de un IVR

- 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. [8]

2.4 PLAN DE MERCADO.

Todas las llamadas realizadas desde, hacia y a través de Asterisk, son manejadas por medios de circuitos lógicos de voz, que puede consistir en una línea telefónica a través de la cual sólo se establecerá una conexión o en una única conexión física donde cientos de comunicaciones comparten la conexión, como ocurre con los teléfonos SIP conectados a Asterisk a través de la interfaz Ethernet.

En cualquier escenario, a estos circuitos lógicos se les conoce como canales, y el de Asterisk es manejar su tráfico de voz acorde a un conjunto de reglas conocidas como plan de marcado (dial-plan).

El efecto que el plan de marcado tiene sobre una llamada, es llamado flujo o secuencia de la llamada.

Asterisk utiliza el plan de marcado con un propósito más 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. [9]

2.5 AGI (Asterisk Gateway Interface).

AGI (Asterisk Gateway Interface) es una manera de interactuar con Asterisk desde un programa de línea de comandos. Este programa puede ser escrito en lenguajes como Perl, PHP, C, Pascal, Bourne Shell y es invocado por Asterisk desde el dial plan.

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. [10]

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. Ver Figura 5.

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. Ver su configuración en el Anexo B.

- Lee desde STDIN para obtener información.
- Escribe en STDOUT para enviar información.
- Escribe en STDERR para enviar información de debugging.

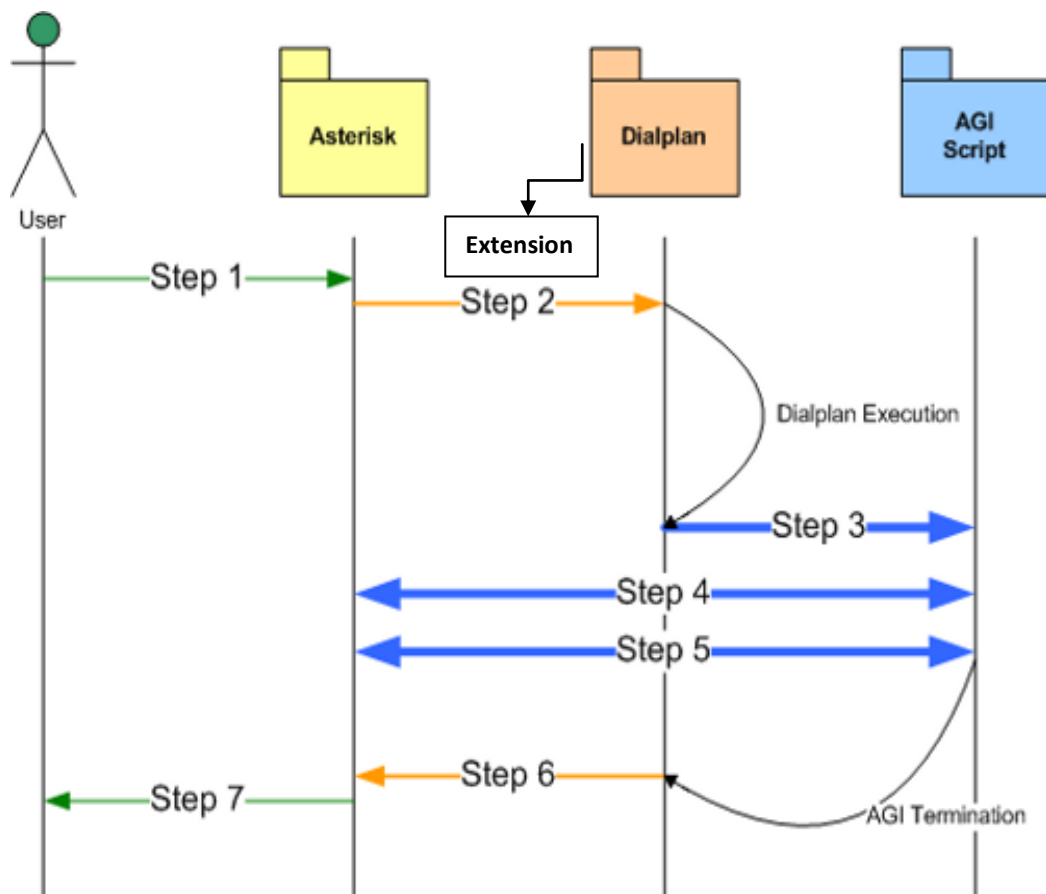


Figura 5: Diagrama de interacción entre Asterisk, Dialplan y AGI.

CAPITULO 3

HERRAMIENTAS PARA LA SOLUCION E IMPLEMENTACION

3.1 ESPECIFICACIONES TÉCNICAS

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:

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.

Costo de la implementación

A continuación en la Tabla 3 detallaremos el costo tanto de hardware como de software para la implementación del IVR bajo Asterisk:

Componente	Característica	Costo
PC	Pentium Dual Core de 1.72GHz; RAM 1GB; 120GB de Disco Duro; Tarjeta de Red 10/100	\$400
Sistema Operativo	Linux – Centos 5.5	Libre
Software IP PBX	Asterisk 1.6.1.9	Libre
Base de datos	Mysql 5.0	Libre
Servidor Web	Apache 2.2.17	Libre

Tabla 3: Costos de implementar VoIP con Software Libre.

En la tabla mostrada nos podemos dar cuenta que el implementar un Sistema IVR bajo software libre como Linux y Asterisk obtenemos un beneficio económico, ya que el único costo que tendríamos es el de el equipo que haría como servidor; también podemos utilizar herramientas de software libres como lo son MySql para Base de datos y Apache como servidor Web; además estas herramientas cuentan con amplio soporte y se nos hace más sencilla la tarea de buscar información para su correcta implementación y soporte posterior.

3.2 HERRAMIENTAS BASADAS EN SOFTWARE.

3.2.1 Softphone

Un softphone es un software que hace una simulación de teléfono convencional por computadora. Es decir permite usar la computadora para hacer llamadas a otros softphone o a otros teléfonos convencionales.

Es parte de un entorno Voz sobre IP y puede estar basado en el estándar SIP. El Softphone a utilizarse es el X-Lite como vemos en la Figura 5, el cual es desarrollado por Counterpath; entre las característica principal encontramos que está basado en el protocolo de señalización SIP lo que le permite trabajar



Figura 6: Vista del softphone X-Lite.

3.2.2 PHP

PHP es un lenguaje interpretado del lado del servidor, ejecutándose justo antes de que se envíe la página a través de internet al cliente. Las páginas que se ejecutan del lado del servidor pueden realizar acceso a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente.

Entre las principales ventajas que encontramos de PHP tenemos las siguientes:

- Es un lenguaje multiplataforma.
- Está completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Posee una amplia documentación en su página oficial.

Es por esto que se ha escogido PHP como lenguaje de programación debido a su potencia, versatilidad, robustez y modularidad; también porque contiene librerías que permiten interactuar con las funciones del API dentro de nuestro plan de mercado.

3.2.3 MySql

Es un lenguaje declarativo de acceso a base de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Debido a su continuo desarrollo y creciente popularidad MYSQL es un competidor cada vez más directo de Gigantes en la materia de Base de Datos. [11]

Sus principales ventajas por el cual lo usamos son:

- Mayor rendimiento.
- Integración perfecta con PHP.
- Sin límites en los tamaños de los registros.
- Mejor control de acceso de usuarios.

3.3 INSTALACION

Primero averiguamos la versión exacta de nuestro kernel con el comando:

```
#uname -a
```

Luego verificamos si tenemos las fuentes del kernel instaladas, para ello utilizamos el comando

```
#rpm -q kernel-devel
```

Instalación de las librerías

```
#yum install kernel-devel
```

```
#yum install kernel
```

```
#yum install bison
```

```
#yum install bison-devel
```

```
#yum install ncurses
```

```
#yum install ncurses-devel
```

```
#yum install zlib
```

```
#yum install zlib-devel
```

```
#yum install openssl
```

```
#yum install openssl-devel
```

```
#yum install gnutls-devel
```

```
#yum install gcc
```

```
#yum install gcc-c++
```

3.3.1 Instalación de Asterisk

Con el comando wget descargamos Asterisk en modo consola:

```
#wget http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-1.6.1.9.tar.gz
```

También descargamos los paquetes de Asterisk addons los cuales proveen drivers para conexión a servidores de MySQL y manejo de base de datos además de proveer de drivers para manejo de archivos en mp3

```
#wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-addons-1.6.1-current.tar.gz
```

También descargamos libpri que le da a nuestra central telefónica la posibilidad de realizar conexiones a líneas primarias de telefonía como las que existen en las PSTN.

```
#wget http://downloads.asterisk.org/pub/telephony/libpri/releases/libpri-1.4.10.2.tar.gz
```

Para realizar la descompresión de los archivos lo hacemos con el comando tar -xzvf + “nombre del paquete”

```
# tar -xzvf libpri-1.4.10.2.tar.gz
```

```
#tar -xzvf asterisk-1.6.1.9.tar.gz
```

```
#tar -xzvf asterisk-addons-1.6.1-current.tar.gz
```

Empezaremos con la instalación de el libpri, ingresamos a la carpeta con el comando cd luego creamos el ejecutable con los siguientes comandos dentro de la carpeta libpri

```
#make clean
```

```
#make
```

```
#make install
```

Ingresamos a la carpeta de asterisk-1.6.1.9

```
#!/configure
```

```
#make
```

```
#make install
```

```
#make samples
```

```
#make config
```

Luego ingresamos a la carpeta de asterisk addons con cd asterisk-addons-1.6.1, para instalar los addons escribimos los siguientes comandos

```
#!/configure
```

```
#make
```

```
#make install
```

```
#make samples
```

Una vez instalado podemos ver que se crea una carpeta llamada Asterisk dentro del directorio /etc donde se guardan los archivos de configuración.

Para iniciar la carga de nuestro servidor escribimos el siguiente comando:

```
#asterisk -vvvvvvvcg
```

Con esto ya tenemos listo nuestro servidor de Asterisk y con los comandos service podemos parar y arrancar el servicio de asterisk

```
#service asterisk stop
```

```
#service asterisk start
```

3.3.2 Instalación de Apache, MySql Y PHP

El primer paso es instalar MySql mediante el siguiente comando:

```
#yum install mysql mysql-server
```

Una vez instalado lo configuramos para que se inicie automáticamente con

```
#chkconfig --levels 235 mysqld on
```

Luego instalamos Apache con:

```
#yum install httpd
```

Una vez instalado lo configuramos para que inicie automáticamente con:

```
#chkconfig --levels 235 httpd on
```

Lo arrancamos con:

```
#/etc/init.d/httpd start
```

En este punto si ponemos la dirección del servidor se mostrara tal como se ve en la Figura 7:

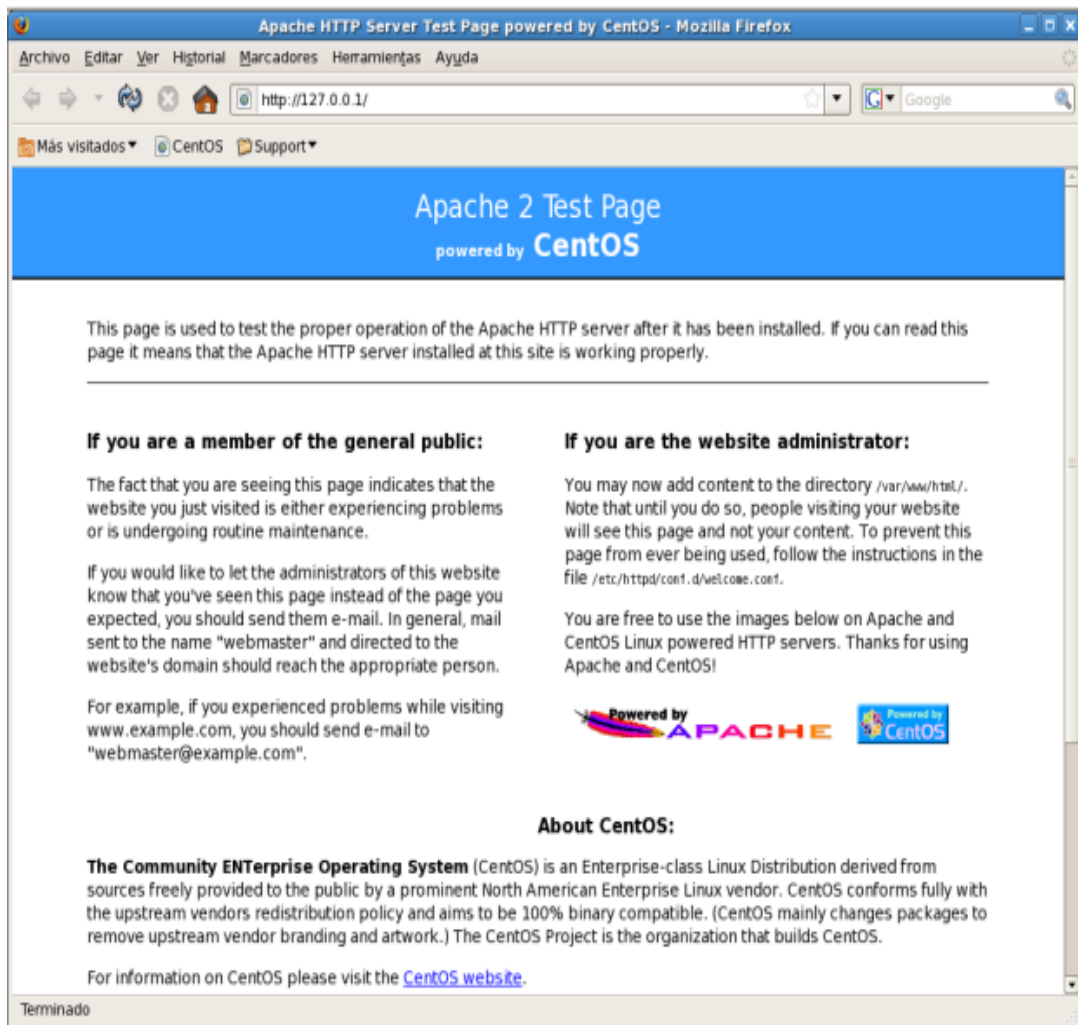


Figura 7: Vista de Apache instalado exitosamente.

Ahora instalamos PHP y lo vinculamos con Apache usando:

```
#yum install php
```


Reiniciamos apache usando:

```
#/etc/init.d/httpd start
```

Generamos un archivo para probar que PHP funciona correctamente usando:

```
#touch /var/www/html/info.php
```

```
#echo '<?php phpinfo(); ?>' > /var/www/html/info.php
```

Si navegamos la URL <http://direccion server/info.php> vemos lo mostrado en la Figura 8:

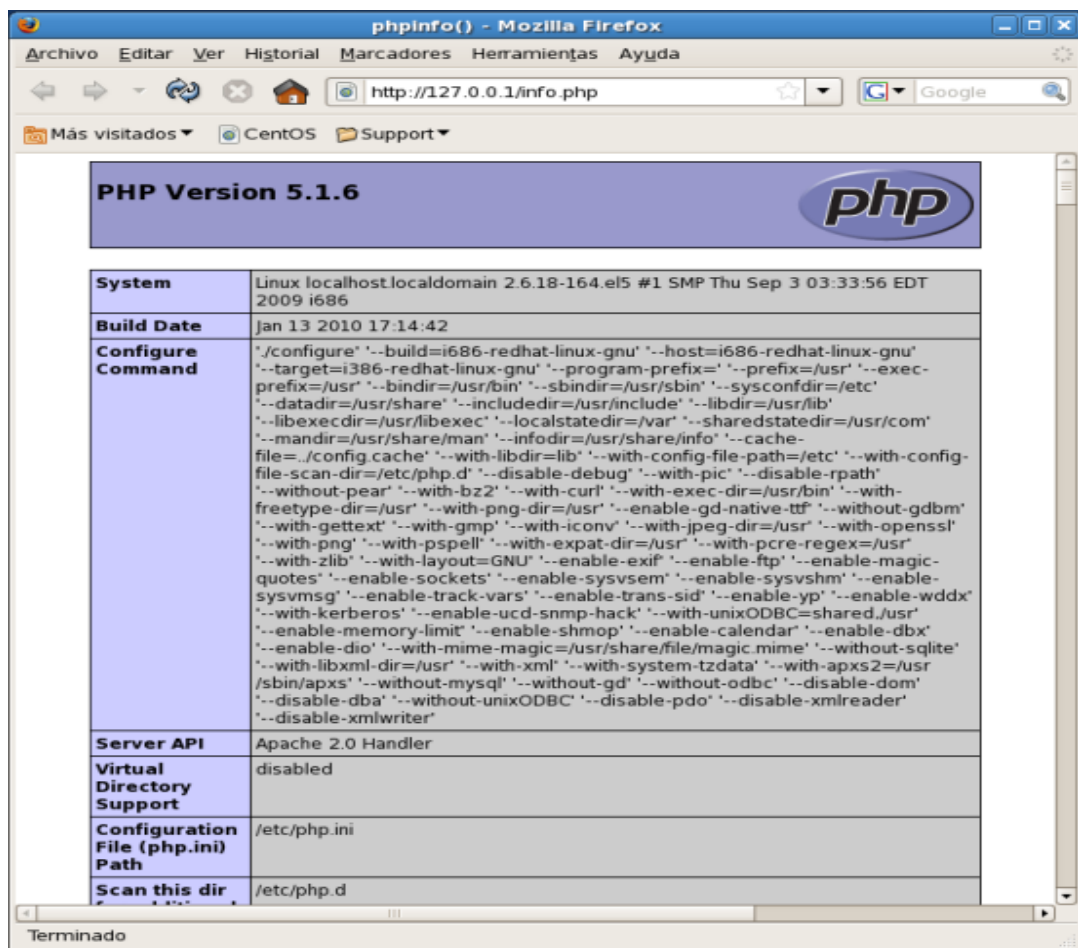


Figura 8: Vista de PHP instalado exitosamente.

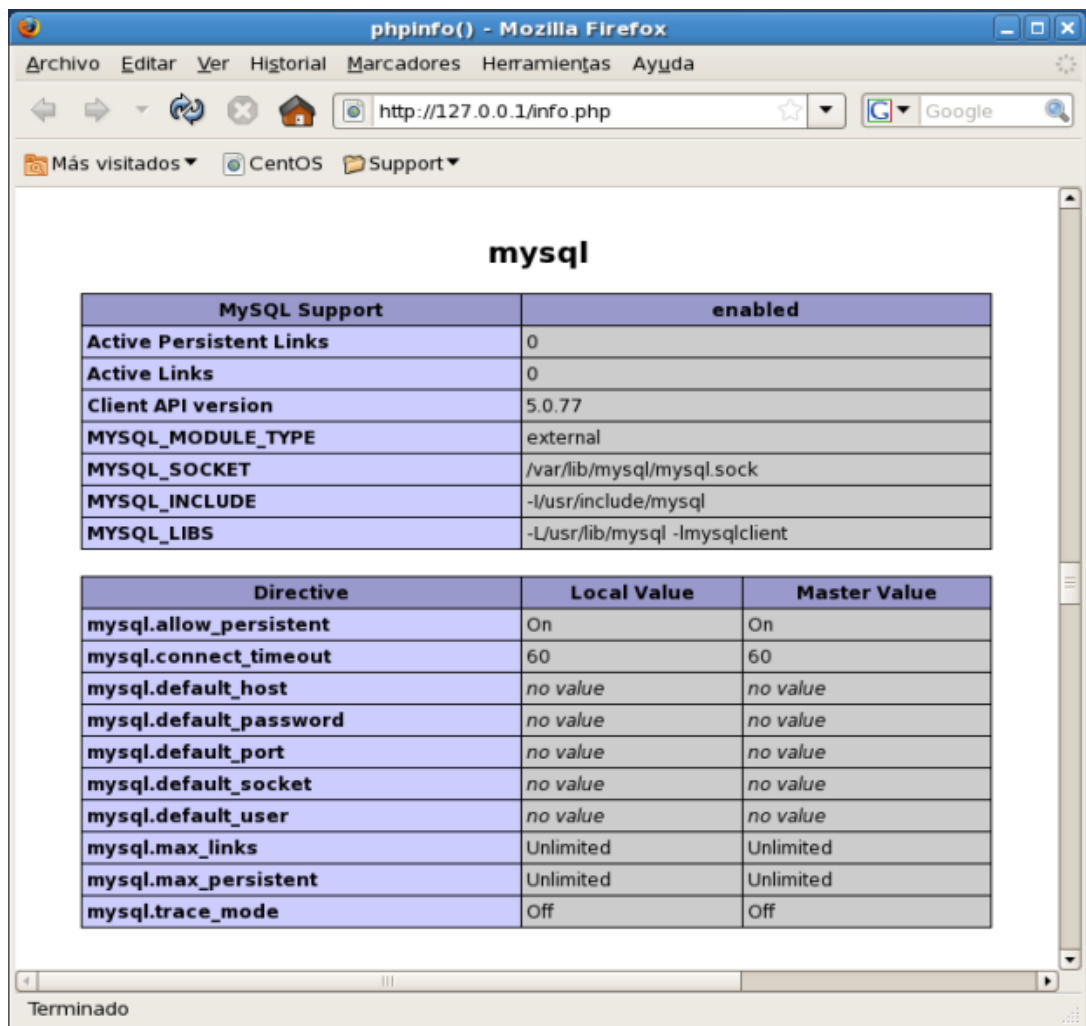
Para finalizar nos resta agregar soporte de MySQL a PHP usando el siguiente comando:

```
#yum install php-mysql
```

Reiniciamos Apache usando

```
#/etc/init.d/httpd start
```

Si navegamos la URL `http://direccion server/info.php` tenemos que ver en la parte media lo q se muestra en la Figura 9:



MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.0.77
MYSQL_MODULE_TYPE	external
MYSQL_SOCKET	/var/lib/mysql/mysql.sock
MYSQL_INCLUDE	-I/usr/include/mysql
MYSQL_LIBS	-L/usr/lib/mysql -lmysqlclient

Directive	Local Value	Master Value
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	no value	no value
mysql.default_password	no value	no value
mysql.default_port	no value	no value
mysql.default_socket	no value	no value
mysql.default_user	no value	no value
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

Figura 9: Vista de la instalación de MySQL exitosamente.

3.3.3 Instalación de voces en español

Como nos podemos dar cuenta, la configuración por defecto de las voces en Asterisk están dadas en el idioma inglés; por lo que se nos hizo necesario cambiar esta configuración y descargar los paquetes de voces en español.

Los paquetes de voces los descargamos desde el sitio Guadalinex.org [12].

Encontrándonos con el inconveniente de que dichos paquetes solo se encuentran en .deb por lo que nos vimos en la necesidad de descargar el software Alien, el cual hace la conversión de los paquetes .deb a .rpm y así poderlos instalar en nuestro Centos sin ningún problema.

Instalación de Alien

Primero descargamos el software desde la siguiente dirección ftp.de.debian.org. [13]

Una vez descargado lo descomprimos

```
#tar xzvf alien_8.72.tar.gz
```

Descargamos unos paquetes que nos hacen falta para la instalación del programa:

```
#yum install perl
```

Instalamos también una utilidad para crear los paquetes en Centos

```
#yum install rpm-build
```

Ahora si entramos a la carpeta de alien y lo instalamos

```
#cd alien
```

```
#perl Makefile.PL
```

```
#make
```

```
#make install
```

Listo ahora descargamos los paquetes debian con las voces desde:

```
#wget http://forja.guadalinex.org/repositorio/frs/download.php/154/festvox-  
sflpc16k_1.0-1_all.deb
```

```
#wget http://forja.guadalinex.org/repositorio/frs/download.php/153/festvox-  
palpc16k_1.0-1_all.deb
```

Transformamos los paquetes en .rpm

```
#alien -rv festvox-palpc16k_1.0-1_all.deb
```

```
#alien -rv festvox-sflpc16k_1.0-1_all.deb
```

Ahora si podemos instalar los paquetes para Centos.

```
#rpm -ivh festvox-palpc16k-1.0-2.noarch.rpm
```

```
#rpm -ivh festvox-sflpc16k-1.0-2.noarch.rpm
```

En la carpeta /usr/share/festival/voices/spanish encontraremos estas dos carpetas:

JuntaDeAndalucia_es_pa_diphone

JuntaDeAndalucia_es_sf_diphone

La primera es la masculina y la segunda la femenina.

Modificamos la configuración de festival para usar la voz femenina.

```
#nano /usr/share/festival/festival.scm
```

Antes de la última línea del archivo pegamos:

```
;(language__spanish)
```

```
(set! voice_default 'voice_JuntaDeAndalucia_es_sf_diphone)
```

```
  (define (tts_textasterisk string mode)
```

```
    "(tts_textasterisk STRING MODE)
```

```
    Apply tts to STRING. This function is specifically designed for  
    use in server mode so a single function call may synthesize the string.
```

```
    This function name may be added to the server safe functions."
```

```
    (let ((wholeutt (utt.synth (eval (list 'Utterance 'Text string)))))
```

```
      (utt.wave.resample wholeutt 8000)
```

```
      (utt.wave.rescale wholeutt 5)
```

```
      (utt.send.wave.client wholeutt)))
```

Guardamos los cambios y ahora iremos a la carpeta `/var/spool/asterisk/tmp` y la vaciamos, hacemos estos porque se debe tomar en cuenta la cache y los ficheros de voz son guardados por 30 días. Listo ahora si tenemos en nuestro proyecto voces en español.

3.3.4 Creación de la Base de Datos

Iniciamos MySQL y para conectarnos usamos el siguiente comando donde `-u` hace referencia a un usuario y `-p` a una clave, con esto lograremos que al ingresar a las base de datos y siempre nos pida un user y una password:

```
#mysql start
```

```
#mysql -u root -p password
```

```
#mysql -p
```

Donde nos pedirán que ingresemos la clave configurada con el anterior comando, como vemos en la Figura 10.

```
[root@localhost ~]# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.0.77 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> █
```

Figura 10: Vista del comando `#mysql -p`.

Para crear nuestra base de datos usamos:

```
#create database notas;
```

Posteriormente verificamos que la base de datos fue creada. (Ver Figura 11):

```
#show databases;
```

```
mysql> show databases;
+-----+
| Database           |
+-----+
| information_schema |
| mysql              |
| notas              |
| test               |
+-----+
4 rows in set (0.01 sec)
```

Figura 11: Salida de pantalla de un show database.

Selecciona la base de datos con:

```
#use notas;
```

Procedemos a crear su tabla con sus respectivos campos:

```
#CREATE TABLE academia(cedula varchar(10) NOT NULL,
nombres varchar(25)NOT NULL,
apellidos varchar(25) NOT NULL,
ccna1 int(3),
ccna2 int(3),
```

```

ccna3 int(3),
ccna4 int(3),
ccnp int(3),
it1 int(3),
redes_inalambricas int(3),
cableado_estructurado int(3)
KEY(cedula) );

```

Para comprobar que la tabla y sus campos fueron creados usamos:

```
#show tables;
```

```
#describe academia;
```

Observaremos tal cual nos muestra la Figura 12.

```

mysql> show tables;
+-----+
| Tables_in_notas |
+-----+
| academia        |
+-----+
1 row in set (0.01 sec)

mysql> describe academia;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| cedula         | varchar(10)   | NO   | MUL | NULL     |       |
| nombres       | varchar(25)   | NO   |     | NULL     |       |
| apellidos     | varchar(25)   | NO   |     | NULL     |       |
| ccna1         | int(3)        | YES  |     | NULL     |       |
| ccna2         | int(3)        | YES  |     | NULL     |       |
| ccna3         | int(3)        | YES  |     | NULL     |       |
| ccna4         | int(3)        | YES  |     | NULL     |       |
| ccnp1         | int(3)        | YES  |     | NULL     |       |
| it1           | int(3)        | YES  |     | NULL     |       |
| cableado_estructurado | int(3)        | YES  |     | NULL     |       |
| redes_inalambricas | int(3)        | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.05 sec)

```

Figura 12: Muestra de pantalla de una base de datos creada.

Luego ingresamos algunos datos en los campos para visualizar que funciona correctamente la tabla usamos la siguiente estructura:

```
#Insert Into academia(cedula, nombres, apellidos, ccna1, ccna2,ccna3,ccna4,ccnp1,it1,redes_inalambricas,cableado_estructurado) Values ('0925470312', 'Sandra Katherine', 'Pineda Obando', 80, 100,88,80,81,85,89,84);
```

Verificamos con:

```
#SELECT * FROM academia;
```

Y nos arrojará el siguiente resultado mostrado en la Figura 13.

```
11 rows in set (0.02 sec)

mysql> SELECT * FROM academia;
+-----+-----+-----+-----+-----+-----+-----+
| cedula      | nombres          | apellidos          | ccna1 | ccna2 | ccna3 | ccna4 |
| ccnp1      | it1             | cableado_estructurado | redes_inalambricas |
+-----+-----+-----+-----+-----+-----+-----+
| 0967345912 | Eduardo Gaston  | Sinche Reyes      | 88    | 80    | 85    | 83    |
| NULL       | 89              | 88                | NULL  |      |      |      |
| 1205487661 | Angel Patricio  | Aguirre Sanabria  | 91    | 83    | 86    | 88    |
| 82         | 90              | 80                | 81    |      |      |      |
| 0925470312 | Sandra Katherine | Pineda Obando     | 88    | 87    | 83    | 84    |
| 90         | 88              | 80                | 81    |      |      |      |
| 0967895218 | Alberto Eduardo | Abad Eras         | 84    | 80    | 83    | 86    |
| 89         | 80              | 82                | 87    |      |      |      |
| 0934789516 | Maria Belen     | Sotaminga Reyes   | 83    | 80    | 82    | 82    |
| 81         | 90              | 92                | 87    |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.02 sec)

mysql> █
```

Figura 13: Vista de una base de datos con sus campos llenos.

3.4 CONFIGURACION DE LOS ARCHIVOS DE ASTERISK.

3.4.1 Configuración del archivo sip.conf

El archivo sip.conf contiene parámetros relacionados con la configuración SIP de Asterisk. En este archivo se definen variables generales, clientes y servidores SIP y se estructura en secciones donde cada sección se define por un nombre entre corchetes seguido de las opciones de cada sección.

La primera sección, definida como general, define las opciones generales del servidor como language, códecs habilitados y no habilitados, etc.

Las siguientes secciones definen parámetros del cliente como el nombre de usuario, clave u otras.

El archivo de configuración de SIP se encuentra en la ruta /etc/asterisk/

[general] ;La sección [general] es donde se define variables globales y aspectos por defecto para los canales SIP.

context=default ;Contexto por defecto donde entrarán las llamadas entrantes por el canal SIP.
Este contexto se define en extensions.conf

srvlookup=yes ;Permite hacer búsquedas de registros DNS.
SRV para llamadas SIP salientes basadas en los nombres de dominio.

language=es ;Aquí definimos el idioma a usar por la extensión.

disallow=all ;Deshabilita todos los códecs.

allow=ulaw,alaw,gsm,wav ;Habilita solo los códec mencionados.

[501] ; Configuración del cliente.
Se le ha definido la extensión 501.

callerid=softphone ;El nombre de identificación de la extensión.

type=friend ;Tipo de extensión, puede ser user, peer o friend, user solo puede recibir y peer solo puede hacer mientras que friend puede hacer y recibir llamadas.

secret=1234	;La clave de la extensión.
qualify=yes	;Para saber el tiempo de respuesta de una extensión y si está alcanzable o no.
nat=yes	;Esto se pone si la extensión se conecta al servidor Asterisk detrás de un cortafuegos.
host=dynamic	;Habilitamos que el teléfono se pueda habilitar desde cualquier ip.
canreinvite=no	;Ponemos no para que Asterisk haga de puente entre las extensiones.
context=interno	;El contexto que se utiliza en el extensions.conf para las llamadas entrantes

Configuración final del sip.conf

[general]

context=default

srvlookup=yes

language=yes

disallow=all

allow=ulaw,alaw,gsm,wav

[501]

callerid=softphone

type=friend

secret=1234

qualify=yes

nat=yes

host=dynamic

canreinvite=no

context=interno

Luego de que hemos ingresado la configuración utilizamos el comando reload en el CLI (línea de comando) de Asterisk con esto recargamos la configuración, aunque también podemos recargar solamente la configuración del archivo sip con el comando “sip reload”.

3.4.2 Configuración del archivo extensions.conf

En el archivo de configuración extensions.conf se configura el plan de marcado y el comportamiento de todas las conexiones a través de la

PBX; controla como se gestionan y encaminan las llamadas entrantes y salientes del sistema Asterisk.

El plan de marcado es aquel que sigue una llamada desde que entra o sale del sistema hasta que llega a su punto final, se divide en secciones llamadas contextos, definidos entre corchetes, donde cada contexto consiste de varias extensiones.

Cada extensión es una lista de comandos a ejecutar con una prioridad y una aplicación concreta para controlar el comportamiento de la llamada y del sistema en si (hangup, dial,etc).

Las extensiones se acceden cuando se recibe una llamada entrante por un canal dado, el usuario que ha llamado marca la extensión o se ejecuta un salto de extensiones desde el plan de marcado de Asterisk. Ver Anexo A.

El archivo de configuración extensions.conf se encuentra en la ruta /etc/asterisk

[general] ;Contexto donde se configuran ajustes a generales para nuestro plan de marcado.

autofallthrough=yes ;Este valor permite que la llamada se termine de la mejor manera.

clearglobalvars=no ;De este modo las variables globales definidas en el archivo extensions.conf persistirán a los reloads y no serán eliminadas.

[interno] ;Contexto que indica que hacer con las llamadas entrantes

exten => _50X,1,Answer()

; Acepta la llamada entrante por el canal

exten => _50X,n,Background(Bienvenida)

; El usuario escucha un mensaje de bienvenida

exten => _50X,n,Background(Pedido)

; Aquí le pedimos al usuario que ingrese 1 para ir al contexto notas a escuchar sus calificaciones, 2 para escuchar información acerca de cursos que dicta la Academia y 3 para colgar la llamada.

exten => _50X,n,WaitExten()

; Mediante WaitExten esperamos el ingreso

exten => 1,1,Goto(notas,s,1)

; Si el usuario ingreso 1 hace el salto acá y va al
contexto notas

exten => 2,1,Goto(informacion,s,1)

; Si el usuario ingreso 2 hace el salto acá y va al
contexto información.

exten => 3,1,Hangup()

; Si el usuario ingreso 3 cuelga la llamada.

exten => i,1,Background(invalido)

; Si el usuario ingresa algún valor no.

exten => i,n,Goto(interno,s,1)

; Válido mediante estas líneas escuchará un
mensaje diciendo que es inválido el número
ingresado y se hace un salto hacia el inicio del
contexto.

exten => t,1,Playback(goodbye)

; Si el usuario no ingresa un valor durante un

exten => t,n,Hangup() ; Tiempo escuchara un mensaje de despedida
y cuelga la llamada

exten => h,1,Hangup()

; Si se produce algún error como esta configurado para que Asterisk termine la llamada de la mejor manera, colgará.

[notas]

; En este contexto se manejará la consulta de notas.

exten => s,1,Background(cedula)

; Se pide al usuario ingrese su número de cédula.

exten => s,2,Read(CI,,10)

; Se almacena el número de cédula en la variable CI.

exten => s,3,SayDigits(\${CI})

; Le repetimos al usuario los dígitos de la cédula que ingresó.

exten => s,4,Playback(validar)

; Escuchará un mensaje el cual le pregunta si el número de cédula esta correcto.

exten => s,5,WaitExten()

; Esperamos que ingrese q si el número está correcto, 2 si no lo está.

exten => 1,1,Agi(notas.agi,{CI})

; Llamamos a la función AGI, pasando como parámetros el archivo agi y la variable CI que contiene el número de cédula.

exten => 2,1,Goto(s,1)

; Si el usuario ingreso la opción 2 entonces regresamos al inicio del contexto para que pueda volver a ingresar su cédula.

exten => t,1,Playback(goodbye)

; Si el usuario no ingresa un valor durante un

exten => t,2,Hangup() tiempo escuchara un mensaje de despedida y cuelga la llamada

[informacion]

; Contexto en el cual se brinda información acerca de los cursos dictados por Cisco

exten => s,1,Background(info)

; El usuario escucha un archivo de audio con los cursos que se dictarán.

exten => s,2,Background(sugerencia)

; Se le sugiere al usuario volver a escuchar el audio de información, ir al contexto notas o colgar.

exten => 1,1,Goto(informacion,s,1)

; Si el usuario eligió 1 se reproduce nuevamente el archivo de información.

exten => 2,1,Goto(notas,s,1)

; Si el usuario ingreso 2 se hace un salto al contexto notas.

exten => 3,1,Hangup()

; Si el usuario ingreso 3 se cuelga.

exten => t,1,Playback(goodbye)

; Si el usuario no ingresa un valor durante un

exten => t,2,Hangup() ;tiempo escuchara un mensaje de despedida y cuelga la llamada.

[grabar]

;En este contexto vamos a grabar los archivos de audio que se reproducen a lo largo del extensions.

exten => 100,1,Answer()

; Se acepta a llamada entrante a la extensión 100.

exten => 100,2,Record(*"nombredelarchivo"*%d.wav)

;Se utiliza la función Record de Asterisk para grabar el archivo en formato wav.

exten => 100,3,Hangup()

; Se cuelga la llamada.

CAPITULO 4

FUNCIONAMIENTO Y PRUEBAS

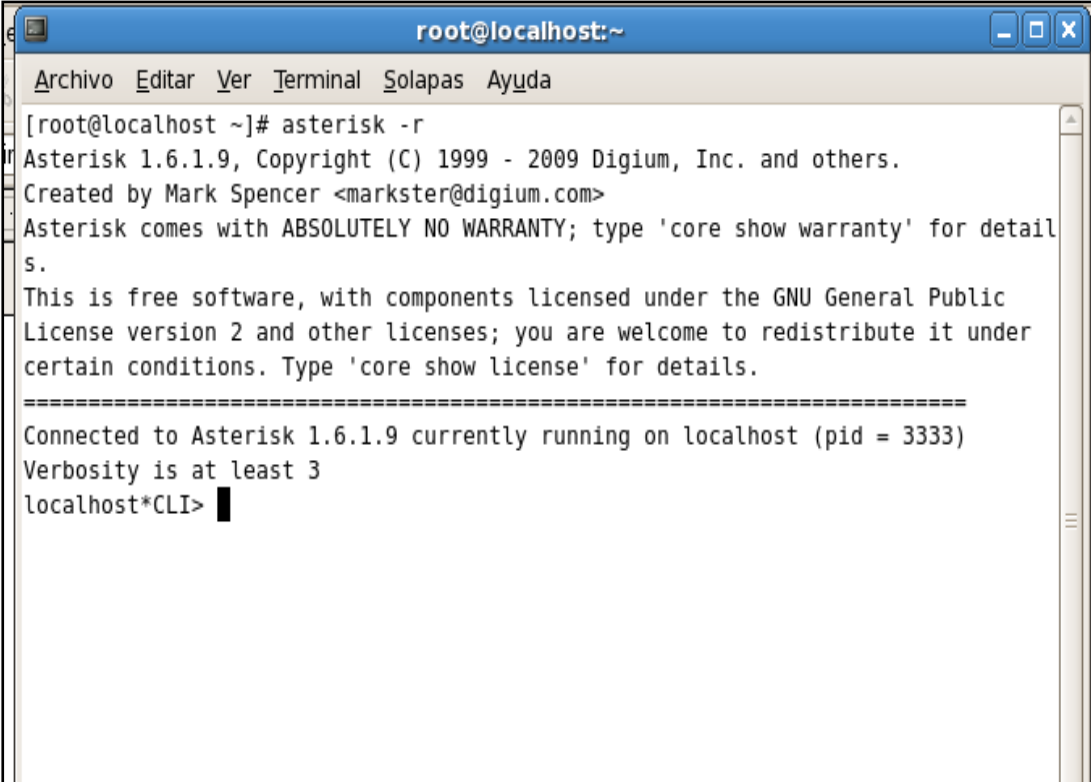
4.1 FUNCIONAMIENTO Y PRUEBAS

Una vez realizada la instalación de todos los componentes necesarios para el buen funcionamiento de nuestro proyecto, lo que haremos ahora son pruebas de funcionalidad ayudándonos de un softphone; pero antes de realizar esto, iniciaremos nuestro servidor para verificar en la consola cuando el softphone se registre.

Para iniciar Asterisk a través de la consola usamos el comando:

```
#asterisk -r
```

Y vemos q el prompt cambia en la Figura 14.



```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ~]# asterisk -r
Asterisk 1.6.1.9, Copyright (C) 1999 - 2009 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for detail
S.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.6.1.9 currently running on localhost (pid = 3333)
Verbosity is at least 3
localhost*CLI> █
```

Figura 14: Consola de inicio de Asterisk.

Ahora que ya tenemos Asterisk iniciado podemos proceder a configurar el softphone X-Lite que se muestra en la Figura 15 con los siguientes valores:

Account name: Prueba

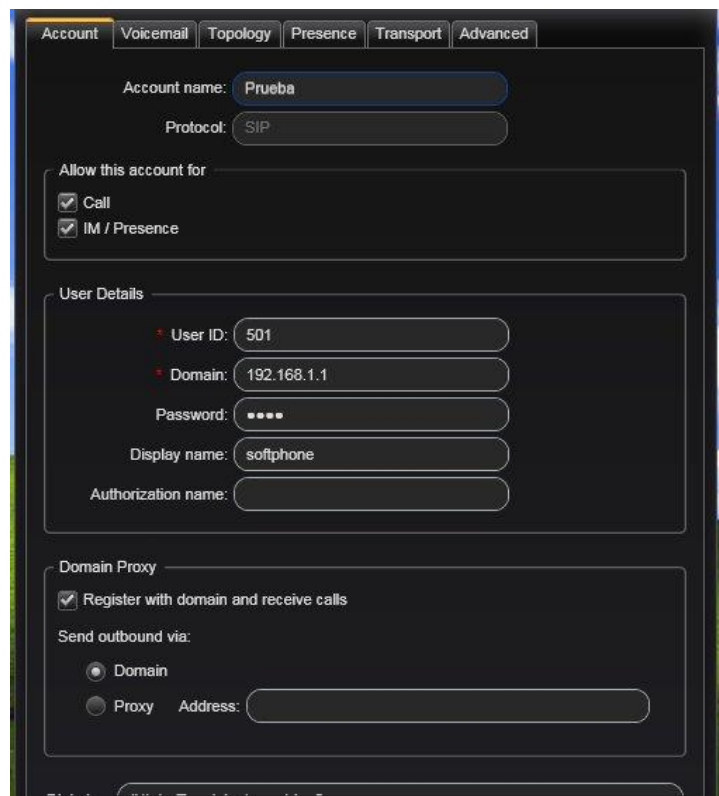
Protocol: SIP

User Id: 501

Domain: 192.168.1.1

Password: 1234

Display name: Softphone



The screenshot shows the Asterisk SIP account configuration interface. At the top, there are tabs for 'Account', 'Voicemail', 'Topology', 'Presence', 'Transport', and 'Advanced'. The 'Account' tab is selected. The configuration fields are as follows:

- Account name: Prueba
- Protocol: SIP
- Allow this account for:
 - Call
 - IM / Presence
- User Details:
 - User ID: 501
 - Domain: 192.168.1.1
 - Password: ****
 - Display name: softphone
 - Authorization name: (empty)
- Domain Proxy:
 - Register with domain and receive calls
 - Send outbound via:
 - Domain
 - Proxy Address: (empty)

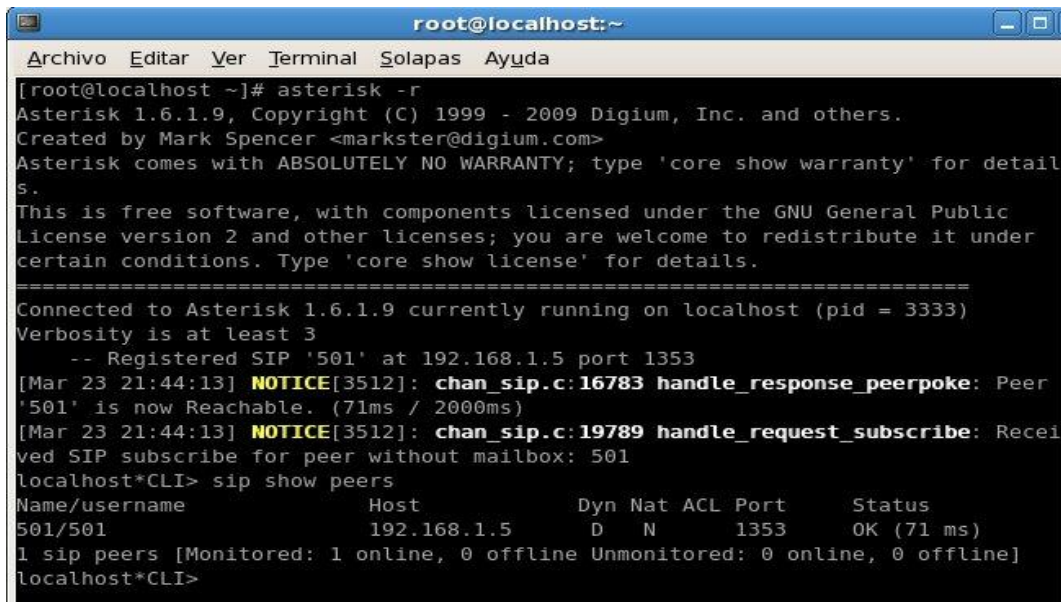
Figura 15: Configuración del Softphone X-Lite.

Para verificar que nuestro softphone se ha registrado correctamente lo hacemos ingresando en la consola (ver Figura 16) y con el comando sip show peers como se muestra en la Figura 17:



```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ~]# asterisk -r
Asterisk 1.6.1.9, Copyright (C) 1999 - 2009 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public License version 2 and other licenses; you are welcome to redistribute it under certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.6.1.9 currently running on localhost (pid = 3333)
Verbosity is at least 3
-- Registered SIP '501' at 192.168.1.1 port 39241
[Mar 23 21:06:15] NOTICE[3512]: chan_sip.c:16783 handle_response_peerpoke: Peer '501' is now Reachable. (39ms / 2000ms)
-- Unregistered SIP '501'
-- Registered SIP '501' at 192.168.1.1 port 39241
[Mar 23 21:06:15] NOTICE[3512]: chan_sip.c:16783 handle_response_peerpoke: Peer '501' is now Reachable. (26ms / 2000ms)
[Mar 23 21:06:15] NOTICE[3512]: chan_sip.c:19789 handle_request_subscribe: Received SIP subscribe for peer without mailbox: 501
-- Using SIP RTP CoS mark 5
```

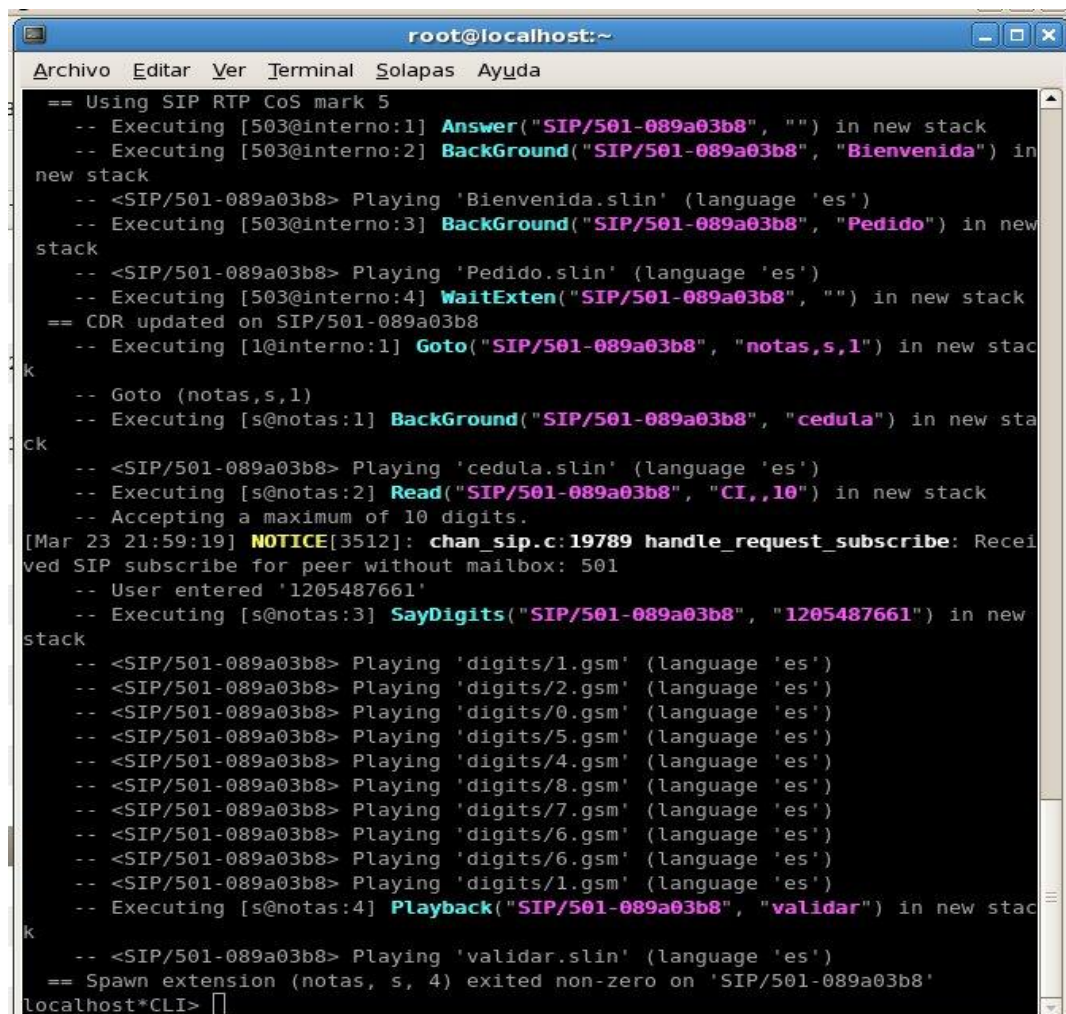
Figura 16: Registro de softphone X-Lite.



```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ~]# asterisk -r
Asterisk 1.6.1.9, Copyright (C) 1999 - 2009 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public License version 2 and other licenses; you are welcome to redistribute it under certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.6.1.9 currently running on localhost (pid = 3333)
Verbosity is at least 3
-- Registered SIP '501' at 192.168.1.5 port 1353
[Mar 23 21:44:13] NOTICE[3512]: chan_sip.c:16783 handle_response_peerpoke: Peer '501' is now Reachable. (71ms / 2000ms)
[Mar 23 21:44:13] NOTICE[3512]: chan_sip.c:19789 handle_request_subscribe: Received SIP subscribe for peer without mailbox: 501
localhost*CLI> sip show peers
Name/username      Host              Dyn Nat ACL Port      Status
501/501            192.168.1.5      D  N   1353     OK (71 ms)
1 sip peers [Monitored: 1 online, 0 offline Unmonitored: 0 online, 0 offline]
localhost*CLI>
```

Figura 17: Salida del comando sip show peers.

Ahora que ya hemos configurado nuestro Softphone X-Lite y éste se ha registrado correctamente, realizamos una llamada para verificar que nuestro plan de marcado se esté realizando correctamente, marcamos la extensión 503 en el softphone y obtendremos en consola lo que vemos en la Figura 18:



```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
== Using SIP RTP CoS mark 5
-- Executing [503@interno:1] Answer("SIP/501-089a03b8", "") in new stack
-- Executing [503@interno:2] Background("SIP/501-089a03b8", "Bienvenida") in
new stack
-- <SIP/501-089a03b8> Playing 'Bienvenida.slin' (language 'es')
-- Executing [503@interno:3] Background("SIP/501-089a03b8", "Pedido") in new
stack
-- <SIP/501-089a03b8> Playing 'Pedido.slin' (language 'es')
-- Executing [503@interno:4] WaitExten("SIP/501-089a03b8", "") in new stack
== CDR updated on SIP/501-089a03b8
-- Executing [l@interno:1] Goto("SIP/501-089a03b8", "notas,s,1") in new stack
-- Goto (notas,s,1)
-- Executing [s@notas:1] Background("SIP/501-089a03b8", "cedula") in new stack
-- <SIP/501-089a03b8> Playing 'cedula.slin' (language 'es')
-- Executing [s@notas:2] Read("SIP/501-089a03b8", "CI,,10") in new stack
-- Accepting a maximum of 10 digits.
[Mar 23 21:59:19] NOTICE[3512]: chan_sip.c:19789 handle_request_subscribe: Received SIP
subscribe for peer without mailbox: 501
-- User entered '1205487661'
-- Executing [s@notas:3] SayDigits("SIP/501-089a03b8", "1205487661") in new
stack
-- <SIP/501-089a03b8> Playing 'digits/1.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/2.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/0.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/5.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/4.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/8.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/7.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/6.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/6.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/1.gsm' (language 'es')
-- Executing [s@notas:4] Playback("SIP/501-089a03b8", "validar") in new stack
-- <SIP/501-089a03b8> Playing 'validar.slin' (language 'es')
== Spawn extension (notas, s, 4) exited non-zero on 'SIP/501-089a03b8'
localhost*CLI> █
```

Figura 18: Vista de un plan de marcado funcionando.

A continuación haremos una explicación del funcionamiento de nuestro proyecto basándonos en las salidas que se dan por consola cuando una llamada esta activa.

== Using SIP RTP CoS mark 5

-- Executing [503@interno:1] Answer("SIP/501-089a03b8", "") in new stack

-- Executing [503@interno:2] Background("SIP/501-089a03b8", "Bienvenida") in new stack

-- <SIP/501-089a03b8> Playing 'Bienvenida.slin' (language 'es')

-- Executing [503@interno:3] Background("SIP/501-089a03b8", "Pedido") in new stack

-- <SIP/501-089a03b8> Playing 'Pedido.slin' (language 'es')

-- Executing [503@interno:4] WaitExten("SIP/501-089a03b8", "") in new stack

== CDR updated on SIP/501-089a03b8

-- Executing [1@interno:1] Goto("SIP/501-089a03b8", "notas,s,1") in new stack

-- Goto (notas,s,1)

En las líneas de la parte superior vemos como una vemos que se marcado la extensión 503 el comando Answer responde la llamada, dándole a escuchar al usuario un mensaje llamado “Bienvenida”, el cual le da la bienvenida al call center de la Academia, luego de esto en la siguiente ejecución oirá un audio en el cual le da opciones a elegir entre las que están; 1 consultar notas, 2 para obtener información y 3 para salir.

El comando WaitExten espera por el ingreso de usuario de las opciones mencionadas, se ingresa 1 por lo que se hace un salto hacia el contexto notas.

```
-- Executing [s@notas:1] Background("SIP/501-089a03b8", "cedula") in
new stack
-- <SIP/501-089a03b8> Playing 'cedula.slin' (language 'es')
-- Executing [s@notas:2] Read("SIP/501-089a03b8", "Cl,,10") in new
stack
-- Accepting a maximum of 10 digits.
-- User entered '1205487661'
-- Executing [s@notas:3] SayDigits("SIP/501-089a03b8", "1205487661")
in new stack
-- <SIP/501-089a03b8> Playing 'digits/1.gsm' (language 'es')
```

```

-- <SIP/501-089a03b8> Playing 'digits/2.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/0.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/5.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/4.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/8.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/7.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/6.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/6.gsm' (language 'es')
-- <SIP/501-089a03b8> Playing 'digits/1.gsm' (language 'es')
--- Executing [s@notas:4] Playback("SIP/501-089a03b8", "validar") in new
stack
-- <SIP/501-089a03b8> Playing 'validar.slin' (language 'es')
-- Executing [s@notas:5] WaitExten("SIP/501-089a03b8", "") in new stack
== CDR updated on SIP/501-089a03b8
--      Executing      [1@notas:1]      AGI("SIP/501-089a03b8",
"notas.agi,1205487661") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/notas.agi

```

Una vez que se está ejecutando el contexto notas, se le hace al usuario el pedido de su número de cédula, el cual es almacenado en una variable con un valor máximo de 10; se le repite dígito por dígito el número ingresado y se le da la opción de escribirlo nuevamente si esta incorrecto

o de validarlo y continuar. Una vez validado el número de cédula se ejecuta una llamada a AGI, el cual ejecuta el script notas.agi que se encuentra almacenado en /var/lib/asterisk/agi-bin/

Dentro del script lo que se realiza es la conexión a la base de datos que contiene las notas de los alumnos registrados en diferentes cursos de la Academia, como valor de referencia se usa su número de cédula almacenado en una variable y que posteriormente fue pasado al AGI como parámetro. Si la consulta es exitosa y se encuentra el valor en la tabla se toma los valores de los otros campos y se los reproduce por medio de la función text2wav.

En el caso de que el usuario haya ingresado el valor 2 al inicio de la llamada se lo direcciona hacia el contexto de información, el cual reproduce un archivo de audio con las materias dictadas por la Academia en los campus Peñas y Prosperina. Aquí también se la brinda la oportunidad de ir hacia el contexto notas a realizar consulta de calificaciones o si lo desea puede volver a escuchar el archivo de información.

CONCLUSIONES Y RECOMENDACIONES

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.

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.

ANEXOS

ANEXO A: ARCHIVO EXTENSIONS

[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) ;Pedimos ingreso de cedula
exten => s,2,Read(CI,,10) ;guardamos el valor de cedula en ci
exten => s,3,SayDigits(\${CI}) ;le decimos los números que ingreso
exten => s,4,Playback(validar) ;pedimos que confirme la cedula
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()

ANEXO B: ARCHIVO AGI

```
#!/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

$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);

if($contenido[0]==$identificador)
{
$agi->text2wav("Nombres $contenido[1]");
$agi->text2wav("Apellidos $contenido[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]");
$agi->text2wav("$contenido[$indice]");
$agi->text2wav("Si desea escuchar de nuevo la nota presione 1");
$repetir = $agi->get_data('beep',3000,1);
}while($repetir['result'] == 1);
}
}

```

```
}  
$agi->exec('Background', 'goodbye');  
}  
else if($contenido[0]!=$identificador){  
    $agi->text2wav("Cedula no registrada en nuestra base de datos");  
}  
fclose($in);  
fclose($stderr);  
  
mysql_close($conectar);  
$agi->hangup();  
  
?>
```

ANEXO C: PAGINA WEB PARA INGRESO DE ALUMNOS A LA BASE

El modelo del formulario hecho en PHP se ve reflejado en la Figura 19.

The image shows a web form titled "ACADEMIA CISCO - ESPOL" with the subtitle "INGRESO DE CALIFICACIONES" and "Nueva Calificación". Below the title is a dashed horizontal line. The form contains the following fields and labels:

- Calificaciones del alumno**
- Ingrese la cedula:
- Ingrese los nombres:
- Ingrese los apellidos:
- Ingrese la nota de ccna1:
- Ingrese la nota de ccna2:
- Ingrese la nota de ccna3:
- Ingrese la nota de ccna4:
- Ingrese la nota de ccnp1:
- Ingrese la nota de it1:
- Ingrese la nota de cableado estructurado:
- Ingrese la nota de redes inalámbricas:
-

Figura 19: Página Web para el ingreso de los alumnos.

Código de Página Web

```
<HTML>
<HEAD>
<TITLE>Academia Cisco - ESPOL</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-
8859-1">

</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0
MARGINWIDTH=0 MARGINHEIGHT=0 text="#17334A" link="#000040"
vlink="#000040" alink="#000040" background="images/index_bg.gif">
<TABLE WIDTH=790 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD COLSPAN=3>&nbsp;</TD>
<TD ROWSPAN=6 background="images/index_02.gif">&nbsp;</TD>
</TR>

<TR>
<TD COLSPAN=3> <div align="center"></div></TD>
</TR>
```

```
<TR>
<TD COLSPAN=3>&nbsp;</TD>
</TR>
```

```
<TR>
<TD COLSPAN=3> <IMG SRC="images/index_05.gif" WIDTH=780
HEIGHT=25></TD>
</TR>
```

```
<TR>
<TD COLSPAN=3> <IMG SRC="images/index_06.gif" WIDTH=780
HEIGHT=4></TD>
</TR>
```

```
<TR>
<TD background="images/index_07.gif" valign="top"><b><font size="2"
face="Verdana, Arial, Helvetica, sans-serif">Main<br>
</font></b><font size="1" face="Verdana, Arial, Helvetica, sans-
serif"><br>
- <a href="link">Link</a></font><font size="2" face="Verdana, Arial,
Helvetica, sans-serif"><br>
</font><font size="1" face="Verdana, Arial, Helvetica, sans-serif">- <a
```

href="link">Link<font size="2" face="Verdana, Arial, Helvetica,
sans-serif">

- Link<font size="2" face="Verdana, Arial, Helvetica,
sans-serif">

- Link<font size="2" face="Verdana, Arial, Helvetica,
sans-serif">

- Link<font size="2" face="Verdana, Arial, Helvetica,
sans-serif">

Affiliates

- Link<font size="2" face="Verdana, Arial,
Helvetica, sans-serif"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif">

- Link<font size="2" face="Verdana, Arial, Helvetica,
sans-serif">

- Link<font size="2" face="Verdana, Arial, Helvetica,

sans-serif">

- Link

- Link

</TD>

<TD background="images/index_08.gif"> </TD>

<TD height="456" valign="top" background="images/index_09.gif">

<p align="center">ACADEMIA CISCO -
ESPOL </p>

<p align="center"><font size="3"
face="Verdana, Arial, Helvetica, sans-serif" color="#000040">INGRESO
DE CALIFICACIONES </p>

<p align="center">Nueva Calificación </p>

<p>-----

-----</p>

<h1>Calificaciones del
alumno</h1>

<form action="ingreso.php" method="get">

Ingrese la cedula:

<input type="text" name="cedula">

Ingrese los nombres:

<input type="text" name="nombres">

Ingrese los apellidos:

<input type="text" name="apellidos">

Ingrese la nota de ccna1:

<input type="number" name="ccna1">

Ingrese la nota de ccna2:

<input type="number" name="ccna2">

Ingrese la nota de ccna3:

<input type="number" name="ccna3">

Ingrese la nota de ccna4:

<input type="number" name="ccna4">

Ingrese la nota de ccnp1:

<input type="number" name="ccnp1">

Ingrese la nota de it1:

```
<input type="number" name="it1"><br>
```

Ingrese la nota de cableado estructurado:

```
<input type="number" name="cableado_estructurado"><br>
```

Ingrese la nota de redes inalámbricas:

```
<input type="number" name="redes_inalambricas"><br>
```

```
<input type="submit" value="Ingresar">
```

```
</form> </p></TD>
```

```
</TR>
```

```
<TR>
```

```
<TD align="top"></TD>
```

```
<TD></TD>
```

```
<TD></TD>
```

```
<TD></TD>
```

```
</TR>
```

```
</TABLE>
```

```
</BODY>
```

```
</HTML>
```

EJEMPLO DE INGRESO DE ALUMNOS POR PÁGINA WEB

A continuación ingresaremos un alumno con varias notas a través de la interfaz web como se ve en la Figura 20, lo que se debería ver reflejado en la tabla una vez que demos clic en Ingresar.

ACADEMIA CISCO - ESPOL

INGRESO DE CALIFICACIONES

Nueva Calificación

Calificaciones del alumno

Ingrese la cedula:

Ingrese los nombres:

Ingrese los apellidos:

Ingrese la nota de ccna1:

Ingrese la nota de ccna2:

Ingrese la nota de ccna3:

Ingrese la nota de ccna4:

Ingrese la nota de ccnp1:

Ingrese la nota de it1:

Ingrese la nota de cableado estructurado:

Ingrese la nota de redes inalambricas:

Figura 20: Ingreso de un alumno mediante el formulario.

Ahora nos vamos a la consola y con el comando SELECT *from academia verificamos como se muestra en la Figura 21 que se ha ingresado con éxito los datos.

```

root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
83 | NULL | 89 | 88 | NULL |
| 1205487661 | Angel Patricio | Aguirre Sanabria | 91 | 83 | 86 |
88 | 82 | 90 | 80 | 81 |
| 0925470312 | Sandra Katherine | Pineda Obando | 88 | 87 | 83 |
84 | 90 | 88 | 80 | 81 |
| 0967895218 | Alberto Eduardo | Abad Eras | 84 | 80 | 83 |
86 | 89 | 80 | 82 | 87 |
| 0934789516 | Maria Belen | Sotaminga Reyes | 83 | 80 | 82 |
82 | 81 | 90 | 92 | 87 |
| 1204567345 | Carlos Jose | Andrade Vera | 87 | 88 | 90 |
93 | 0 | 88 | 80 | 81 |
| 1203675683 | Jose Antonio | Mendoza Buenaño | 80 | 84 | 90 | 8
9 | 79 | 85 | 92 | 88 |
| 0925467883 | Andres Esteban | Leon Vite | 88 | 85 | 81 |
92 | 90 | 87 | 0 | 0 |
| 1204567342 | Jose Carlos | Perez Andrade | 87 | 88 | 90 |
93 | 0 | 88 | 80 | 81 |
| 0958732117 | Jonhatan David | Escobar Mejia | 87 | 90 | 92 |
89 | 0 | 89 | 0 | 84 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>

```

Figura 21: Base de datos, tomando resultados del formulario.

GLOSARIO DE TERMINOS

VoIP: Voz sobre el protocolo internet, es un método por el cual tomamos señales de audio analógicas y la transforma en datos digitales que pueden ser transmitidos por Internet a través de un IP determinada.

PSTN: Red de Telefonía Pública Conmutada (Public Switched Telephone Network), cubre tanto la telefonía fija como la móvil.

OpenSource: Es el termino con el que se conoce a los softwares distribuidos y desarrollados libremente.

PBX: Es la Central Telefónica Digital. Es un conmutador privado, que sirve de interfaz y maneja las comunicaciones dentro de una red telefónica privada.

SIP: Protocolo de iniciación de sesión, es un protocolo de señalización de telefonía IP utilizado para establecer, modificar y terminar llamadas VoIP. SIP fue desarrollado por el IETF y publicado como RFC 3261.

IAX/IAX2: Inter - Asterisk eXchange protocol, es un protocolo propietario, desarrollado por Mark Spencer (creador de Asterisk), aún no es un estándar.

X-lite: Softphone gratuito, que permite realizar llamadas desde el ordenador hacia los usuarios del Internet.

URL: Uniform Resource Locator, es decir, localizador uniforme de recurso y se refiere a la dirección única que identifica a una página web.

.deb: Los archivos .deb son los que provienen del sistema operativo Debian.

.rpm: Extensión de los archivos propios de Centos.

VXML: VoiceXML, permite el acceso a los servicios apoyándose de una arquitectura cliente/servidor, y de la infraestructura telefónica, el usuario puede acceder tanto con su línea fija como por móviles a los servicios.

NLU: (Natural Language Understanding). Es la habilidad que tienen los portales de voz para entender el habla natural del usuario.

BIBLIOGRAFIA

- [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.
- [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” (2ª edición),
<https://www.camundanet.com/attachments/article/86/unificadas1.pdf>, 2008-2009.
- [11]. “Lista de Bases de datos populares con Servidor”,
<http://www.entrebts.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.