

ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

INFORME DE MATERIA DE GRADUACIÓN

**“Diseño e Implementación de un Sistema Telefónico
Interactivo que me Permita Automatizar la Asignación
y Consulta de Citas Médicas”**

Previa a la obtención del TITULO de:

LICENCIADO EN REDES Y SISTEMAS OPERATIVOS

Presentada por:

**MARIA CRISTINA ALVARADO TUMBACO
STEPHANIE ALEXANDRA KRAVAROVICH GRANADOS**

GUAYAQUIL – ECUADOR

AÑO
2011

AGRADECIMIENTO

Agradezco a Dios por la fortaleza y las bendiciones que ha derramado en mí durante mi vida universitaria, para poder culminar con éxito el presente trabajo, como también agradezco a mis profesores de quienes aprendí ser perseverante y responsable.

Cristina Alvarado T.

Agradezco a Dios, mis padres, amigos y en especial a la persona con quien compartí esta última etapa por ayudarme a finalizar este proyecto, sin olvidarme de mis maestros quienes han sido el pilar principal para mi educación.

Stephanie Kravarovich G.

DEDICATORIA

Dedico el presente trabajo a mis padres quienes son la base principal para seguir adelante, también se lo dedico a mis profesores de quienes recibí un apoyo incondicional y además de mis amigos que también forman parte de mi culminación exitosa de mi carrera.

Cristina Alvarado T.

Dedico este trabajo a esas personas que me dieron su apoyo incondicional para terminar mi carrera.

Stephanie Kravarovich G.

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)

María Cristina Alvarado Tumbaco

Stephanie Alexandra Kravrovich Granados

TRIBUNAL DE SUSTENTACION

Ing. Gabriel Astudillo Brocel

PROFESOR DE LA MATERIA DE GRADUACIÓN

Ing. Albert Espinal Santana

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

RESUMEN

Este proyecto consistió en una solución para los centros médicos u hospitales, donde se busca otorgar una atención de manera eficaz y eficiente al momento, que los pacientes soliciten una cita.

Debido a la gran demanda que existe en estas instituciones, se puede observar el caso, que para obtener una cita médica se debe acudir a la institución en busca de un turno, en donde frecuentemente se ven reflejadas por filas extensas y con largas horas de espera todo con la finalidad de ser atendidos, en otro de los casos se realiza la solicitud de cita haciendo una llamada al centro médico, en donde es atendido por un agente, pero la probabilidad de que alguna de las llamadas que ingresan no se pierdan, o que a su vez sean atendidas las 24 horas y los 365 días del año son mínimas.

Existen Instituciones que por factor económico acuden a procesos en donde la inversión se ve limitada, como evitar el aumento de contrataciones de personal o de ejecutar grandes gastos en compras de equipos de cómputo, licencias de software, teléfonos etc. Por esto y otros motivos expresados anteriormente, el proyecto es basado fundamentalmente para obtener beneficios al implementar una PBX.

El software instalado es llamado Asterisk(software libre) donde su función fue la de actuar como central telefónica PBX, aquí se receptaron las llamadas que en conjunto con el sistema IVR (Interactive Voice Response) se logró exponer un menú sencillo, con opciones que debió ser escogida por el paciente, haciendo uso de un teléfono digital, su principal propósito fue la de obtener su cita médica con éxito, de esa manera se alcanzó a disminuir las llamadas perdidas, como también se consiguió la disponibilidad de uso a este sistema a un servicio de 24 horas del día y los 7 días de la semana.

El menú que se expuso, al momento que el paciente realizó la llamada al centro médico se le pidió el ingreso de su número de cedula, el cual inmediatamente fue validado por la base de datos del centro médico, una vez que la cédula fue encontrada, los procesos a continuación fueron el de escoger la especialidad médica, digitar la fecha en el formato solicitado y luego de eso, le fue automáticamente asignado la hora de la cita y para finalizar el proceso, el sistema le exigió la confirmación de su resultado para así poder ser almacenado a la base de datos.

Los requerimientos de hardware no se basó en mayores gastos, un ejemplo es el del teléfono IP el cual fue remplazado por el Softphone también perteneciente al grupo de software libre y al servidor le fue instalado el sistema operativo Linux, de la distribución conocida como CENTOS.

INDICE DE CONTENIDO

CAPÍTULO I ANTECEDENTES Y JUSTIFICACION.....	-1-
1.1 ANTECEDENTES	- 4 -
1.2 JUSTIFICACIÓN	- 5 -
1.3 DESCRIPCIÓN DEL PROYECTO.....	- 5 -
1.3.1 OBJETIVO GENERAL	- 8 -
1.3.2 OBJETIVOS ESPECÍFICOS	- 8 -
1.4 METODOLOGÍA	- 8 -
1.5 PERFIL DE LA TESIS.....	¡Error! Marcador no definido.
CAPÍTULO II ASTERISK Y PHP AGI.....	- 8 -
2.1 ASTERISK	- 9 -
2.1.1 CARACTERÍSTICAS PRINCIPALES.....	- 10 -
2.1.2 FUNCIONES BÁSICAS	- 11 -
2.1.3 FUNCIONES AVANZADAS	- 11 -
2.1.4 FUNCIONES BÁSICA DE USUARIO.....	- 12 -
2.1.5 PRINCIPALES VENTAJAS.....	- 13 -
2.2 AGI (ASTERISK GATEWAY INTERFACE)	- 14 -
2.3. CLASE PHP AGI.....	- 17 -
2.4 IVR	- 17 -
CAPÍTULO III IMPLEMENTACION	- 8 -
3.1 INTRODUCCION.....	- 20 -
3.2 HARDWARE	- 21 -
3.2.1 SERVIDOR.....	- 21 -
3.2.2 TELÉFONO IP.....	- 22 -
3.3 SOFTWARE.....	- 22 -
3.3.1 SERVIDOR	- 22 -
3.3.2 INSTALACIÓN DE ASTERISK	- 23 -
3.3.3 INSTALACIÓN DE MYSQL.....	- 26 -
3.3.4 INSTALACIÓN DE PHP	- 27 -
3.3.5 INSTALACIÓN DEL PHPMYADMIN.....	- 28 -

3.3.6	SOFTPHONE	- 29 -
3.4	BASE DE DATOS	- 30 -
3.4.1	INGRESO A MYSQL.....	- 30 -
3.4.2	CREACIÓN DE LA BASE DE DATOS.....	- 31 -
3.5	CONFIGURACIONES DE ASTERISK.....	- 33 -
3.5.1	CONFIGURACIÓN DEL ARCHIVO SIP.CONF.....	- 33 -
3.5.2	CONFIGURACIÓN DEL ARCHIVO EXTENSIONS.CONF	- 34 -
3.6	CONFIGURACIONES DE PHPAGI.....	- 36 -
3.6.1	CONFIGURACIONES DEL ARCHIVO CITAMEDICA.PHP.....	- 36 -
3.6.2	DESCRIPCIÓN DE LAS FUNCIONES UTILIZADAS	- 44 -
3.6.3	FUNCIONES SOBRE EL PLAN DE MARCADO	- 46 -
3.6.4	DESCRIPCIÓN DE ARCHIVOS DE AUDIO.....	- 47 -
3.7	SISTEMA DE ADMINISTRACION DE LAS CITAS MÉDICAS	- 49 -
3.7.1	DESCRIPCION GENERAL	- 49 -
3.7.2	OPCIONES DEL SISTEMA.....	- 50 -
3.8	CONFIGURACION DEL TELEFONO IP	- 55 -
CAPITULO IVFUNCIONAMIENTOS Y PRUEBAS.....		- 53 -
4.1	INICIO DEL SERVIDOR MYSQL	- 57 -
4.2	INICIO DE ASTERISK.....	- 58 -
4.3	LLAMADA AL SISTEMA DE CITAS MEDICAS.....	- 61 -
4.3.1	PROCESO DEL SISTEMA AL INGRESO DE UNA CEDULA REGISTRADA.....	- 62 -
4.3.2	PROCESO DEL SISTEMA DE UNA CEDULA NO REGISTRADA-	63 -
4.3.3	CONFIRMACIÓN DEL USUARIO	- 64 -
4.3.4	RESPUESTA A UNA FECHA DE CITA NO DISPONIBLE Ó MAL DIGITADA	- 65 -
4.3.5	FIN DE LLAMADA.....	- 65 -

INDICE DE FIGURAS

FIGURA 1 SISTEMA PARA SOLICITAR CITA MÉDICA	- 7 -
FIGURA 2 TELÉFONO DE VOZ SOBRE IP GRANDSTREAM GXP2000	- 22 -
FIGURA 3 INTERFAZ DE INICIO – PHPMYADMIN	- 28 -
FIGURA 4 SOFTPHONE – ZOIPER	- 29 -
FIGURA 5 INTERFAZ GRÁFICA – SISTEMA DE ADMINISTRACIÓN	- 49 -
FIGURA 6 USUARIO Y PASSWORD – SISTEMA DE ADMINISTRACIÓN	- 50 -
FIGURA 7 OPCIONES – SISTEMA DE ADMINISTRACIÓN	- 51 -
FIGURA 8 CONSULTA DE PACIENTE – SISTEMA DE ADMINISTRACIÓN	- 51 -
FIGURA 9 INGRESO DE PACIENTE – SISTEMA DE ADMINISTRACIÓN	- 52 -
FIGURA 10 CONSULTA DE CITA – SISTEMA DE ADMINISTRACIÓN	- 52 -
FIGURA 11 CONSULTA DE PACIENTE – SISTEMA DE ADMINISTRACIÓN	- 53 -
FIGURA 12 CONSULTA DE REGISTRO – SISTEMA DE ADMINISTRACIÓN ...	- 54 -
FIGURA 13 INICIANDO MYSQL – CENTOS	- 57 -
FIGURA 14 INICIO DE ASTERISK – CENTOS	- 58 -
FIGURA 15 MODULE RELOAD –CENTOS	- 59 -
FIGURA 16 EXTENSIONS.CONF – CENTOS	- 59 -
FIGURA 17 SIP.CONF – CENTOS	- 60 -
FIGURA 18 CÉDULA REGISTRADA – ASTERISK	- 62 -
FIGURA 19 CÉDULA NO REGISTRADA – ASTERISK	- 63 -
FIGURA 20 CONFIRMACIÓN – ASTERISK	- 64 -
FIGURA 21 FECHA DE CITA – ASTERISK	- 65 -
FIGURA 22 FIN DE LLAMADA – ASTERISK	- 65 -

INDICE DE TABLAS

TABLA 1 CARACTERÍSTICAS DEL SERVIDOR..... - 21 -

TABLA 2 CARACTERÍSTICAS DEL SERVIDOR – SOFTWARE..... - 23 -

INTRODUCCION

En la actualidad, el avance tecnológico provoca una variedad de cambios en todo el mundo, con el fin de satisfacer en sus necesidades cotidiana al ser humano, en este proyecto se tendrá como objetivo brindar asistencia y facilitar la solicitud de una cita médica evitando de esta forma que el paciente se deba acercar previamente al centro médico para obtener un turno, solo bastará que el usuario realice una llamada en un teléfono digital y siga las instrucciones que indica el menú.

La implementación y el Diseño se fundamenta en un sistema telefónico interactivo en el cual tendrá como objetivo el de gestionar las citas, disminuir las llamadas perdidas como también las llamadas en espera y aumentar la eficacia en la atención, otorgando así una mejora en el servicio.

El proyecto se realiza tomando en consideración el factor económico, donde se justifica la instalación de un sistema operativo LINUX (Open Source) distribución CENTOS en donde se le agrega las funcionalidades de una central telefónica (PBX), software libre llamado ASTERISK, en él se desarrollará la configuración de un IVR (Interactive voice response) que permite exponer un menú sencillo, el cual guiará paso a paso al usuario y así poder acceder a sus diversas opciones con solo presionar uno o varios botones de un teléfono digital y antes de que culmine la llamada el sistema

se encargara de solicitar la confirmación de su cita repitiendo la especialidad médica y la hora asignada, si está de acuerdo con el resultado deberá de presionar un botón con el que inmediatamente se le indica que su cita fue exitosa.

Adicionalmente se realizará un sistema de administración a la Base de Datos del centro médico, para que personal interno de la institución pueda realizar con mayor facilidad consultas, actualizaciones, insertar o cambiar datos del paciente o el de una cita médica.

CAPÍTULO I

ANTECEDENTES Y JUSTIFICACION

1.1 ANTECEDENTES

En el país existen algunos sistemas de comunicación para realizar citas médicas, pero no se han considerado algunas características importantes como es la gran demanda que se refleja diariamente, las necesidades más relevante que existe en cada una de estas instituciones médicas, como también la calidad del servicio que ofrecen a sus pacientes, entre otras.

La solución que el proyecto expone es la utilización de un software OpenSource con el que se implementará una PBX o “centralita telefónica”, llamada ASTERISK, usado por sistemas operativo como Linux, Unix y DOS, que puede trabajar con telefonía analógica o digital, desarrollada en 1999 por Mark Spencer, quien fundó lo que hoy se conoce como Digium empresa que ha desarrollado aplicaciones y tarjetas compatibles con Asterisk, es un software de código abierto que se comporta como una PBX tradicional, es decir es una IP-PBX que permite adaptarse a la mayoría de las tecnologías y protocolos de VoIP y entre sus características más notable está la flexibilidad y la escalabilidad, en donde permite abaratar costos.

Como nueva alternativa que se presenta en este proyecto es donde el paciente pueda acceder a efectuar una cita vía telefónica y que esta quede registrada en un sistema de Database como MySQL, este sistema es relacional, multihilo y multiusuario y cabe recalcar que se

puede ejecutar en cualquier plataforma de Windows, Linux, Unix y más.

En ASTERISK se empleará un sistema conocido como IVR (Interactive Voice Response) el cual ayudara a la atención de llamadas telefónicas, cuya función será indicar al paciente los pasos que debe proceder para realizar su cita con ello lograremos tiempos de respuesta óptimos y atención satisfactoria al cliente las 24 horas del día y los 365 días del año.

1.2 JUSTIFICACIÓN

Debido a una gran demanda del servicio y por motivos económico se ha optado por hacer uso de un Sistema de Software Libre conocido también como Open Source (código abierto), el cual se basa es una solución principal de la red telefónica y del sistema interactivo a implementar en este proyecto.

Voz sobre IP (VoIP) concede transmitir la voz digitalizada por medio de paquetes a través de una red de datos, usando el protocolo IP (Internet Protocol), los cuales pueden ser transportados sin la necesidad de disponer de circuitos conmutados convencionales como lo es la PSTN (Red Telefónica Pública Conmutada), la cual en esta ocasión ayudará con la comunicación de los usuarios al Call Center.

1.3 DESCRIPCIÓN DEL PROYECTO

La implementación de este proyecto tiene como punto de alcance dar el beneficio a los pacientes de un hospital, en el cual se va a tratar de optimizar el tiempo y la gestión administrativa de los procesos. Tener un mayor incremento de la eficiencia del call center y ayudar a reducir los turnos de los operadores. Incrementar las horas de servicio a las 24 horas, de lunes a domingo para dar una mejor atención a los pacientes del hospital. Obtener una Disminución de la tasa de llamadas perdidas y poder así tener el Incremento de la disponibilidad de nuestros operadores.

Como descripción de este proyecto tenemos lo siguiente:

- Para obtener una cita médica el usuario podrá acceder mediante dos medios: el teléfono convencional o de un celular.
- Una vez marcado el número el usuario podrá interactuar con el sistema mediante un menú principal que se fundamenta por ser sencillo a la hora de realizar los pasos.

A continuación se muestra en la Figura 1 un diagrama de flujo de acuerdo a la función que tiene el sistema.

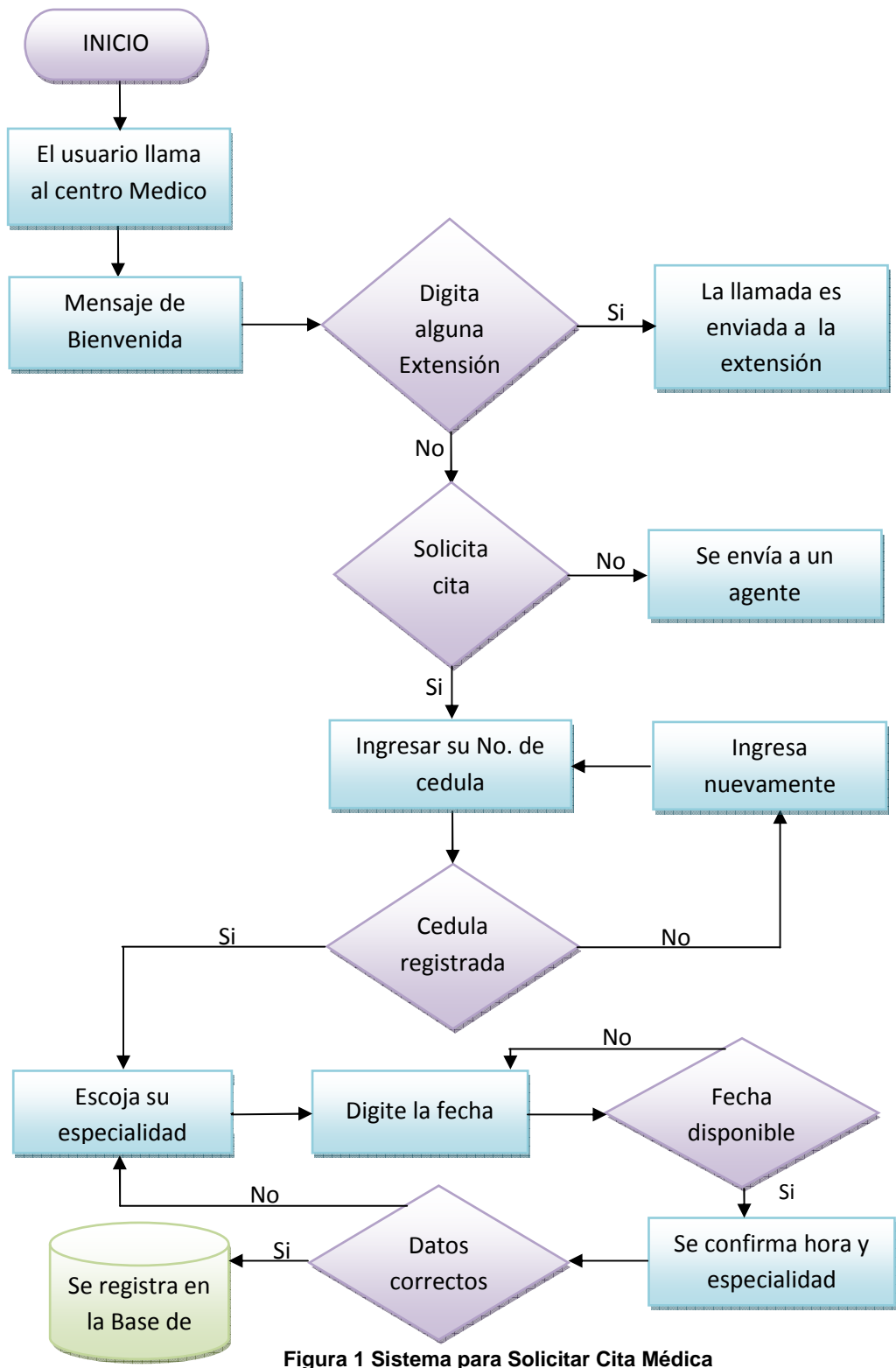


Figura 1 Sistema para Solicitar Cita Médica

1.3.1 OBJETIVO GENERAL

Diseñar e implementar un sistema telefónico interactivo para mejorar el servicio que se le brinda a la comunidad.

1.3.2 OBJETIVOS ESPECÍFICOS

- ✓ Seleccionar el equipo idóneo que proporcione los requerimientos de una forma optima y evitar cualquier tipo de anomalía en la comunicación.
- ✓ Generar un sistema telefónico que permita interactuar con las necesidades del usuario para así, satisfacer la demanda de la comunidad.
- ✓ Realizar un sistema de administración para el personal que labora internamente y así llevar un control más eficiente de los datos del usuario.

1.4 METODOLOGÍA

- Instalación de Centos 5.6
- Instalación de la base de datos MySQL.
- Instalación de php 5
- Instalación de Asterisk

- Configuración de teléfono IP con un usuario SIP
- Implementación del programa que maneja la base de datos
- Diseño e implementación del sistema usando php-agi

1.5 PERFIL DE LA TESIS

En este proyecto como objetivo principal, es mejorar la atención hacia el usuario, haciendo uso de un sistema telefónico interactivo, en el que se basa de un menú fácil de usar, los beneficios que se brinda con este sistema es el de evitar un gasto al usuario, en el que deba asistir anticipadamente a la institución médica en busca de un turno para la cita médica ó que no logre encontrar a alguien que le pueda conceder una atención eficiente.

El desarrollo de este sistema es implementar una solución altamente disponible, disminuir las llamadas perdidas por largas esperas debido al congestionamiento y abarcar con la mayor cantidad de llamadas a menor tiempo.

En el capítulo dos se describe el aspecto teórico del sistema que tendrá la central telefónica como lo es Asterisk.

En el capítulo tres se destaca la implementación que tendría el sistema para llegar a su objetivo.

En el capítulo cuatro finalmente se realizara las pruebas y el funcionamiento de todo lo que se implemento en este sistema.

CAPÍTULO II

ASTERISK Y PHP AGI

2.1 ASTERISK

El proyecto de lo que hoy es Asterisk comenzó en el año de 1999, cuando Mark Spencer decide implementar una centralita telefónica propia, ya que no podía comprar una pues era muy costosa; viendo el éxito funda Linux Support Services a finales de ese mismo año, luego esta empresa en el 2002 se convierte en Digium. Asterisk es uno de los proyectos más exitosos en lo que software libre respecta, su nombre lo lleva porque Linux, Unix y DOS son sistemas operativos que usan el símbolo asterisco en ciertas instrucciones.

Cuenta con un doble licenciamiento GNU/GPL lo cual implica que tenemos las siguientes libertades como son: usarlo con cualquier finalidad, estudiar, alterar y distribuir los cambios realizados en el software completamente; la otra es la licencia propietaria cuyo objetivo es incluir el soporte para el protocolo G.729, el mismo que está sujeto a las limitaciones de una patente, aunque el codificador funciona indistintamente con una u otra versión.

Inicialmente este sistema fue desarrollado para GNU/LINUX, pero ahora se distribuye para diferentes versiones para sistemas operativos como BSD, MacOSX, Solaris y Microsoft.

Este sistema revolucionó el mundo de las telecomunicaciones ya que trata de interconectarlo todo, es decir, se comporta como una PBX (Private Branch Exchange a una Private Automatic Branch Exchange para una PABX), lo cual significa que es cualquier central analógica conectada a la red telefónica pública por medio de líneas troncales para la gestión de llamadas internas, entrantes o salientes con autonomía sobre cualquier otra central telefónica.

Cabe recalcar que es Asterisk un sistema tan versátil debido a que es capaz de trabajar prácticamente con casi todos los estándares de telefonía tradicional y soporta casi a todos los protocolos.

2.1.1 CARACTERÍSTICAS PRINCIPALES

- ✓ Respuesta Automatizada.
- ✓ Bloqueo de llamadas en origen
- ✓ Recuperación de llamadas.
- ✓ Colas de llamadas.
- ✓ Monitoreo de llamadas
- ✓ Listas negras.
- ✓ Autenticación de Usuarios.
- ✓ Desvío de llamadas si la extensión está ocupada o no responde.
- ✓ Transferencia de llamadas.

- ✓ Llamada en espera.
- ✓ Identificación de llamadas.

2.1.2 FUNCIONES BÁSICAS

Puede comportarse como cualquier central tradicional y abarcar todas sus funciones:

- ✓ Conexión con líneas de telefonía tradicional, mediante interfaces FXO para líneas de teléfono fijo o móvil y RDSI con interfaces (BRI o PRI).
- ✓ Soporte con extensiones analógicas para teléfonos analógicos o bien equipos de fax.
- ✓ Soporte de líneas troncales IP: SIP, H323 o IAX.
- ✓ Soporte de extensiones IP: SIP, SCCP, MGCP, H323 o IAX.
- ✓ Música en espera.

2.1.3 FUNCIONES AVANZADAS

Incorpora muchas funcionalidades avanzadas, tales como:

- ✓ Buzón de Voz.- sistema que contesta automáticamente por el usuario.
- ✓ Sistema de Audioconferencias.- permite la conexión remota con diferentes usuarios que desean tener una sesión virtual y

gestiona de manera adecuada el control de los usuarios que se incorporan a ella.

- ✓ IVR.- sistema automático de respuesta que permite redirigir las llamadas entrantes en función de las opciones que señalo el llamante.
- ✓ CDR.- detalle de llamadas realizadas/recibidas por extensión, para obtener control de costos departamentales o de facturación.
- ✓ ACD.- sistema automático de distribución de llamadas entrantes.
- ✓ 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.

2.1.1 FUNCIONES BÁSICA DE USUARIO

- ✓ Transferencia directa o consultiva.
- ✓ Desvíos
- ✓ Capturas (grupo de extensión)

- ✓ Conferencias múltiples
- ✓ Call Parking
- ✓ Llamada directa a cada extensión
- ✓ Callback
- ✓ Paging – Megafonía por medio del altavoz del teléfono

2.1.5 PRINCIPALES VENTAJAS

- ✓ Funcionalidad.- Dispone de funciones de grandes centralitas propietarias (Cisco, Avaya, Alcatel, Siemens, etc) desde las tareas más básicas como desvíos, capturas, transferencias, muti-conferencias, etc. y las más avanzadas como buzones de voz, IVR, CTI, ACD, entre otras.
- ✓ Escalable.- Este sistema puede brindar sus servicios desde 10 usuarios para una empresa pequeña hasta una multinacional en diversas sedes de hasta 10.000 usuarios.
- ✓ Bajo Coste.- No solo porque es Open Source sino por su arquitectura ya que usa un servidor estándar y las tarjetas PCI

para las interfaces de telefonía han bajado sus costos progresivamente.

- ✓ Interoperabilidad.- Pues ha incorporado la mayoría de los estándares tanto los tradicionales como los de telefonía IP.

- ✓ Flexibilidad.- Permite la conexión con redes públicas de telefonía tradicional y se integran fácilmente con centrales tradicionales y centrales IP.

2.2 AGI (ASTERISK GATEWAY INTERFACE)

Es una interfaz que agrega funcionalidad a Asterisk para que interactúe con diversos lenguajes de programación como: Perl, PHP, C, PASCAL, Bourne Shell, etc., y así realizar sistemas más complejos y de mayor utilidad.

- ✓ AGI puede controlar el plan de discado, llamado en `extensions.conf`.

- ✓ EAGI da a la aplicación la posibilidad para tener acceso y para controlar los canales de sonidos además de la interacción con el plan del discado.

- ✓ FastAGI se puede utilizar para hacer el procesamiento en una máquina remota por conexión de red.
- ✓ El DEADAGI da el acceso a un canal muerto después de colgar, deprecated.

Las variables pasadas por asterisk son:

agi_request – El nombre de archivo del código.

agi_channel – El canal de origen (su teléfono).

agi_language – El lenguaje del código (ej. "es").

agi_type – El tipo de canal de origen (ej. "SIP" o "IAX2").

agi_uniqueid - El ID único para la llamada.

agi_version - La versión de Asterisk (desde Asterisk 1.6).

agi_callerid - El número del identificador de llamada (o "unknown" en caso de que sea desconocido).

agi_calleridname – El nombre del identificador de llamada (o "unknown" en caso de que sea desconocido).

agi_callingpres – La presentación para el identificador de llamada en un canal ZAP.

agi_callingani2 – El número que está definido en ANI2. Revisar en la lista detallada de variables (sólo para canales PRI).

agi_callington – El tipo de número usado en canales PRI. Revisar en la lista detallada de variables.

agi_callingtns – un número de 4 dígitos opcional(Selector de Tránsito de Red) usado en canales PRI. Revisar en la lista detallada de variables.

agi_dnid – El id del número marcado (o "unknown" en caso de que sea desconocido).

agi_rdnis – El número referente a DNIS (o "unknown" en caso de que sea desconocido).

agi_context – contexto origen en extensions.conf.

agi_extension – El número llamado.

agi_priority – La prioridad desde la cual fue ejecutado en el plan de marcado.

agi_enhanced – El valor de la bandera es 1.0 si empezó como un código EAGI, 0.0 de otro modo.

agi_accountcode – Código de la cuenta del canal de origen.

agi_threadid – ID del hilo del código agi (desde Asterisk 1.6)

2.3 CLASE PHP AGI

Existe una clase con varias funciones previamente definidas que hacen más fácil la tarea de elaborar códigos PHP para Asterisk. Existen dos versiones de esta clase, la más reciente es la 2.20, pero en este proyecto se hará uso de la 2.14 y lo único que hay que hacer es descargarla, descomprimirla e instalarla (copiarla) dentro del directorio `/var/lib/asterisk/agi-bin/`. Ahora, para poder utilizar las funciones que en esta clase se definen, se debe incluir la clase en el código mediante la sentencia:

```
require "/var/lib/asterisk/agi-bin/phpagi-2.14/phpagi.php";
```

Con estas funciones es posible construir aplicaciones más complejas de forma sencilla. Se recomienda revisar las funciones disponibles dentro de cada clase en la documentación correspondiente dentro de su sitio oficial.

2.4 IVR

Es un Sistema telefónico que nos ayuda interactuar con el ser humano, mediante grabaciones de voz y el reconocimiento de respuestas que no son tan complejas, también se puede mencionar que este sistema es automatizado donde está orientado a la entrega de información por vía telefónica permitiéndonos así acceder a un sin números de servicios, información o alguna otra operación.

En este proyecto se abarca con la satisfacción del usuario en el proceso de solicitar una cita médica, con el sistema IVR se presentará un menú claro y de fácil comprensión, ya que este sistema ayuda a recibir grandes cantidades de llamadas, con la finalidad de así poder reducir el tiempo de espera para los pacientes que requieren ser atendidos.

Además, el Sistema IVR aporta una gran ayuda al poder enrutar llamadas a los agentes o extensiones del centro médico; una función más detallada es la de realizar una llamada al call center del centro médico donde será contestado por el sistema de audio-respuesta y el cual le da a conocer al usuario diferentes acciones u opciones para realizar su solicitud, el usuario al elegir una opción puede encontrarse con más menús o simplemente el sistema realizará su respectivo enrutamiento de la llamada hacia un operador.

IVR tiene la facilidad de atender miles de llamadas en un día, permitiendo así una consulta periódica y sin mayores problemas, se pondrá implementar de esta manera la atención a las 24 horas al día y los 365 días del año con el fin de aprovechar esta tecnología que mejora la atención a los pacientes en las entidades públicas.

CAPÍTULO III

IMPLEMENTACION

3.1 INTRODUCCION

En este capítulo queremos retomar las funcionalidades que tiene el software libre ya que permite bajar los costos de implementación para dar solución a distintas problemáticas que existen actualmente en las comunicaciones.

Cabe indicar que Asterisk es uno de los proyectos de software libre más exitosos, ya que se pueden hacer cambios a la plataforma para adaptarlo a las necesidades requeridas, con lo cual, se puede decir que es un sistema escalable y se aplica a cualquier estándar y a casi todos los protocolos; debido a esto podemos decir que las centrales telefónicas tradicionales quedarían a un segundo plano, ya que serían reemplazadas por un computador.

El computador que reemplazaría a la central telefónica analógica mediría los costos que estos varían de acuerdo con el tipo de hardware que es requerido para soportar la gran concurrencia de llamadas telefónicas que atendería nuestro sistema telefónico, pero estos gastos siempre serán menores que implementar una central telefónica tradicional.

3.2 HARDWARE

Al momento de seleccionar las características de las herramientas basadas en hardware esto, solo dependerá de las funcionalidades a la que se le va a integrar al sistema, para así llegar a un correcto desarrollo y ambiente eficiente con sus requerimientos.

3.2.1 SERVIDOR

Es una computadora donde se incorpora el sistema PBX Asterisk, el mismo que permite el proceso de las llamadas de los usuarios a la clínica.

A continuación se muestra en la tabla 1 las características utilizadas para la implementación de este proyecto:

Procesador	Intel Pentium 4 de 3GHZ
Memoria RAM	2048 MB DDR 400 RAM
Disco Duro	250GB
Tarjeta de Red	Fastethernet 10/100 Mbps

Tabla 1 Características del Servidor

3.2.2 TELÉFONO IP

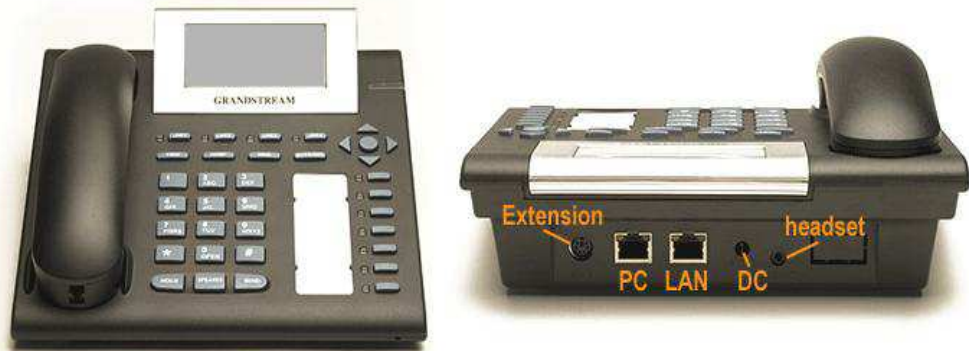


Figura 2 Teléfono de Voz sobre IP Grandstream GXP2000

Basado en los estándares abiertos de la industria (compatible con Asterisk), identificador de llamada, llamada en espera, transferencia de llamada, servidor web para la administración y configuración, registro de llamadas, cancelación de eco (G.168), asegura una calidad de audio de alta fidelidad, 2 puertos Ethernet 10/100 Mbps, del cual se utiliza conectores RJ-45 para conectar el teléfono a la computadora y a la red, tal como se muestra en la Figura 2.

3.3 SOFTWARE

3.3.1 SERVIDOR

En la tabla 2 se detalla los sistemas y las aplicaciones instalados en el servidor para este proyecto:

Sistema Operativo	Linux
Distribución	Centos 5.6
Arquitectura	X86
Software IP PBX	Asterisk Versión 1.4
Base de Datos	MySQL
Software Adicional	PHPMYADMIN 2.11

Tabla 2 Características del Servidor – Software

3.3.2 INSTALACIÓN DE ASTERISK

Se necesita instalar algunas dependencias en el servidor, previamente a la instalación de Asterisk, a continuación se detalla los pasos:

1. Descargar los paquetes en /usr/src

✓ wget-c <http://downloads.digium.com/pub/asterisk/asterisk-1.4-current.tar.gz>

✓ wget-c <http://downloads.digium.com/pub/asterisk/asterisk-addons-1.4-current.tar.gz>

✓ wget-c <http://downloads.digium.com/pub/telephony/dahdi-tools/dahdi-tools-current.tar.gz>

✓ wget -c <http://downloads.digium.com/pub/telephony/dahdi-linux/dahdi-linux-current.tar.gz>

✓ `wget -c http://downloads.digium.com/pub/libpri/libpri-1.4-current.tar.gz`

2. Deberá ser descomprimido en `/usr/src/`

✓ `tar -xvzf asterisk-1.4-current.tar.gz`

✓ `tar -xvzf asterisk-addons-1.4-current.tar.gz`

3. Instalarlas librerías que son indispensables para la correcta ejecución de la PBX Asterisk

✓ `yum install kernel`

✓ `yum install kernel-devel`

✓ `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++

4. Instalar las dependencias y soporte para DB:

```
yum -y install mysql-server mysql-devel newt-devel  
unixODBC unixODBC-devel libtool-ltdl libtool-ltdl-devel mysql-  
connector-odbc
```

En el caso de tener problemas de “sources for kernel”:

✓ yum -y install kernel-devel

✓ yum -y update kernel

✓ uname -r

5. Ejecutar el siguiente grupo de comandos:

✓ cd ../libpri-1.4.10.2, make, make install

✓ cd ../asterisk-1.4.30, make clean, ./configure, make

✓ cd ../asterisk-addons-1.4.11, make clean, ./configure

✓ cd dahdi-linux-current, make, install

- ✓ `cd dahdi-tools --current, ./configure, make, make install, make config.`

3.3.3 INSTALACIÓN DE MYSQL

Para efectuar la instalación de MySQL en Asterisk se debe realizar los siguientes pasos:

1. Verificar e instalar MySQL:

- ✓ `yum install mysql* unixODBC-devel -y` o sino:
- ✓ `yum -y install mysql-server mysql-devel newt-devel unixODBC unixODBC-devel libtool-ltdl libtool-ltdl-devel mysql-connector-odbc`

2. Levantar MySQL y asignar password al usuario root:

- ✓ `service mysqld start mysqladmin -u <root> password <contraseña_de_root>`

3. Ingresar al directorio donde se encuentran las fuentes de asterisk add-ons e instalar el soporte para mysql:

- ✓ `cd /usr/src/asterisk-addons-`
- ✓ `./configure`
- ✓ `make menuselect`
- ✓ `make`

- ✓ make install

3.3.4 INSTALACIÓN DE PHP

Se instala el compilador php versión 5, para ejecutar los códigos en cgi realizados con lenguaje de programación php

1. Para instalar php se debe ejecutar el siguiente comando:

- ✓ yum install php*

2. Para usar el archivo que utilizará el cgi debemos cambiar los permisos de ejecución:

- ✓ chmod 777 <ruta_absoluta_o_nombre del archivo.cgi>

3. Para ejecutar un código escrito en php en la consola se ubica la ruta donde se encuentra alojado el archivo o se especifica la ruta absoluta en el comando:

- ✓ !/usr/bin/php -q <archivo.cgi>

4. Descargar y ubicar la librería phpagi de la versión 2.14 en la ruta /var/lib/asterisk/cgi-bin de esta manera podremos ejecutar los script creados.

3.3.5 INSTALACIÓN DEL PHPMYADMIN

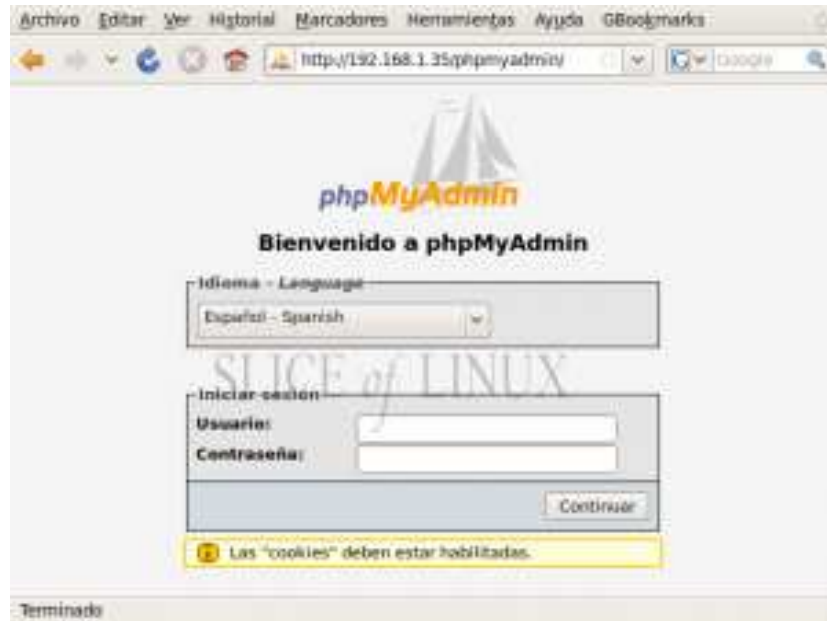


Figura 3 Interfaz de inicio – phpMyAdmin

Esta herramienta permite la administración de base de datos MySQL, haciendo uso de un navegador, en el se puede crear o eliminar base de datos; crear, eliminar o alterar tablas, eliminar, editar o agregar campos, y además también la de ejecutar consultas SQL, su interfaz es como se muestra en la Figura 3.

Entre las características principales se destaca por ser open source y multilinguaje en donde la actualidad cuenta con 50 idiomas, otra de las características está la de ser multiplataforma.

Para su instalación se procede a detallar los pasos:

1. Ingresar en el navegador la siguiente dirección que tiene como referencia [1], en el cual encontraremos el paquete phpmyadmin-2.11.11-1.el5.rf.noarch.rpm.
2. Se procede a descomprimir e instalar el archivo en la siguiente ruta:

/usr/share

3.3.6 SOFTPHONE



Figura 4 Softphone – Zoiper

Comunicador IAX y SIP, softphone Zoiper es una herramienta de comunicación convergente de Internet con alta calidad de voz. Se ejecuta en Windows, Mac OS X, Linux y Solaris, como se muestra en la Figura 4.

3.4 BASE DE DATOS

Este sistema contiene una base de datos implementada en MySQL, la misma que se compone de 6 tablas:

1. Cita
2. Paciente
3. Registro
4. Usuarios
5. Especialidad
6. Horaturn

En el cual el paciente tendrá como objetivo solicitar su cita médica en la especialidad que escoja y con su respectivo turno automáticamente asignado mediante la fecha seleccionada previamente, cuyos datos se encuentran localizados en las tablas.

3.4.1 INGRESO A MYSQL

Para ingresar a MySQL podemos hacerlo de dos maneras:

1. La primera es por la consola de Linux en el que se ingresa el siguiente comando:

✓ `mysql -u root -p labtelecom10`

2. La segunda es por vía browser en donde se hace uso de la herramienta de administración “phpMyAdmin”

✓ Ingrese al navegador (ej. Mozilla Firefox) escribiendo la siguiente dirección `http://localhost/phpmyadmin`

✓ En la ventana principal se debe de ingresar el user y el password.

3.4.2 CREACIÓN DE LA BASE DE DATOS

El nombre de la base de datos es “db_cita” y emplearemos el siguiente comando para crear la base de datos:

✓ `create database db_cita`

Para la creación de las tablas:

create table paciente (

idpaciente int(2) NOTNULL auto_increment

cedula varchar(10) NOTNULL

nombres varchar(20) NOTNULL

apellidop varchar(20) NOTNULL

apellidom varchar(20) NOTNULL

telefono varchar(9) NOTNULL

```

edad      int(2)      NOTNULL
sexo      varchar(1) NOTNULL
numcarp   int(4)      NOTNULL
primary key (`idpaciente`)
) ENGINE=INNODB DEFAULT CHARSET= utf8 COLLATE=utf8_unicode_ci

```

```

create table cita (
idcita      int(2) NOTNULL,
fechacita   date NOTNULL,
t_idturno   int(2) NOTNULL
especialidad int(2) NOTNULL
primary key (`idcita`),
) ENGINE=INNODB DEFAULT CHARSET= utf8 COLLATE=utf8_unicode_ci

```

```

create table registro(
idregistro  int(2)  NOTNULL
fecharegistro datetime NOTNULL
idpaciente  int(2)  NOTNULL
Idcita      int(2)  NOTNULL
primary key (`idregistro`),
foreign key (`idpaciente`), references `paciente` (`idpaciente`) on delete no action on
update no action
foreign key (`idcita`), references `cita` (`idcita`) on delete no action on update no
action
) ENGINE=INNODB DEFAULT CHARSET= utf8 COLLATE=utf8_unicode_ci

```

```

create table `especialidad` (
`ides` int(2) NOT NULL AUTO_INCREMENT,
`descripcion` varchar(30) COLLATE utf8_unicode_ci NOT NULL,

```

```
PRIMARY KEY (`ides`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=3;
```

```
create table `horaturn`  
  `idhora` int(2) NOT NULL AUTO_INCREMENT,  
  `hora` varchar(5) COLLATE utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`idhora`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=3;
```

```
create table `usuarios` (  
  `id` int(4) NOT NULL AUTO_INCREMENT,  
  `usuario` varchar(15) COLLATE utf8_unicode_ci NOT NULL,  
  `password` varchar(15) COLLATE utf8_unicode_ci NOT NULL,  
  `tipo` varchar(8) COLLATE utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=3;
```

3.5 CONFIGURACIONES DE ASTERISK

3.5.1 CONFIGURACIÓN DEL ARCHIVO SIP.CONF

Este archivo de configuración se encuentra localizado en la ruta `/etc/asterisk` y sirve para registrar los usuarios SIP con las extensiones que usaran la PBX, el mismo se encuentra estructurado en dos bloques, la parte general donde se definen

las extensiones, y el bloque final donde configuramos nuestras extensiones internas y externas.

```
[general]
rtcachefriends=yes
rtupdate=no
rtautoclear=yes
srvlookup=yes
disallow=all
allow=alaw
allow=ulaw
allow=gsm
language=es
calltransfer=no
callforwarding=no
qualify=yes
```

```
[2442268]
type=friend
secret=2442268
qualify=yes
nat=no
host=dynamic
canreinvite=no
context=internal
```

3.5.2 CONFIGURACIÓN DEL ARCHIVO EXTENSIONS.CONF

Está ubicado en la ruta `/etc/asterisk`, Este archivo comprende en el plan de marcado y es el archivo más importante de Asterisk, su función principal es invocar al código donde se encuentra almacenado el programa principal, aquí encontramos dos bloques de código el contexto general y el internal:

- ✓ El general es donde establecemos los parámetros generales del plan de marcado.
- ✓ El contexto internal establece el procedimiento a realizarse con alguna extensión en el interior de la PBX.

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

```
[globals]
ANDRES=SIP/801
```

```
[internal]
exten=>2442268,1,set(CHANNEL(language)=es)
exten=>2442268,2,background(bienvenida0)
exten=>2442268,3,Wait(4)
exten=>1,1,Goto(agente,801,1)
exten=>2,1,Goto(cita,701,1)
exten=>2442268,4,hangup
```

```
exten=>555,1,Record(despedida%d.gsm)
exten=>555,2,playback(${RECORDED_FILE})
exten=>555,3,hangup()
```

```
exten=>100,1,playback(especialidad3)
exten=>100,2,hangup()
```

```
[cita]
exten=>701,1,AGI(citasmedicas.php)
exten=>701,2,hangup
```

```
[agente]
exten=>801,1,Dial(${ANDRES},10,r)
exten=>801,2,playback(vm-nobodyavail)
exten=>801,3,hangup
exten=>801,102,playback(tt-allbusy)
exten=>801,103,hangup
```


3.6 CONFIGURACIONES DE PHPAGI

3.6.1 CONFIGURACIONES DEL ARCHIVO CITAMEDICA.PHP

```
#!/usr/bin/php -q
<?php
set_time_limit(30);
error_reporting(E_ALL);
require("phpagi/phpagi.php");

//conexión a la base de datos
$con = mysql_connect ('localhost', 'root', 'labtelecom10') or die
('Error: no se puede conectar a la BD');
mysql_select_db('db_cita',$con)or die ( mysql_error());

$agi = new AGI();
$agi->answer();

$flag=1;
do{

//se pide la cédula al paciente
$result=$agi->get_data("cedula0",3000,1|10); //-----corregir
audio
$nummarcado=$result["result"];

//si digita 9 sale del sistema
if($nummarcado=='9'){
break;
}else{

//caso contrario se valida la cédula en la base de datos
if(strlen($nummarcado)==10){
$query = mysql_query("select idpaciente from paciente where
cedula='".$nummarcado.'",$con) or die ('ERROR en consulta
a la BD');

$re = mysql_fetch_assoc($query);
$idpaciente=$re["idpaciente"];
```

```

if($idpaciente!=""){

//se pide al paciente que escoja la especialidad
$result = $agi->get_data("especialidad0",3000,1);
$digmarcado =$result['result'];

if($digmarcado!=""){

//aqui solo se cambia el nombre a la variable
$idespecialidad = $digmarcado;

if($idespecialidad!=""){

//se pide que digite la fecha
$agi->exec(background,"fecha1");

$result = $agi->get_data("ano0", 6000,4);
$var1 = $result['result'];

$result = $agi->get_data("mes0", 6000,2);
$var2 = $result['result'];

$result = $agi->get_data("dia0", 6000,2);
$var3 = $result['result'];

$fecha =$var1."-". $var2."-".$var3;

//se valida la fecha para que no sea ni sábado ni domingo
$dias =
array('Lunes','Martes','Miercoles','Jueves','Viernes','Sabado','D
omingo'); $dia_semana = $dias[date('N', strtotime($fecha))-1];

$fech_actual=strftime("%Y-%m-%d", time());

if($dia_semana!='Sabado' &&
$dia_semana!='Domingo'&&$fecha>$fech_actual){

//se escoge el máximo turno de acuerdo a la fecha y la
especialidad

```

```

$result_max = mysql_query("select max(t_idturno) as maximo
from cita where fechacita=".$fecha."and
especialidad=".$idespecialidad."" , $con) or die ("no select");

$row_maximo = mysql_fetch_assoc($result_max);

//para cuando es el primer turno
if($row_maximo["maximo"]!=""){

//para cuando ya existe turno y debe de ser menor a 22
if($row_maximo["maximo"]<22){

$result2 = mysql_query("select t_idturno from cita where
fechacita=".$fecha."and especialidad=".$idespecialidad." and
t_idturno=".$row_maximo["maximo"], $con);
$result2 = mysql_fetch_assoc($result2);
$t_idtur=$result2["t_idturno"];
$t_idturn=$t_idtur["t_idturn"]+1;

//indica la hora
switch($t_idturn){
case 2: $cad="hora2"; break;
case 3: $cad="hora3"; break;
case 4: $cad="hora4"; break;
case 5: $cad="hora5"; break;
case 6: $cad="hora6"; break;
case 7: $cad="hora7"; break;
case 8: $cad="hora8"; break;
case 9: $cad="hora9"; break;
case 10: $cad="hora10"; break;
case 11: $cad="hora11"; break;
case 12: $cad="hora12"; break;
case 13: $cad="hora13"; break;
case 14: $cad="hora14"; break;
case 15: $cad="hora15"; break;
case 16: $cad="hora16"; break;
case 17: $cad="hora17"; break;
case 18: $cad="hora18"; break;
case 19: $cad="hora19"; break;
case 20: $cad="hora20"; break;
case 21: $cad="hora21"; break;
case 22: $cad="hora22"; break;}

```

```

//indica la especialidad
switch($idespecialidad){
case 1: $doc="especialidad1"; break;
case 2: $doc="especialidad2"; break;
case 3: $doc="especialidad3"; break;}
//se le indica los datos que ingreso hora y especialidad
$agi->exec(Playback,"suhora0");
$agi->exec(Playback,$cad);
$agi->exec(Playback,"suespecialidad0");
$agi->exec(Playback,$doc);

//se pide confirmacion de hora y especialidad
$result = $agi->get_data("confirmacion0", 7000,1);
$confir=$result["result"];

//cuando se digita 5 se inserta en la base de datos si digita
cero o nada simplemente no se guarda
if($confir==5){

$sql=mysql_query("INSERT into
cita(fechacita,t_idturno,especialidad)
values('.$fecha.', '$t_idturn.', '$idespecialidad. ')", $con);

$scit=mysql_query("select idcita from cita where
fechacita='.$fecha.'"and t_idturno='.$t_idturn.'" and
especialidad='.$idespecialidad.'" , $con) or die ('no se
selecciona');
$numcit1= mysql_fetch_assoc($scit);
$idcit1=$numcit1["idcita"];

$fecha_actual1=strftime( "%Y-%m-%d %H:%M:%S", time() );

$sql2=mysql_query("INSERT into
registro(fecharegistro,idpaciente,idcita)
values('.$fecha_actual1.', '$idpaciente.', '$idcit1. ')", $con)
or die ('no se registra');
$agi->exec(Playback,"suregistro0");
$flag = 0;
}
}else{$result=$agi->get_data("otrafecha0",6000,1);
$flag=$result['result'];
}

```

```

//aquí empieza la función "do"
do {

$agi->exec(Playback,"fecha1");

$result = $agi->get_data("ano0", 6000,4);
$var1 = $result['result'];

$result = $agi->get_data("mes0", 6000,2);
$var2 = $result['result'];

$result = $agi->get_data("dia0", 6000,2);
$var3 = $result['result'];

$fecha =$var1."-". $var2."-".$var3;

$dias =
array('Lunes','Martes','Miercoles','Jueves','Viernes','Sabado','D
omingo'); $dia_semana = $dias[date('N', strtotime($fecha))-1];

if($dia_semana!='Sabado' && $dia_semana!='Domingo'){

$result_max = mysql_query("select max(t_idturno) as maximo
from cita where fechacita=".$fecha."and
especialidad=".$idespecialidad.""", $con) or die ("no select");

$row_maximo = mysql_fetch_assoc($result_max);

if($row_maximo["maximo"]!=""){
if($row_maximo["maximo"]<22){

$result2 = mysql_query("select t_idturno from cita where
fechacita=".$fecha."and especialidad=".$idespecialidad." and
t_idturno=".$row_maximo["maximo"], $con);

$re2 = mysql_fetch_assoc($result2);

$t_idtur=$re2["t_idturno"];
$t_idturn=$t_idtur["t_idturn"]+1;

switch($t_idturn){
case 2: $cad="hora2"; break;
case 3: $cad="hora3"; break;
case 4: $cad="hora4"; break;

```

```

case 5: $cad="hora5"; break;
case 6: $cad="hora6"; break;
case 7: $cad="hora7"; break;
case 8: $cad="hora8"; break;
case 9: $cad="hora9"; break;
case 10: $cad="hora10"; break;
case 11: $cad="hora11"; break;
case 12: $cad="hora12"; break;
case 13: $cad="hora13"; break;
case 14: $cad="hora14"; break;
case 15: $cad="hora15"; break;
case 16: $cad="hora16"; break;
case 17: $cad="hora17"; break;
case 18: $cad="hora18"; break;
case 19: $cad="hora19"; break;
case 20: $cad="hora20"; break;
case 21: $cad="hora21"; break;
case 22: $cad="hora22"; break;}
switch($idespecialidad){
case 1: $doc="especialidad1"; break;
case 2: $doc="especialidad2"; break;
case 3: $doc="especialidad3"; break;}

```

```

$agi->exec(Playback,"suhora0");
$agi->exec(Playback,$cad);
$agi->exec(Playback,"suespecialidad0");
$agi->exec(Playback,$doc);

```

```

$result = $agi->get_data("confirmacion0", 7000,1);
$confir=$result['result'];

```

```

if($confir==5){

```

```

$sql=mysql_query("INSERT into
cita(fechacita,t_idturno,especialidad)
values('".$fecha."','".$t_idturno."','".$idespecialidad."')",$con);

```

```

$cit=mysql_query("select idcita from cita where
fechacita='".$fecha.'"and t_idturno='".$t_idturno.'" and
especialidad='".$idespecialidad.'" , $con) or die ('no se
selecciona');

```

```

$numcit1= mysql_fetch_assoc($cit);
$idcit1=$numcit1["idcita"];

```

```

$fecha_actual1=strftime( "%Y-%m-%d %H:%M:%S", time() );

$sql2=mysql_query("INSERT into
registro(fecharegistro,idpaciente,idcita)
values(".$fecha_actual1.", ".$idpaciente.", ".$idcit1."),$con)
or die ('no se registra');
$agi->exec(Playback, "suregistro0");
break;}break;
}else{$result=$agi->get_data("otrafecha0",6000,1);
$repetirfech=$result['result'];//repetir fecha y darle opción 1
para seguir ó 0 para salir
}
}else{$t_idturno ='1';

switch($t_idturno){
case 1: $cad1="hora1";break;}

switch($idespecialidad){
case 1: $doc="especialidad1"; break;
case 2: $doc="especialidad2"; break;
case 3: $doc="especialidad3"; break;}

$agi->exec(Playback, "suhora0");
$agi->exec(Playback,$cad1);
$agi->exec(Playback, "suespecialidad0");
$agi->exec(Playback,$doc);

$result = $agi->get_data("confirmacion0", 7000,1);
$confir=$result['result'];

if($confir==5){

$sql2=mysql_query("INSERT into
cita(fechacita,t_idturno,especialidad)
values(".$fecha.", '1', ".$idespecialidad."),$con)or die ('no
agrega');;
$cit1=mysql_query("select idcita from cita where
fechacita=".$fecha."and t_idturno='1' and
especialidad=".$idespecialidad.""", $con) or die ('no
selecciona');
$numcit = mysql_fetch_assoc($cit1);
$idcit=$numcit["idcita"];
$fecha_actual=strftime( "%Y-%m-%d %H:%M:%S", time() );

```

```

$sql3=mysql_query("INSERT into
registro(fecharegistro,idpaciente,idcita)
values(".$fecha_actual.", ".$idpaciente.", ".$idcit."),$con) or
die ('no se registra');
$agi->exec(Playback, "suregistro0");
break;}break;}
}else{$result=$agi->get_data("otrafecha0",6000,1);
$repetirfech=$result['result'];
}
}while($repetirfech!='0'); $flag=0; //aqui termina el "do" usando
la funcion "while"
}
}else{ $t_idturno ='1';

switch($t_idturno){
case 1: $cad="hora1";break;}

switch($idespecialidad){
case 1: $doc="especialidad1"; break;
case 2: $doc="especialidad2"; break;
case 3: $doc="especialidad3"; break;}

$agi->exec(Playback, "suhora0");
$agi->exec(Playback,$cad);
$agi->exec(Playback, "suespecialidad0");
$agi->exec(Playback,$doc);

$result = $agi->get_data("confirmación0", 7000,1);
$confir1=$result["result"];

if($confir1==5){
$sql2=mysql_query("INSERT into
cita(fechacita,t_idturno,especialidad)
values(".$fecha.", '1', ".$idespecialidad."),$con)or die ('no
agrega');;
$cit1=mysql_query("select idcita from cita where
fechacita=".$fecha."and t_idturno='1' and
especialidad=".$idespecialidad."" , $con) or die ('no
selecciona');
$numcit = mysql_fetch_assoc($cit1);
$idcit=$numcit["idcita"];
$fecha_actual=strftime( "%Y-%m-%d %H:%M:%S", time() );
$sql3=mysql_query("INSERT into
registro(fecharegistro,idpaciente,idcita)

```



```

values(".$fecha_actual.", ".$idpaciente.", ".$idcit.")", $con) or
die ('no se registra');
$agi->exec(Playback, "suregistro0");
    $flag = 0;}}
} else{ $result=$agi->get_data("otrafecha0", 6000, 1);
$flag=$result['result'];
}
}
}
}
}

}while($flag);
//audio de la despedida y cuelga
$agi->exec(Playback, "despedida0");
$agi->hangup();
?>

```

3.6.2 DESCRIPCIÓN DE LAS FUNCIONES UTILIZADAS

La librería phpagi-2.14 se encuentra en la ruta /var/lib/Asterisk/agi-bin y es la que permite realizar diferentes operaciones sobre el plan de marcado y además sobre la base de datos. A continuación se indica algunas funciones que nos proporciona:

```
mysql_connect ('localhost', 'root', 'labtelecom10')
```

Función que nos permite conectar con la base de datos, el primer parámetro indica la ubicación de la base de datos, el segundo parámetro indica el nombre del usuario al cual

pertenece la base de datos y el tercer parámetro indica la clave del usuario.

```
mysql_select_db('db_cita',$con)or die ( mysql_error());
```

Función que selecciona la base de datos con la cual vamos a trabajar, el primer parámetro indica el nombre de la base de datos y el segundo parámetro indica el nombre del puntero que tiene la conexión hacia la base de datos.

```
mysql_query("select idpaciente from paciente where cedula=".$nummarcado.", $con)or die ( mysql_error());
```

Función que ejecuta una consulta, el primer parámetro es una cadena de caracteres que contiene la petición a ser ejecutada y el segundo parámetro indica el nombre del puntero que tiene la conexión hacia la base de datos.

```
$re = mysql_fetch_assoc($result)
```

Función que devuelve el resultado de una fila asociada de acuerdo a la petición ejecutada.

3.6.3 FUNCIONES SOBRE EL PLAN DE MERCADO

```
error_reporting(E_ALL)
```

Crea un log con todos los errores ocurridos durante la ejecución del código.

```
$agi = new AGI()
```

Crea una nueva instancia de la clase agi

```
require ('phpagi/phpagi.php')
```

Permite que las funciones en la librería phpagi sean usadas en el código principal.

```
$agi->exec(Playback,"ArchivodeAudio")
```

Reproduce una grabación.

```
$agi->get_data("ArchivodeAudio", tiempoEspera,MaxDigitos);
```

Captura la extensión marcada luego de reproducirse un archivo de audio dentro de un tiempo de espera, el tercer parámetro indica la cantidad máxima de dígitos que se pueden ingresar.

```
$agi->hangup()
```

Termina o cuelga la llamada.

3.6.4 DESCRIPCIÓN DE ARCHIVOS DE AUDIO

El audio es grabado marcando la extensión 555 y para poder escucharlo se marca la extensión 100, estos archivos son almacenados en /var/lib/asterisk/sounds/ en formato gsm.

Bienvenida: “Usted se ha comunicado con el centro médico esperanza, si conoce el numero de la extensión a la que desee comunicarse dígtelo ahora, caso contrario presione 1 para comunicarse con un agente o 2 para solicitar una cita”

Cedula: “Estimado paciente por favor digite su número de cedula o digite 9 para salir del sistema”

Especialidad: “Escoja su especialidad médica a la que desee asistir, 1 Medicina General, 2 Cardiología, 3 Neurología”

Fecha: “Ahora debe ingresar la fecha en la que desee asistir a su cita, deberá de hacerlo en el siguiente formato ejemplo: si su fecha es el 15 de marzo del 2011 cuando le soliciten el año digitará dos, cero, uno, uno, el mes cero, cinco, el día quince, verifique que su fecha no sean los días sábados o domingo porque no será válida”

Año: “Digite el año”

Mes: "Digite el mes"

Día:"Digite el día"

Hora1:"9 de la mañana"

Hora2:"9 y 15 de la mañana"

Hora3:"hasta la Hora22: "3 y 45 de la tarde"

Especialidad1: "Medicina General"

Especialidad2:"Cardiología"

Especialidad3: "Neurología"

Suhora: "su hora es"

Suespecialidad: "Su especialidad es"

Confirmación: "Si está de acuerdo por favor digite 5 sino digite 0 para salir"

Suregistro: "Su registro fue exitoso"

Otrafecha: "La fecha ingresada no está disponible o verifique que no sea una fecha anterior a la de hoy, si desea volver a intentar con otra fecha digite 1 sino digite 0 para salir"

Despedida:"Agradecemos su llamada"

3.7 SISTEMA DE ADMINISTRACION DE LAS CITAS MÉDICAS

Este sistema de administración es realizado con el objetivo de poder ayudar en el proceso de las citas médicas que realiza a cada paciente que está registrado o quizás requiera ser registrado en la base de datos de la clínica, además cabe mencionar que su interfaz es muy sencilla para un fácil manejo el cual será utilizado constantemente por el personal interno de la clínica.

3.7.1 DESCRIPCION GENERAL

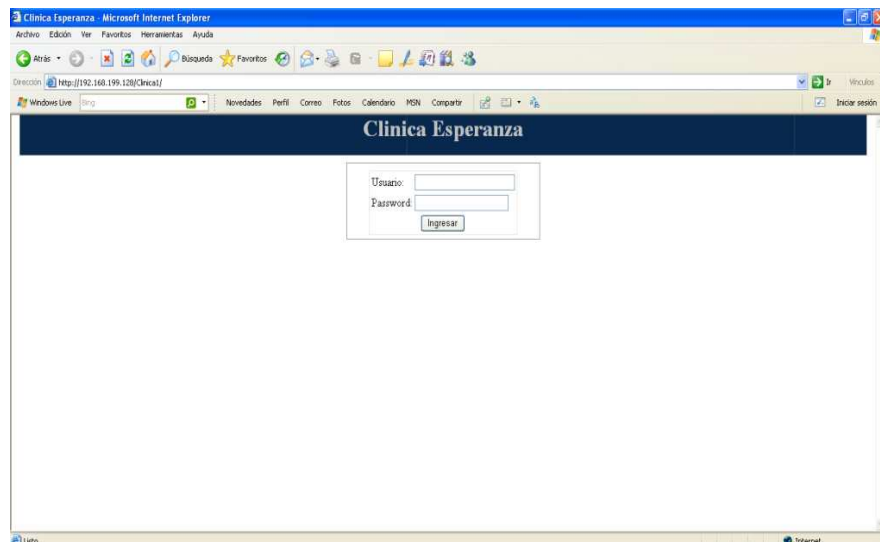


Figura 5 Interfaz Gráfica – Sistema de Administración

Para acceder al sistema se debe validar los datos de usuario y password como se muestra en la figura 5, un vez validando se

permite identificar al personal que necesita ingresar al sistema, este podría ser un médico de la institución o un agente que pertenece al call center, recuerden que este sistema será usado internamente, por lo que quizás se deba aumentar datos de usuario y password a medida que ingrese personal a laboral en la clínica o como también puede ser el caso de eliminar datos por existir algún cambio de personal

3.7.2 OPCIONES DEL SISTEMA

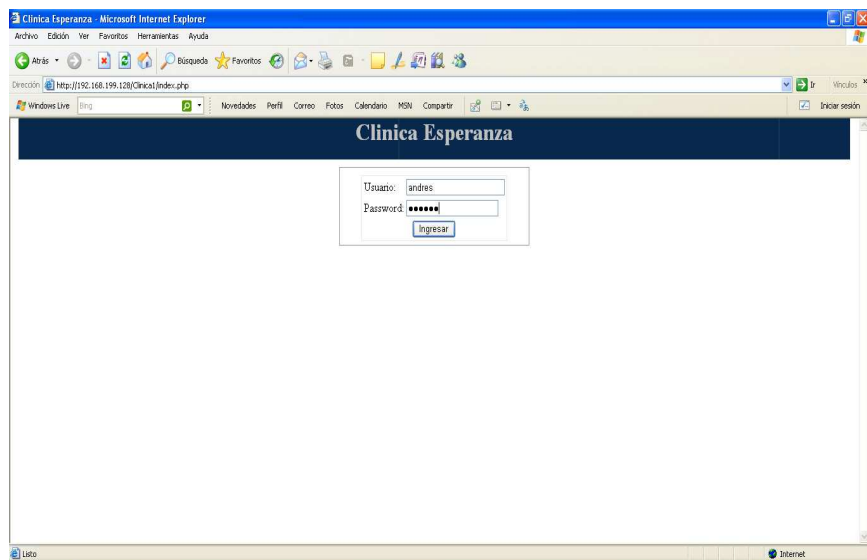


Figura 6 Usuario y Password – Sistema de Administración

Para hacer uso de este sistema de administración se valida usuario y contraseña los cuales se ingresan en el campo respectivo, como se muestra en la Figura 6.

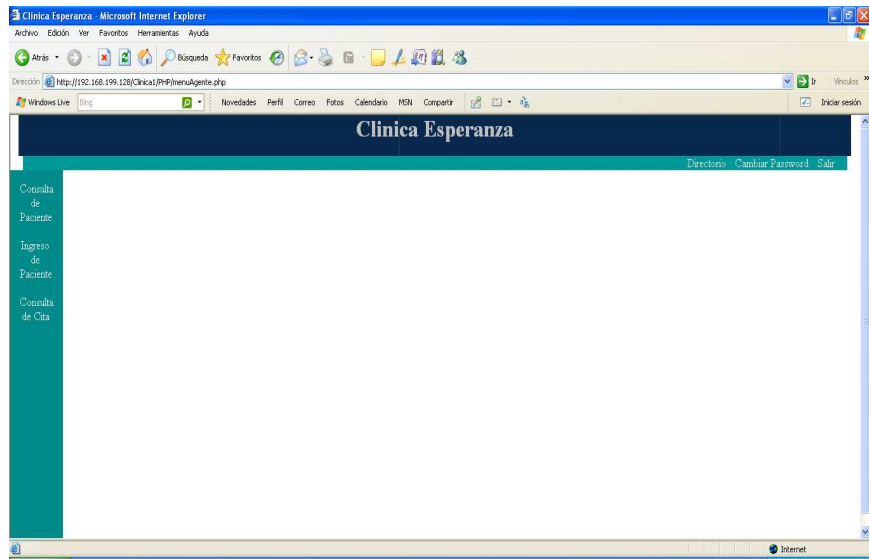


Figura7 Opciones – Sistema de Administración

Las opciones para los usuarios de tipo agente se muestra en la Figura 7, dependiendo de las opciones que el paciente solicite, se procedera a realizarlo en el sistema.

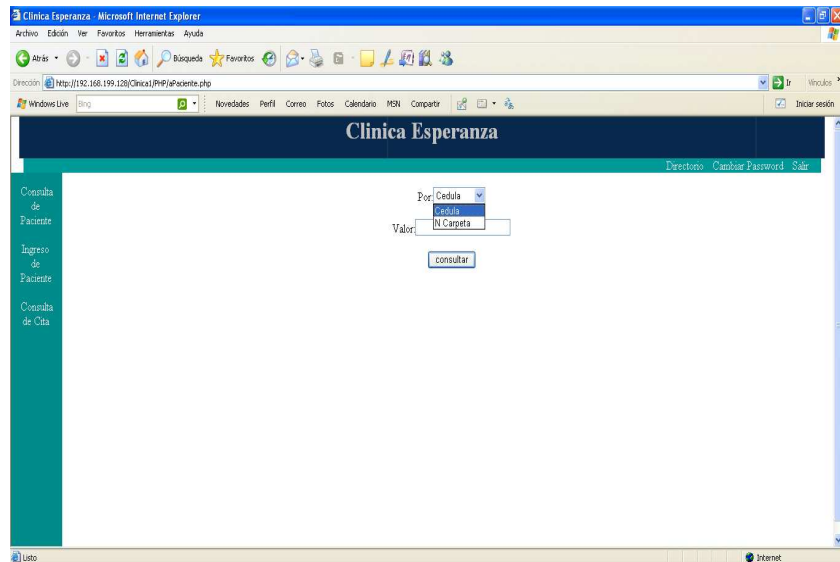


Figura 8 Consulta de Paciente – Sistema de Administración

Entre las opciones del agente está la de consulta de paciente como se muestra en la Figura 8, ahí se presentará la

información detallada del paciente realizando la búsqueda por medio del No. de cédula o el No. de carpeta y adicionalmente se podrá editar o borrar.

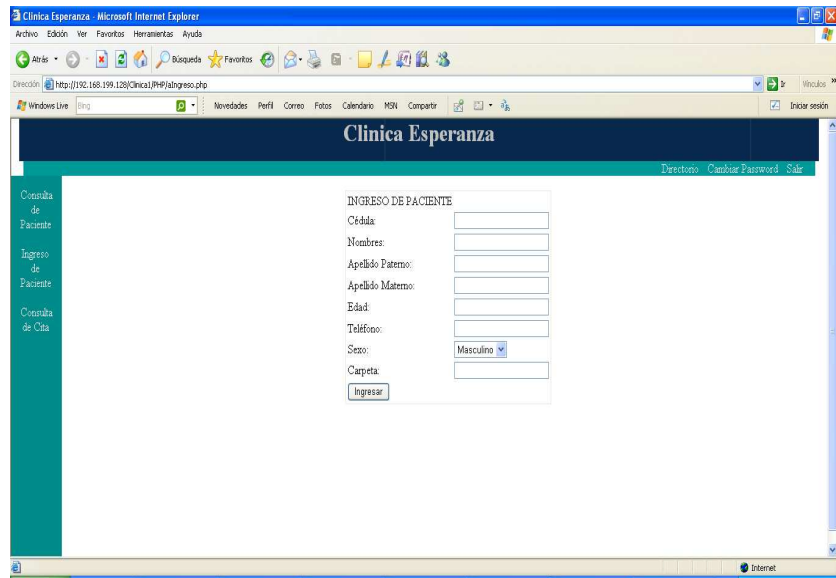


Figura 9 Ingreso de Paciente – Sistema de Administración

La opción ingreso del paciente, Como se muestra en la Figura 9, se procederá agregar la información de un nuevo paciente.

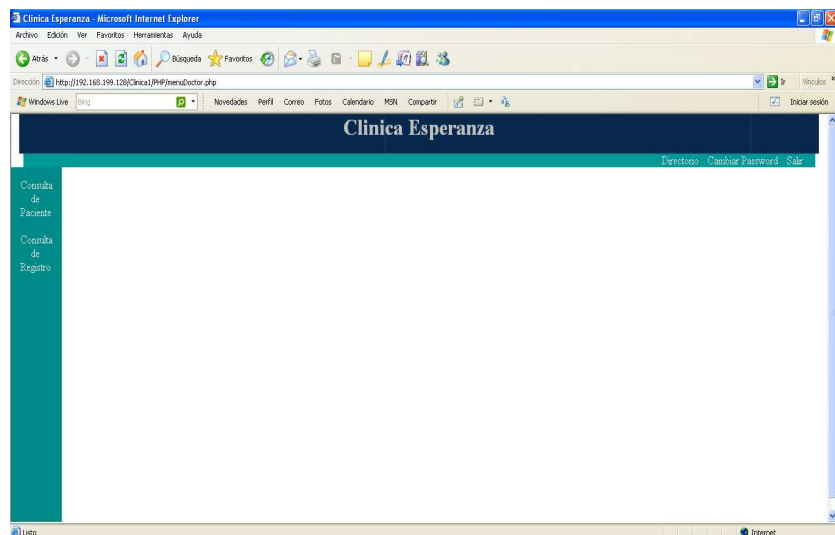


Figura 10 Consulta de Cita – Sistema de Administración

En La opción consulta de cita como se muestra en la Figura 10, se ayudará al paciente con la consulta o el recordatorio de su cita. El agente deberá hacerlo por medio del número de cédula o el número de la carpeta.

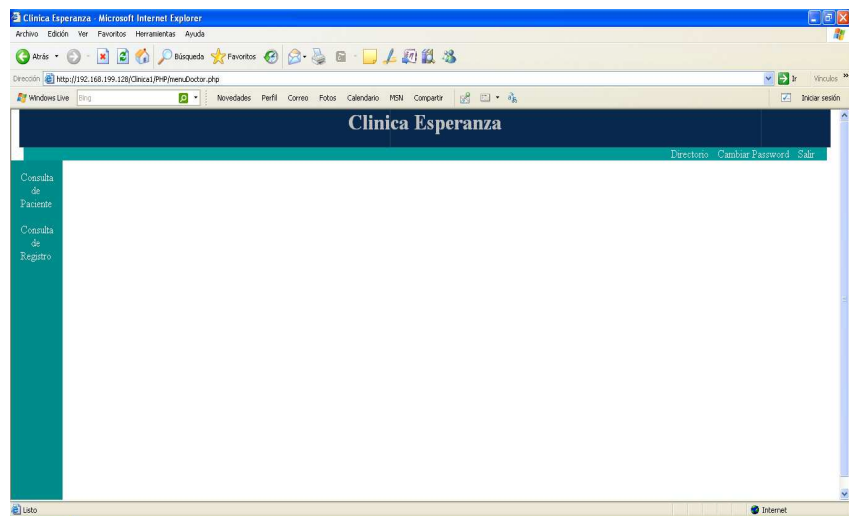


Figura 11 Consulta de Paciente – Sistema de Administración

Al ingresar como usuario tipo Doctor sus opciones son similares a las del agente, observe la Figura 11, pero con la diferencia que en la opción de consulta de paciente no podrá borrar ni editar como lo hace el usuario tipo agente.

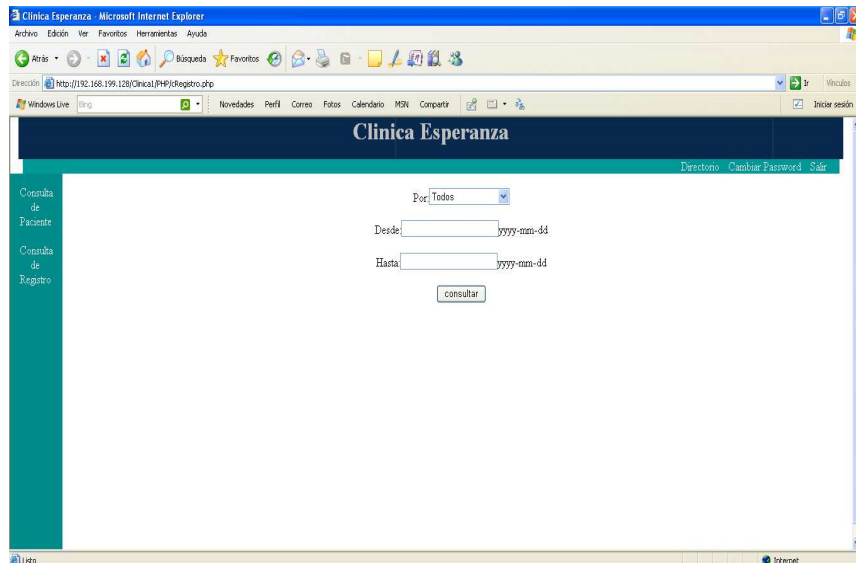


Figura 12 Consulta de Registro – Sistema de Administración

En la opción consulta de registro se conseguirá observar un registro con la información detallada de los pacientes para las citas almacenadas en la base de datos, para esto deberá escoger las opciones que se muestra en la Figura 12, realizando un filtrado de los datos por especialidad e intervalo de fecha a la que se desee consultar, caso contrario al dar click en la opción consultar se mostrara todas las citas con las información de cada paciente y en todas las fechas.

3.8 CONFIGURACIÓN DEL TELÉFONO IP

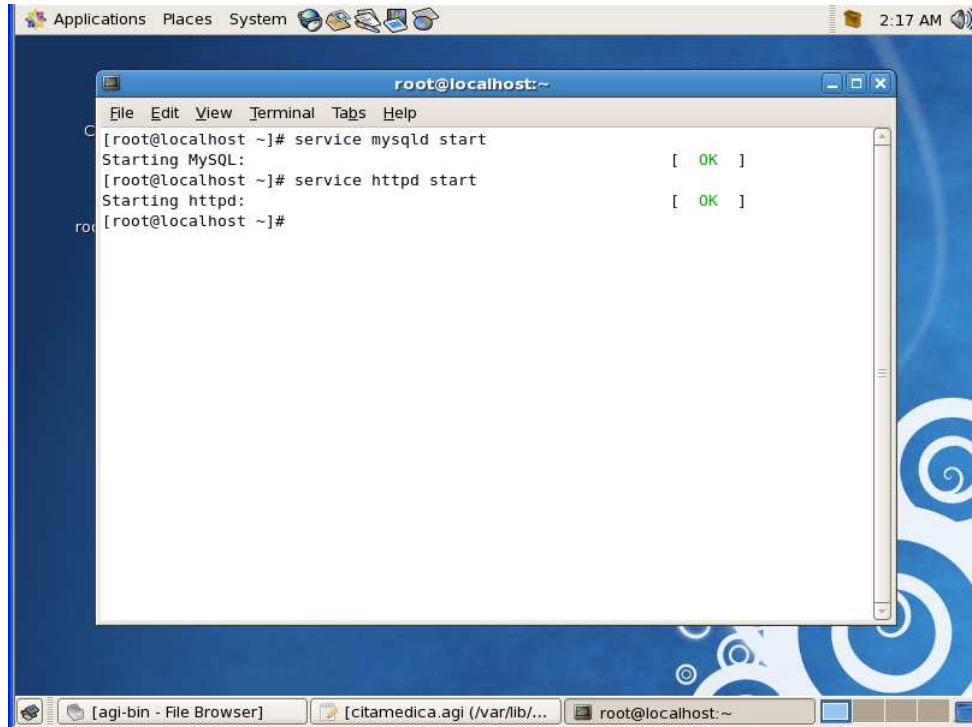
Para configurar el teléfono IP con los usuarios SIP y poder realizar llamadas, ingresamos a un navegador de internet colocando la dirección IP del teléfono, ingresamos usuario y clave del administrador luego configuramos el teléfono con los usuarios SIP ya configurados en el archivo sip.conf, a continuación mostraremos la configuración:

- ✓ Account Name: 2442268
- ✓ SIP Server: 2442268
- ✓ Outbound proxy: dirección IP del servidor de Asterisk
- ✓ SIP User ID: 244226
- ✓ Authenticate ID: 2442268
- ✓ Authenticate Password: 2442268
- ✓ Name: 2442268

CAPITULO IV

FUNCIONAMIENTOS Y PRUEBAS

4.1 INICIO DEL SERVIDOR MYSQL

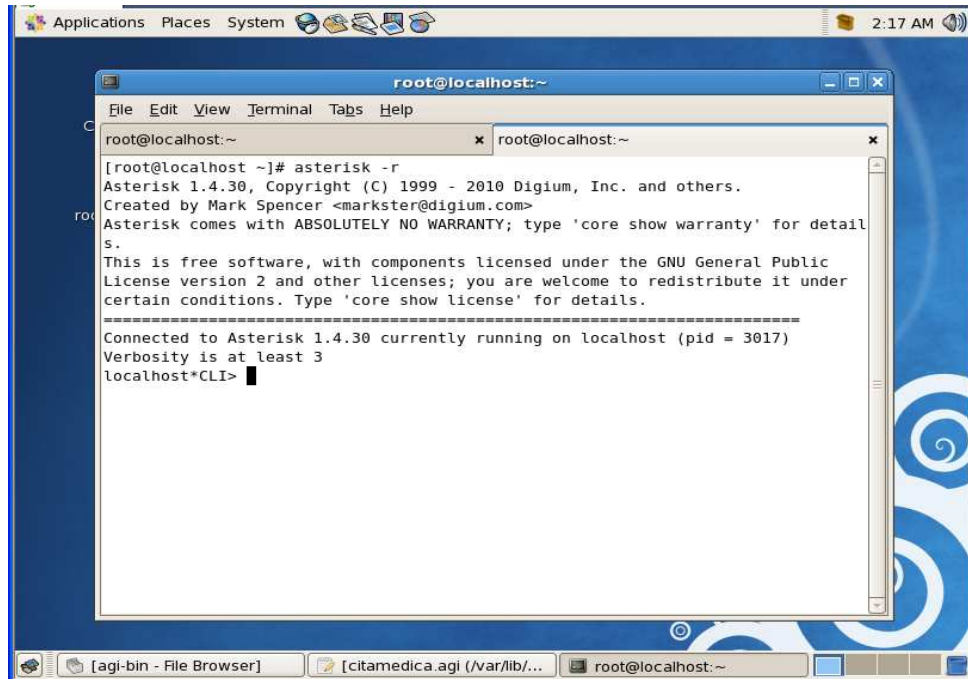


```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service mysqld start  
Starting MySQL: [ OK ]  
[root@localhost ~]# service httpd start  
Starting httpd: [ OK ]  
[root@localhost ~]#
```

Figura 13 Iniciando MySQL – CENTOS

Para poder hacer uso de la base de datos donde se tiene toda la información de los pacientes y el personal de la institución médica se debe previamente ingresar el comando que permita iniciar el servicio MySQL, tal como se muestra en la Figura 13.

4.2 INICIO DE ASTERISK



```
root@localhost:~]# asterisk -r
Asterisk 1.4.30, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
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 1.4.30 currently running on localhost (pid = 3017)
Verbosity is at least 3
localhost*CLI>
```

Figura 14 Inicio de Asterisk – CENTOS

Para poder iniciar el sistema principal de este proyecto se hace uso del comando Asterisk – r y deberá de presentarse como lo muestra la Figura 14.

Después de que Asterisk se inicio se deberá cargar todos los módulos donde están sus respectivas configuraciones para el uso de este sistema como se podrá observar en la Figura 15.

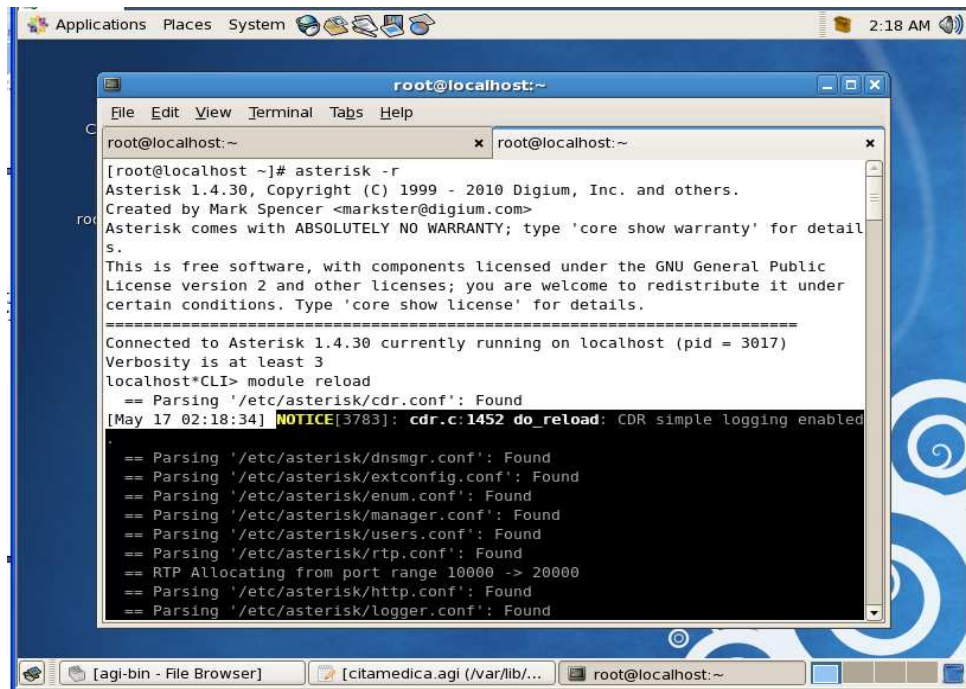


Figura 15 Module reload –CENTOS

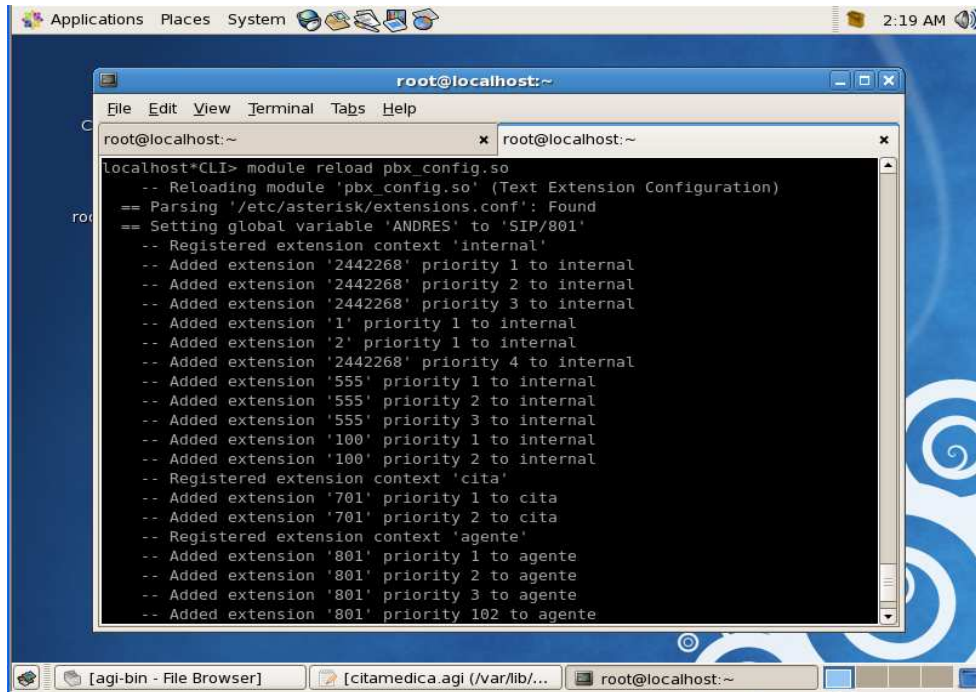
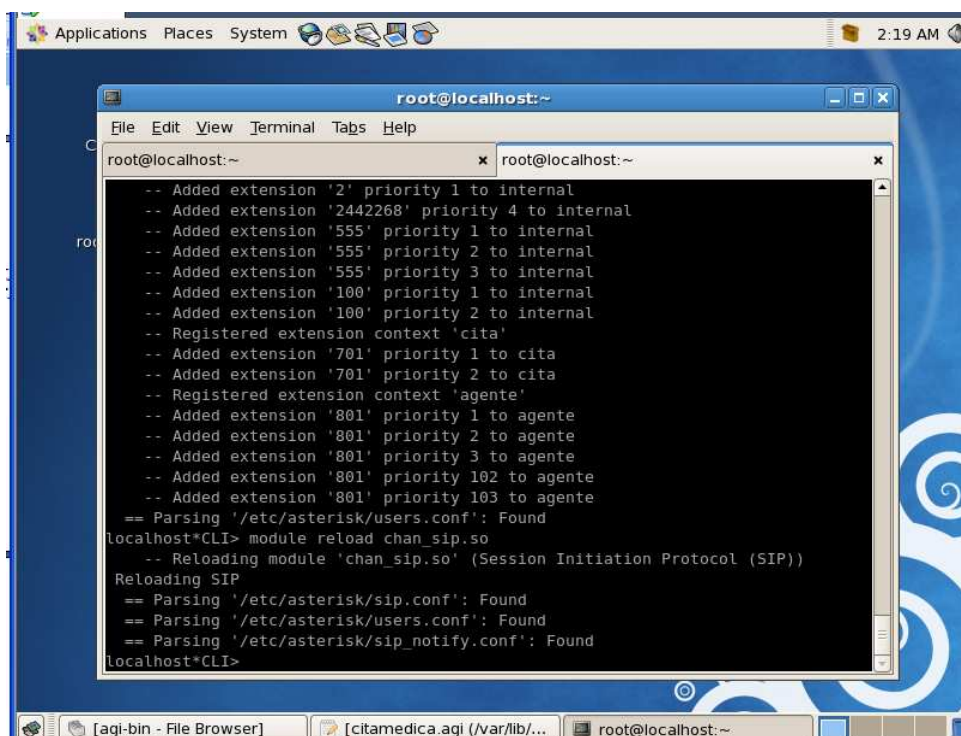


Figura 16 Extensions.conf – CENTOS

Se debe saber que cuando se realice cambios en el archivo extensions.conf y este sea reconocido por el sistema se ejecuta el comando que se muestra en la Figura 16, adicional a esto también será el mismo caso para el archivo sip.conf con los comandos que se muestra en la Figura 17



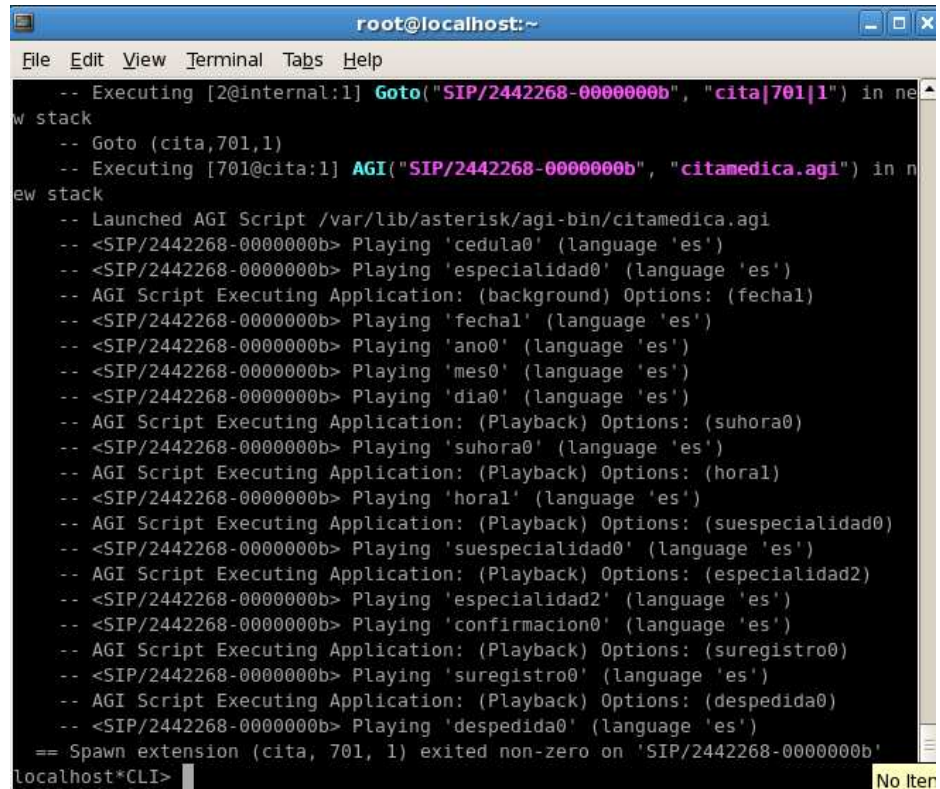
```
root@localhost:~  
File Edit View Terminal Tabs Help  
root@localhost:~ x root@localhost:~  
-- Added extension '2' priority 1 to internal  
-- Added extension '2442268' priority 4 to internal  
-- Added extension '555' priority 1 to internal  
-- Added extension '555' priority 2 to internal  
-- Added extension '555' priority 3 to internal  
-- Added extension '100' priority 1 to internal  
-- Added extension '100' priority 2 to internal  
-- Registered extension context 'cita'  
-- Added extension '701' priority 1 to cita  
-- Added extension '701' priority 2 to cita  
-- Registered extension context 'agente'  
-- Added extension '801' priority 1 to agente  
-- Added extension '801' priority 2 to agente  
-- Added extension '801' priority 3 to agente  
-- Added extension '801' priority 102 to agente  
-- Added extension '801' priority 103 to agente  
== Parsing '/etc/asterisk/users.conf': Found  
localhost*CLI> module reload chan_sip.so  
-- Reloading module 'chan_sip.so' (Session Initiation Protocol (SIP))  
Reloading SIP  
== Parsing '/etc/asterisk/sip.conf': Found  
== Parsing '/etc/asterisk/users.conf': Found  
== Parsing '/etc/asterisk/sip_notify.conf': Found  
localhost*CLI>
```

Figura 17 sip.conf – CENTOS

4.3 LLAMADA AL SISTEMA DE CITAS MEDICAS

El Paciente realizara la llamada al centro médico al número de teléfono 2442268, al marcar este número escuchara un mensaje de bienvenida seguido con opciones que deberá digitar para poder continuar con su petición, si el paciente digita 1, la llamada es transferida a un agente quien se encargará de ayudar al paciente a: consultar su cita, ingresar o actualizar información personal etc., si digita la opción 2 la llamada pasa inmediatamente a la solicitud de cita en el que debe ingresar su número cédula, si no está registrada deberá volver a ingresar nuevamente caso contrario deberá volver a llamar y escoger la opción 1 para que un agente lo ayude, pero si está registrada se le presenta un menú donde escogerá la especialidad a la que desee asistir, continuando con el proceso se le pide digitar la fecha en un formato indicado previamente, permitiendo así asignarle una turno automáticamente escogido por el sistema todo esto dependede que si la fecha ingresa es correcta o esté disponible, finalmente se repetirá los datos para confirmar la hora y la especialidad, si está de acuerdo digitara 5 para ser almacenada a la base de datos del centro médico, caso contrario la llamada será colgada y la información descartada.

4.3.1 PROCESO DEL SISTEMA AL INGRESO DE UNA CÉDULA REGISTRADA



```
root@localhost:~
File Edit View Terminal Tabs Help
-- Executing [2@internal:1] Goto("SIP/2442268-0000000b", "cita|701|1") in new
w stack
-- Goto (cita,701,1)
-- Executing [701@cita:1] AGI("SIP/2442268-0000000b", "citamedica.agi") in n
ew stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/citamedica.agi
-- <SIP/2442268-0000000b> Playing 'cedula0' (language 'es')
-- <SIP/2442268-0000000b> Playing 'especialidad0' (language 'es')
-- AGI Script Executing Application: (background) Options: (fechal)
-- <SIP/2442268-0000000b> Playing 'fechal' (language 'es')
-- <SIP/2442268-0000000b> Playing 'ano0' (language 'es')
-- <SIP/2442268-0000000b> Playing 'mes0' (language 'es')
-- <SIP/2442268-0000000b> Playing 'dia0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (suhora0)
-- <SIP/2442268-0000000b> Playing 'suhora0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (horal)
-- <SIP/2442268-0000000b> Playing 'horal' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (suespecialidad0)
-- <SIP/2442268-0000000b> Playing 'suespecialidad0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (especialidad2)
-- <SIP/2442268-0000000b> Playing 'especialidad2' (language 'es')
-- <SIP/2442268-0000000b> Playing 'confirmacion0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (suregistro0)
-- <SIP/2442268-0000000b> Playing 'suregistro0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (despedida0)
-- <SIP/2442268-0000000b> Playing 'despedida0' (language 'es')
== Spawn extension (cita, 701, 1) exited non-zero on 'SIP/2442268-0000000b'
localhost*CLI>
```

Figura 18 Cédula Registrada – Asterisk

Se realiza la búsqueda de la cédula registrada en la base de datos y reproduce la grabación para poder proceder con el siguiente paso como el de escoger especialidad, digitar fecha y confirmar, como se muestra en la Figura 18

4.3.2 PROCESO DEL SISTEMA DE UNA CÉDULA NO REGISTRADA

```
-- Executing [2442268@internal:1] Set("SIP/2442268-00000010", "CHANNEL(Language)=es") in new stack
-- Executing [2442268@internal:2] Background("SIP/2442268-00000010", "bienvenida0") in new stack
-- <SIP/2442268-00000010> Playing 'bienvenida0' (language 'es')
== CDR updated on SIP/2442268-00000010
-- Executing [2@internal:1] Goto("SIP/2442268-00000010", "cita|701|1") in new stack
-- Goto (cita,701,1)
-- Executing [701@cita:1] AGI("SIP/2442268-00000010", "citamedica.agi") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/citamedica.agi
-- <SIP/2442268-00000010> Playing 'cedula0' (language 'es')
-- <SIP/2442268-00000010> Playing 'cedula0' (language 'es')
-- <SIP/2442268-00000010> Playing 'cedula0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (despedida0)
-- <SIP/2442268-00000010> Playing 'despedida0' (language 'es')
== Spawn extension (cita, 701, 1) exited non-zero on 'SIP/2442268-00000010'
localhost*CLI>
```

Figura 19 Cédula No Registrada – Asterisk

Si al no encontrar la cédula en la base de datos esta reproduce nuevamente la grabación indicándole que ingrese su número de cédula o digite 9 para salir del sistema, tal como se muestra en la Figura 19

4.3.3 CONFIRMACIÓN DEL USUARIO

```
-- AGI Script Executing Application: (Playback) Options: (suhora0)
-- <SIP/2442268-0000000b> Playing 'suhora0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (horal)
-- <SIP/2442268-0000000b> Playing 'horal' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (suespecialidad0)
-- <SIP/2442268-0000000b> Playing 'suespecialidad0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (especialidad2)
-- <SIP/2442268-0000000b> Playing 'especialidad2' (language 'es')
-- <SIP/2442268-0000000b> Playing 'confirmacion0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (suregistro0)
-- <SIP/2442268-0000000b> Playing 'suregistro0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (despedida0)
-- <SIP/2442268-0000000b> Playing 'despedida0' (language 'es')
== Spawn extension (cita, 701, 1) exited non-zero on 'SIP/2442268-0000000b'
```

Figura 20 Confirmación – Asterisk

Se repite los datos ingresado en el transcurso de la llamada a lo que respecta sólo a la especialidad escogida y la hora asignada, sí el paciente está de acuerdo con el resultado, se le confirmara al sistema al digitar 5, luego se reproduce la grabación indicando que su registro fue exitoso y se procede a colgar la llamada. Como se muestra en la Figura 20

El paciente deberá obligatoriamente digitar el 5 para que su información pueda ser almacenada en la base de datos de la clínica caso contrario pasara a ser descartada automáticamente.

4.3.4 RESPUESTA A UNA FECHA DE CITA NO DISPONIBLE Ó MAL DIGITADA

```
-- Executing [2442268@internal:1] Set("SIP/2442268-00000012", "CHANNEL(language)=es") in new stack
-- Executing [2442268@internal:2] Background("SIP/2442268-00000012", "bienvenida0") in new stack
-- <SIP/2442268-00000012> Playing 'bienvenida0' (language 'es')
== CDR updated on SIP/2442268-00000012
-- Executing [2@internal:1] Goto("SIP/2442268-00000012", "cita|701|1") in new stack
-- Goto (cita,701,1)
-- Executing [701@cita:1] AGI("SIP/2442268-00000012", "citamedica.agi") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/citamedica.agi
-- <SIP/2442268-00000012> Playing 'cedula0' (language 'es')
-- <SIP/2442268-00000012> Playing 'especialidad0' (language 'es')
-- AGI Script Executing Application: (background) Options: (fechal)
-- <SIP/2442268-00000012> Playing 'fechal' (language 'es')
-- <SIP/2442268-00000012> Playing 'ano0' (language 'es')
-- <SIP/2442268-00000012> Playing 'mes0' (language 'es')
-- <SIP/2442268-00000012> Playing 'dia0' (language 'es')
-- <SIP/2442268-00000012> Playing 'otrafecha0' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (despedida0)
-- <SIP/2442268-00000012> Playing 'despedida0' (language 'es')
== Spawn extension (cita, 701, 1) exited non-zero on 'SIP/2442268-00000012'
localhost*CLI>
```

Figura 21 Fecha de Cita – Asterisk

Si la fecha es anterior a la de hoy o no se encuentra disponible se le volverá a solicitar que ingrese una nueva fecha, pero ingresando nuevamente su cédula y especialidad, caso contrario digitará cero para poder salir del sistema. Como se muestra en la Figura 21.

4.3.5 FIN DE LLAMADA

```
-- AGI Script Executing Application: (Playback) Options: (despedida0)
-- <SIP/2442268-0000000b> Playing 'despedida0' (language 'es')
== Spawn extension (cita, 701, 1) exited non-zero on 'SIP/2442268-0000000b'
localhost*CLI>
```

Figura 22 Fin de llamada – Asterisk

La llamada finaliza en los casos donde el paciente digita la opción de salir del sistema, o si confirma sus datos ingresados en que ambos caso se reproduce una grabación de despedida. Véase Figura 22

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Con el uso de la interfaz de desarrollo de Asterisk AGI, las librerías phpagi facilita en gran parte el desarrollo de aplicaciones complejas, permitiendo de esta manera trabajar en conjunto con el motor de base de datos tan utilizado como lo es MySQL.
2. La implementación de un sistema telefónico con el fin de solicitar citas médicas de una manera sencilla y que pueda solventar a las necesidades del paciente, es fácil de realizar con las funciones para crear IVR's que nos ofrece Asterisk.
3. La instalación de software libre provee diferentes herramientas y librerías, con un tiempo menor de proceso lo que permite una implementación de sistemas de gran utilidad, eficacia, eficiencia, adaptables, escalables, confiables y a un bajo costo.
4. Para la administración de la base de datos que maneja el sistema de citas médicas se necesita desarrollar una aplicación que permita poder ejecutar todas las operaciones sobre ella.

RECOMENDACIONES

1. Usar el Sistema Operativo Linux de la distribución de CENTOS por tener las características de ser estable, seguro y dedicado a servidores.
2. Iniciar el servicio de MySQL, como administrador para así poder acceder a los datos e informaciones del sistema.
3. Realizar la descarga del software phpMyAdmin para la creación o eliminación de tablas en la base de datos por medio del navegador, esto será muy útil para personas no expertas en los comandos de la Base de Datos.
4. Realizar actualizaciones del sistema operativo Centos para un correcto funcionamiento.

GLOSARIO DE TERMINOS

Asterisk.-Software de código abierto, liberado bajo licencia GPL. Soporta todas las funcionalidades de las centralitas tradicionales/IP.

Add-on.- Pequeños programas opcionales que solo funcionan anexados a otros y que sirven para aumentar o complementar sus funcionalidades.

AGI.-AGI es una interfaz de gateway para el Asterisk similar al CGI usado por los servidores Web. Este permite el uso de diferentes lenguajes de programación como el Perl, PHP, C a su elección. AGI es llamado a partir del plan de discado definido en extensions.conf.

Dialplan.- Un dial plan (plan de marcado) es un software residente en algunos gateways y en casi todas las IP-PBX que permite determinar el tratamiento que debe darse a un número discado.

Gateway.- Puerta de enlace, acceso, pasarela, nodo en una red informática que sirve de punto de acceso a otra red.

CLI.-(Línea de comandos, Intérprete de comandos, Terminal, Consola, Shell, CLI, Command line interface). Tipo de interfaz para manipular un programa o

sistema operativo con instrucciones escritas. Cada instrucción es escrita en una línea de texto y suelen ejecutarse al presionar ENTER. También se suele permitir archivos scripts para la ejecución automática de varias líneas de comandos que cumplen alguna función. Si bien se siguen usando en algunos ámbitos, este tipo de interfaz ha evolucionado hacia los GUI, que son interfaces gráficas. Sistemas operativos de la familia DOS son ejemplos de uso de interfaz CLI

CentOS.-Community ENTERprise Operating System.CentOS es una distribución de Linux gratuita que está basada en la distribución Red Hat Enterprise Linux (RHEL). CentOS es muy similar al RHEL, pero gratuito, aunque no es mantenido por Red Hat.

Digital.-Cualquier señal o modo de transmisión que utiliza valores discretos en lugar de un espectro continuo de valores (como las señales analógicas). Los valores pueden medir voltaje, frecuencia, amplitud, ubicación, etc. En informática se suele utilizar el sistema digital de unos y ceros (sistema binario) para transmitir, procesar o almacenar información. Por ejemplo, el reloj del microprocesador trabaja en dos voltajes distintos, cada uno representa un uno o un cero. Con la combinación de unos y ceros se puede procesar todo tipo de información.

Analógico.-Se refiere a las magnitudes o valores que varían con el tiempo en forma continua (distancia, temperatura, velocidad, voltaje, frecuencia, amplitud, etc.) y pueden representarse en forma de ondas. Las computadoras emplean lo digital y, por lo tanto, si entra información analógica, se debe convertir; este es el caso de la conexión a Internet por Dial up, donde un módem convierte la señal analógica (el sonido) en digital.

Digium.-Digium, empresa que promueve el Asterisk, invierte en ambos aspectos, el desenvolvimiento de código fuente y en hardware de telefonía de bajo costo que funciona con Asterisk.

Hardware.-En computación, término inglés que hace referencia a cualquier componente físico tecnológico, que trabaja o interactúa de algún modo con la computadora. No sólo incluye elementos internos como el disco duro, CD-ROM, sino que además hace referencia al cableado, circuitos, etc. E incluso hace referencia a elementos externos como la impresora, el mouse, el teclado, el monitor y otros periféricos. El hardware contrasta con el software, que es intangible y le da lógica al hardware (además de ejecutarse dentro de éste). El hardware no es frecuentemente cambiado, en tanto el software puede ser creado, borrado y modificado sencillamente. (Excepto el firmware, que es un tipo de software que raramente es alterado).

Internet.-Conocida como la red de redes, pues se trata de una de las redes más grandes. Para su función utiliza el conjunto de protocolos TCP/IP. Desde que fue creada la WWW, el número de usuarios no paró de crecer; pero ese no es el único servicio de internet: podemos acceder remotamente a otras máquinas (telnet y SSH), transferir archivos (FTP), conversar con personas (chat y mensajeros), servicio de correo electrónico (email), grupos de noticias, etc.

IP.-Protocolo para la comunicación en una red a través de paquetes conmutados, es principalmente usado en Internet. Los datos se envían en bloques conocidos como paquetes (datagramas) de un determinado tamaño (MTU). El envío es no fiable (conocido también como best effort o mejor esfuerzo); se llama así porque el protocolo IP no garantiza si un paquete alcanza o no su destino correctamente. Un paquete puede llegar dañado, repetido, en otro orden o no llegar. Para la fiabilidad se utiliza el protocolo TCP de la capa de transporte. Los paquetes poseen una cabecera con información sobre la máquina de origen y la de destino (sus direcciones IP), con esta información los enrutadores determinan por dónde enviar la información. Cada paquete de un mismo archivo puede enviarse por diferentes rutas dependiendo de la congestión del momento.

Java.- Lenguaje de programación orientado a objetos. Fue desarrollado por James Gosling y sus compañeros de Sun Microsystems al principio de la década de los 90. La programación en Java es compilada en bytecode, el cuál es ejecutado por la máquina virtual Java. Usualmente se usa un compilador JIT. El lenguaje es parecido a C y C++, aunque su modelo de objetos es más sencillo, y fue influenciado también por Smalltalk, y Eiffel.

Kernel.-Núcleo. Parte esencial de un sistema operativo que provee los servicios más básicos del sistema. Se encarga de gestionar los recursos como el acceso seguro al hardware de la computadora. Se encarga también del multiplexado, determinando qué programa accederá a un determinado hardware si dos o más quieren usarlo al mismo tiempo. El kernel también ofrece una serie de abstracciones del hardware para que los programadores no tengan que acceder directamente al hardware, proceso que puede ser complicado.

Linux.-Sistema operativo que posee un núcleo del mismo nombre. El código fuente es abierto, por lo tanto, está disponible para que cualquier persona pueda estudiarlo, usarlo, modificarlo y redistribuirlo. El término Linux se utiliza para describir al sistema operativo tipo Unix que utiliza filosofías y metodologías libres y que está constituido por la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU, además de otros

proyectos libres y no libres. El término Linux también hace referencia al kernel que utilizan múltiples sistemas operativos. Es ampliamente popular en el mercado de servidores.

Mysql.-MySQL es un sistema de gestión de bases de datos (SGBD) multiusuario, multiplataforma y de código abierto. MySQL pertenece a la compañía sueca MySQL AB, a la que le pertenece casi todos los derechos del código fuente. La compañía desarrolla y mantiene el sistema, vendiendo soporte y servicios, como también las licencias para usar MySQL.

ODBC.-En informática, el ODBC (Open Database Connectivity) es un estándar de acceso a bases de datos, que permite mantener independencia entre los lenguajes de programación, los sistemas de bases de datos (las bases de datos y su software gestor), y los sistemas operativos. ODBC inserta una "capa" entre la aplicación y el SGBD (sistema gestor de base de datos). Esta capa es llamada "manejador de base de datos". El objetivo de la capa es traducir las consultas a la base de datos (u otras acciones) por parte de la aplicación a una consulta que el SGBD comprenda.

Para que esto sea posible, tanto la aplicación como el SGBD deben ser compatibles con ODBC. ODBC es desarrollado por Microsoft Corporation.

Open source.-Denominación para aquellas aplicaciones que tienen su código fuente liberado. En general, los programas de código abierto suele ser libres. Aunque existen aplicaciones de código abierto que no son libres. Open Source es utilizado también para hacer referencia a un nuevo movimiento de software, la Open Source Initiative.

PBX.-(Private Branch Exchange) es una central telefónica privada. Permite interconectar los teléfonos internos de una empresa. Selecciona la línea saliente de forma (más o menos) Automática. Algunos permiten transferir llamadas, realizar conferencias, llevar un control de los números marcados. Es usado por empresas grandes y pequeñas para reducir costos. Hoy en día muchos PBX soportan el protocolo IP.

PHP.-(PHP Hypertext Pre-processor). Lenguaje de programación usado generalmente en la creación de contenidos para sitios web. Es un lenguaje interpretado especialmente usado para crear contenido dinámico web y aplicaciones para servidores, aunque también es posible crear aplicaciones gráficas utilizando la biblioteca GTK+. Generalmente los scripts en PHP se embeben en otros códigos como HTML, ampliando las posibilidades del diseñador de páginas web enormemente. La interpretación y ejecución de los scripts PHP se hacen en el servidor, el cliente (un navegador que pide una página web) sólo recibe el resultado de la ejecución y jamás ve el código

PHP. Permite la conexión a todo tipo de servidores de base de datos como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. PHP es una alternativa a otros sistemas como el ASP.NET/C#/VB.NET de Microsoft o a ColdFusion de Macromedia, a JSP/Java de Sun Microsystems, y a CGI/Perl. La ventaja con los de Microsoft o Macromedia es que es totalmente gratuito, no hay que pagar licencias.

PHP-AGI.-Librería que encapsula las funciones de Asterisk para que sea más fácil desarrollar programas en Asterisk para PHP.

Script.-Grupo de lenguajes de programación que son típicamente interpretados y pueden ser tipados directamente desde el teclado. Los scripts son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución, se distinguen de los programas, pues deben ser convertidos a un archivo binario ejecutable para correrlos. Los scripts pueden estar embebidos en otro lenguaje para aumentar las funcionalidades de este, como es el caso los scripts PHP o Javascript en código HTML.

Servidor.-En redes, computadora central en un sistema de red que provee servicios a otras computadoras. En internet, los servidores son los proveedores de todos sus servicios, incluyendo la WWW (las páginas web),

el FTP, el correo electrónico, los grupos de noticias, etc. Básicamente, una computadora conectada a internet emplea una dirección (dirección web, dirección IP, dirección FTP, etc.) para poder comunicarse con el servidor al que le corresponde. La computadora envía (utilizando el protocolo adecuado) las distintas solicitudes al servidor, y el servidor responde (empleando el protocolo adecuado) las solicitudes. El servidor también puede solicitar datos de la computadora, y la computadora le responde.

Softphone.-Software que realiza una simulación de teléfono en una computadora, permitiendo así la comunicación con otras computadoras que posean este mismo software, usando un VSP (VoIP Service Provider Proveedor de Servicios de VoIP, carrier voip, etc). Lo más comunes son: Skype, WengoPhone, Gizmo, XLite, el Aim, SjPhone, y últimamente Zoiper.

Telefonía IP.-Hace referencia a comunicaciones telefónicas realizadas a través de redes TCP/IP. A diferencia de PSTN, que se compone de señales analógicas y digitales a través de una red con conmutación de circuitos, la telefonía IP utiliza conmutación de paquetes. Toda la información que se va a transmitir a través de la red se divide en paquetes de datos.

Voz sobre IP.-Dice que es la tecnología que posibilita el uso de redes IPs como medio de transmisión de voz. El concepto es simple y consiste en

convertir los paquetes de voz, analógicos, en paquetes digitales y hacerlos transitar por internet. Con un relativo ancho de banda (128Kbps ya se garantiza una buena calidad de voz, siendo de 256Kbps en adelante mejor) es posible usar la capacidad de transporte del protocolo IP para transmitir voz.

X86.-Nombre dado al grupo de microprocesadores de la familia de Intel y a la arquitectura que comparten estos procesadores. Existen X86 tanto de 16 bits como de 32 bits, estos últimos llamados IA-32 (x86-32). Los microprocesadores x86 fueron 8086, 80286, 80386, 80486. Luego continuó con el Pentium. Su sucesor inmediato es el x86-64, creado por AMD (con el nombre de AMD64), y luego Intel creó su propia versión llamada Intel 64. Ambas implementan los 64 bits y son muy similares entre sí.

Zoiper.- Véase softphone.

ANEXOS

ANEXO A

Para el sistema de administración de citas medicas se realizó una página web en código PHP, la carpeta fue nombrada Clinica1 y ubicada en la siguiente ruta /var/www/Clinica1.

Dentro de la carpeta Clinica1 existen archivos con el código PHP, el cual nos permite interactuar con la base de datos que posee la clínica.

En el siguiente código se muestra las acciones que se deberá realizar al momento de seleccionar algún enlace u opciones.

```
<?php
    session_start();
if(!isset($_SESSION['usuario'])){
echo "<meta http-equiv='refresh' content='0; url=../index.php?error=2'>";
die; }
if(isset($_GET['opcion']))
    {
switch($_GET['opcion'])
    {
case 'Salir':
unset($_SESSION['usuario']);
        session_destroy();
echo "<meta http-equiv='refresh' content='0; url=../index.php'>";
break; }
die; }
```

```
include_once '../sql/conexion.php';

$con = new conexion();

$action = $_POST['accion'];
if($accion == "Borrar")
{
    $id = $_POST['id'];
$query = "UPDATE paciente set estado ='E' Where idpaciente = $id";
    $con->modificarSQL($query);
echo "<meta http-equiv='refresh' content='0; url=../PHP/menuAgente.php'>";
}elseif($accion == "Editar") {
    $id = $_POST['id'];
echo "<form name='f' method='post' action='../PHP/aEdicion.php'>
<input name='id' value='$id' />
</form>
<script type='text/javascript'>document.f.submit();</script>";
}elseif($accion == "Actualizar")
{
    $id = $_POST['id'];
    $cedula = $_POST['cedula'];
$nombre = $_POST['nombres'];
$apellido = $_POST['apellido'];
$apellidoM = $_POST['apellido1'];
$edad = $_POST['edad'];
    $sexo = $_POST['sexo'];
$telefono = $_POST['telefono'];
$carpeta = $_POST['carpeta'];
```

```

$query = "UPDATE paciente set cedula='$cedula',
nombres='$nombre',
apellidop='$apellido',
apellidom='$apellidoM',
telefono='$telefono',
edad=$edad,
sexo='$sexo',
numcarp=$carpeta
where idpaciente=$id";

$con->modificarSQL($query);
echo "<meta http-equiv='refresh' content='0; url=../PHP/menuAgente.php'>";
}elseif($accion=="Resetear")
{
    $op = $_POST['Opass'];
    $np = $_POST['Npass'];
    $np2 = $_POST['N2pass'];
    $us = $_POST['usuario'];
    $tipo = $_POST['tipo'];
    if($np != $np2)
    {
        echo "<meta http-equiv='refresh' content='0;
url=../PHP/areseteoPassword.php?error='>";
    }else{
        $query = "Select password from usuarios where usuario = '$us'";
        $a = $con->informacionSQL($query);
        if($a[0]->password!=$op)

```

```

echo          "<meta          http-equiv='refresh'          content='0;
url=../PHP/$tipo"."reseteoPassword.php?error='>";

else    {

        $query = "UPDATE usuarios set password ='$np' where usuario =
'$us'";

        $con->modificarSQL($query);

if($_POST['tipo']=='a')
echo "<meta http-equiv='refresh' content='0; url=../PHP/menuAgente.php'>";
elseif($_POST['tipo']=='d')
echo "<meta http-equiv='refresh' content='0; url=../PHP/menuDoctor.php'>";
        }
    }
}
?>

```

El siguiente Código es para poder realizar la conexión con MySQL y así facilitar las respectivas consultas de las tablas realizadas.

```

<?php
class conexion
{
    function informacionSQL($query)
    {
        $conexion = mysql_connect('localhost', 'root', 'labtelecom10');
        mysql_select_db('db_cita',$conexion);
        $result = mysql_query($query,$conexion);
    }
}

```

```

        $arr=array();
        while ($row = mysql_fetch_object($result)) {
            $arr[]=$row;
        }
        mysql_free_result($result);
        mysql_close($conexion);
        return $arr;
    }
    function modificarSQL($query)
    {
        $conexion = mysql_connect('localhost', 'root', 'labtelecom10');
        mysql_select_db('db_cita',$conexion);
        $result = mysql_query($query,$conexion);
        mysql_close($conexion);
    }
}??>

```

Continuando con la explicación acerca de la función de cada código en el siguiente permitirá realizar ingreso de pacientes, pero cabe recordar que esta opción sólo está disponible para los usuarios de tipo Agente.

```

<?php
    session_start();
    if(!isset($_SESSION['usuario'])){

```



```

echo "<meta http-equiv='refresh' content='0; url=../index.php?error=2'>";
die;
    }
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title></title>
</head>
<body>
<?php
    include_once 'conexion.php';
    $cedula = $_POST['cedula'];
    $nombre = $_POST['nombres'];
    $apellido = $_POST['apellido'];
    $apellidoM = $_POST['apellido1'];
    $edad = $_POST['edad'];
    $sexo = $_POST['sexo'];
    $telefono = $_POST['telefono'];
    $carpeta = $_POST['carpeta'];
    $query = "INSERT INTO paciente
(cedula,nombres,apellidop,apellidom,telefono,edad,sexo,numcarp,estado)
VALUES('$cedula','$nombre','$apellido','$apellidoM','$telefono','$edad','$sexo',
$carpeta,'A')";
    $conexion = new conexion();
    $conexion->modificarSQL($query);

```

```
echo "<meta http-equiv='refresh' content='0; url=../PHP/menuAgente.php'>";
    ?>
</body>
</html>
```

Este código que se presenta a continuación es de mucha ayuda para el sistema, porque con este código se podrá acceder al sistema e identificar si es un usuario de tipo Doctor o tipo Agente.

```
<html>
<body>
<?php
    include_once "conexion.php";
    $user = $_POST["user"];
    $pass = $_POST["pass"];
    $c = new conexion();
    $query = "select * from usuarios where usuario ='$user' and password
= '$pass'";
    $a = $c->informacionSQL($query);
if(count($a)>0)
    {
    session_start();
    $_SESSION['usuario']=$user;
    if($a[0]->tipo == "doctor")
echo "<meta http-equiv='refresh' content='0; url=../PHP/menuDoctor.php'>";
        elseif($a[0]->tipo == "agente")
```

```
echo "<meta http-equiv='refresh' content='0; url=../PHP/menuAgente.php'>";
    }else
    {
echo "<meta http-equiv='refresh' content='0; url=../index.php?error=1'>";
    }
?>
</body>
</html>
```

Ya una vez ingresando los datos de usuario y contraseña, este identifica que es un usuario de tipo Agente se procederá a ejecutar el siguiente comando el cual muestra las opciones que hace diferencia con el del Doctor.

```
<?php
    session_start();
if(!isset($_SESSION['usuario'])){
echo "<meta http-equiv='refresh' content='0; url=../index.php?error=2'>";
die;
    }
?>
<html>
<head>
<title>Clinica Esperanza</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link href="../estilo.css" rel="stylesheet" type="text/css"/>
</head>
```

```

        <body>
<div id="cabecera">
<h1>Clinica Esperanza</h1>
</div>
<div id="menuTop">
<a href="aDirectorio.php">Directorio</a>
<a href="areseteoPassword.php">Cambiar Password</a>
<a href=" ../sql/AccionPaciente.php?opcion=Salir">Salir</a>
</div>
        <div id="menu">
                <p><a href="aPaciente.php">Consulta de
Paciente</a></p>
                <p><a href="aIngreso.php">Ingreso de Paciente</a></p>
                <p><a href="aCita.php">Consulta de Cita</a></p>
        </div>
</body>
</html>

```

El mismo caso que el anterior código aquí se ejecutará el del usuario tipo Doctor quien tiene opciones muy específicas para su labor.

```

<?php
    session_start();
    if(!isset($_SESSION['usuario'])){
    echo "<meta http-equiv='refresh' content='0'; url=../index.php?error=2'>";
    die; }

```

```
?>
<html>
<body>
<head>
<title>Clinica Esperanza</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link href="../estilo.css" rel="stylesheet" type="text/css"/>
</head>
<div id="cabecera">
<h1>Clinica Esperanza</h1>
</div>
<div id="menuTop">
<a href="cDirectorio.php">Directorio</a>
<a href="dreseteoPassword.php">Cambiar Password</a>
<a href="../sql/AccionPaciente.php?opcion=Salir">Salir</a>
</div>
        <div id="menu">
        <p><a href="cPaciente.php">Consulta de Paciente</a></p>
        <p><a href="cRegistro.php">Consulta de Registro</a></p>
        </div>
    </body>
</html>
```

BIBLIOGRAFÍA

[1]Phpmyadmin.com,http://www.phpmyadmin.net/home_page/index.php, Septiembre 2010.

[2] asteriskguide.com, Integración de Asterisk usando AGI y AMI, http://www.asteriskguide.com/mediawiki/index.php/Integraci%C3%B3n_de_Asterisk_usando_AGI_y_AMI, Septiembre 2010.

[3] asteriskguide.com, Introducción al Asterisk, http://www.asteriskguide.com/mediawiki/index.php/Introducc%C3%ADn_al_Asterisk, octubre 2010.

[4] VOIP-Info.org, Asterisk AGI, <http://www.voip-info.org/wiki/view/Asterisk+AGI>, Noviembre 2010

[5] Packtpub.com, Asterisk Gateway Interface Scripting with PHP <http://www.packtpub.com/article/asterisk-gateway-interface-scripting-with-php>, Noviembre 2010

[6]Van Meggelen J., Smith J., Madsen L., Asterisk The Future of Telephony, Editorial O' Relly Media, 2005, publicado en <http://www.asterix.org>, Julio 2010

[7]VOIP-Info.org, Asterisk simple php lookup MySQL database,
[http://www.voip-
info.org/wiki/view/Asterisk+simple+php+lookup+mysql+database+to+s
et+callerid+name](http://www.voip-info.org/wiki/view/Asterisk+simple+php+lookup+mysql+database+to+set+callerid+name), Enero 2011

[8]sourceforge.net,PHPAGI the connection de php and asterisk,
<http://phpagi.sourceforge.net/>, Enero 2011

[9] blog.oaxrom.com, Como implementar un agi con php,
[http://blog.oaxrom.com/index.php/blog/show/Como_implementar_un_A
GI_con_PHP_en_un_conmutador_de_VoIP_sobre_Asterisk.html](http://blog.oaxrom.com/index.php/blog/show/Como_implementar_un_AGI_con_PHP_en_un_conmutador_de_VoIP_sobre_Asterisk.html),
Enero 2011

[10] Wikipedia, Asterisk, <http://es.wikipedia.org/wiki/Asterisk>,
Noviembre 2010.