

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA DE INGENIERÍA ELÉCTRICA



## **Servicio de respuesta de voz iterativa para el Hospital Nacional Rosales**

PRESENTADO POR:  
**RONY STALYN SÁNCHEZ MORALES**

PARA OPTAR AL TITULO DE:  
**INGENIERO ELECTRICISTA**

CIUDAD UNIVERSITARIA, AGOSTO DEL 2012

# **UNIVERSIDAD DE EL SALVADOR**

RECTOR :

**ING. MARIO ROBERTO NIETO LOVO**

SECRETARIA GENERAL :

**DRA. ANA LETICIA ZAVALA DE AMAYA**

## **FACULTAD DE INGENIERÍA Y ARQUITECTURA**

DECANO :

**ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL**

SECRETARIO :

**ING. JULIO ALBERTO PORTILLO**

## **ESCUELA DE INGENIERÍA ELÉCTRICA**

DIRECTOR :

**ING. JOSÉ WILBER CALDERÓN URRUTIA**

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

**INGENIERO ELECTRICISTA**

Título :

**Servicio de respuesta de voz iterativa  
para el Hospital Nacional Rosales**

Presentado por :

**RONY STALYN SÁNCHEZ MORALES**

Trabajo de Graduación Aprobado por :

Docente Director :

**Dr. CARLOS EUGENIO MARTÍNEZ CRUZ**

San Salvador, Agosto de 2012

Trabajo de Graduación Aprobado por:

Docente Director :

**Dr. CARLOS EUGENIO MARTÍNEZ CRUZ**

## ACTA DE CONSTANCIA DE NOTA Y DEFENSA FINAL

En esta fecha, 11 de julio de 2012, en la Planta Baja de la Escuela de Ingeniería Eléctrica, a las 4:00 horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. Ing. José Wilber Calderón Urrutia  
Director

Firma:  


2. Ing. Salvador de Jesús Germán  
Secretario

Firma:   


Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

1- Ing. Román Abad Tobías Vides

Firma:  


2- Ing. Werner David Meléndez Valle



3- Ing. Carlos Osmín Pocasangre Jiménez



Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

Servicio de respuesta de voz iterativa para el Hospital Nacional Rosales.

A cargo del Bachiller:

Rony Stanlyn Sánchez Morales

Habiendo obtenido el presente Trabajo una nota final, global de: 8.7

( OCHO PUNTO SIETE )

## **Dedicatoria**

A mi madre y hermanos, por creer en mi y apoyarme siempre hasta la finalización de esta carrera, han sido ellos realmente los únicos que han hecho de mi la persona que soy.

Especialmente dedicado a la memoria de nuestro amigo Christian Ramos, que dios te tenga en su gloria y que este trabajo de graduación sea un instrumento para evitar que mas seres queridos sigan abandonado sus sueños.

## **Agradecimientos**

En primer lugar agradecerle a Dios por darme fuerzas y sabiduría, fundamental para finalizar este objetivo.

Infinitamente agradecido a mi amada madre Gloria Elizabeth Morales Valdizón que junto a mis hermanos Pamela y Jeancarlos , hacemos la mejor familia del mundo.

A mis tíos: Israel, Ernesto y Mario que me demuestran su cariño y apoyo muy a su manera, me siento muy afortunado de tener una figura paterna de sus personas.

Gracias a mi Abuela Marina y a mi tía Elba por ser como mis madres y apoyarme siempre en este camino.

A todos los chicos de la Familia: Mónica, Mario jr, Luis y Roy que me alegran la vida con sus ocurrencias .

Todos mis amigos y compañeros con quienes compartí momentos inolvidables en la Universidad de El Salvador: Pablo Cartagena, Luis Colocho, Luis Valdiviezo, Antonio Lemus, Oscar Morales, Roman Tobias, Jose Galvez y Felix Campos.

Amistades sinceras y queridas de las diferentes carreras en la facultad de Ingeniería y Arquitectura de la UES.

A mi asesor de Tesis el Dr. Carlos Eugenio Martínez, quien ha sido como un mentor en mi carrera.

Y finalmente pero no menos importante a toda la comunidad desarrolladora de software y hardware libre, ya que sin sus aportes, este trabajo de graduación no se hubiera podido llevar acabo.

# Índice de contenido

CAPITULO I. INTRODUCCIÓN.....	1
1.1 Necesidad de la investigación.....	1
1.2 Motivación para realizar el proyecto.....	2
1.3 Objetivos y organización.....	2
1.3.1 Objetivo General.....	2
1.3.2 Objetivos Específicos. ....	2
1.3.3 Organización.....	2
CAPITULO II. CONCEPTOS INVOLUCRADOS EN EL SERVICIO ITERATIVO DE VOZ.....	4
2.1 Protocolos de señalización.....	4
2.1.1 SIP.....	4
2.1.2 IAX.....	7
2.2 Codec de voz utilizados por Asterisk.....	10
2.2.1 G711u y G729a.....	10
2.2.2 GSM.....	10
2.2.3 Método simplificado para el Cálculo de ancho de banda utilizado por Codecs.....	10
2.3 Troncales SIP.....	11
2.4 Tipos de tecnologías en interfaces de voz.....	12
2.4.1 Detección de tonos DTMF.....	12
2.4.2 Reconocimiento de voz ASR.....	12
2.4.3 Síntesis de voz TTS.....	12
2.5 Definición de una IVR.....	13
2.6 Principios generales en el diseño.....	13
2.7 Principales errores de ergonomía.....	14
2.8 Cantidad óptima de menús a presentar al usuario.....	14
2.9 Opciones de navegación recomendables.....	15
CAPITULO III. INSTALACIÓN DEL SOFTWARE.....	16
3.1 Instalación de Asterisk 1.8 en Ubuntu 11.04 LTS.....	16
3.1.1 LIBPRI.....	17
3.1.2 DAHDI.....	17
3.1.3 Asterisk.....	17
3.2 Instalación de MySQL.....	18
3.3 Instalación de unixODBC.....	19
3.4 Instalación de Festival.....	19
CAPITULO IV. IMPLEMENTACIÓN DEL SERVICIO IVR PARA EL HNR.....	20
4.1 Servicio IVR para HNR.....	20
Menú principal.....	20
Menú 1. Cirugía ambulatoria.....	21
Menú 2. Laboratorio clínico.....	21
Menú 3. Registro clínico.....	22
Menú 4. Atención personalizada por operadores.....	23
4.2 Diseño de la Red Telefónica.....	23
4.3 Configuración de la troncal SIP en servidor Asterisk UES.....	25
4.4 Configuración de extensiones SIP internas.....	26
4.5 Configuración de MySQL, unixODBC y Festival para acceder a bases de datos.....	27
4.5.1 Base de datos para los registros de pacientes.....	28



4.5.2 Configuración de unixODBC.....	29
4.5.3 Configuración de los módulos res_odbc y func_odbc.....	30
4.5.4 Configuración de Festival en Asterisk.....	32
4.6 Configuración del plan de marcado de la red interna.....	32
4.7 Configuración del plan de marcado para llamadas entrantes.....	33
4.7.1 Contexto IVR.....	33
4.7.2 Contexto IVR1.....	34
4.7.3 Contexto IVR2.....	35
4.7.4 Contexto IVR3.....	36
4.7.5 Contexto IVR4.....	37
CAPITULO V. IMPLEMENTACIÓN DEL SERVICIO IVR PARA UN ENTORNO RURAL.....	38
5.1 Configuración de Red inalámbrica para integrar los ECOS.....	39
5.2 Configuración de las Troncales SIP en routers MP.....	40
5.2.1 troncal SIP MP 01.....	40
5.2.2 troncal SIP MP 02.....	41
5.2.3 troncal SIP MP 03.....	41
5.3 Configuración de las extensiones SIP en routers MP.....	41
5.3.2 Extensiones en MP 02.....	42
5.3.3 Extensiones en MP 03.....	42
5.3 Configuración del plan de marcado en routers MP.....	42
CAPITULO VI. CONCLUSIONES Y LINEAS FUTURAS.....	44
6.1 Conclusiones.....	44
6.2 Lineas futuras.....	45
APÉNDICES.....	46
APÉNDICE A. Software complementario utilizado en el servicio.....	46
Wireshark.....	46
Configuración de un cliente IAX con ZOIPER.....	46
APÉNDICE B. Tipos de Infraestructura en redes inalámbricas.....	48
Redes con infraestructura.....	48
Redes sin infraestructura (Ad-Hoc).....	48
APÉNDICE C. Redes WIFI MESH.....	49
Funcionamiento de la Red.....	49
Protocolo de Enrutamiento B.A.T.M.A.N.....	50
APÉNDICE D. Configuración del Adaptador PAP2T.....	50
BIBLIOGRAFÍA.....	52

## Índice de ilustraciones

Ilustración 1: Proxy SIP.....	5
Ilustración 2: Protocolo SIP.....	6
Ilustración 3: Protocolo IAX.....	8
Ilustración 4: Ejemplo de calculo del ancho de banda para codec.....	10
Ilustración 5: Consideraciones para simplificar la cantidad de lineas telefónicas utilizadas.....	11
Ilustración 6: Ejemplo para calcular el ancho de banda necesario.....	11
Ilustración 7: funcionamiento de las troncales SIP .....	11
Ilustración 8: Menú principal del servicio IVR.....	20
Ilustración 9: Menú 1 para proporcionar información de cirugía ambulatoria.....	21
Ilustración 10: Menú 2 para proporcionar información de Laboratorio Clínico.....	22
Ilustración 11: Menú 3 para proporcionar información de registro clínico.....	22
Ilustración 12: menú 4 para desvió de llamada a operadores internos.....	23
Ilustración 13: Diseño de la red telefónica.....	24
Ilustración 14: Conexión entre Asterisk y MySQL.....	27
Ilustración 15: Redes de Bajo costo basadas en estándar WIFI.....	38
Ilustración 16: Diseño de la Red para dar servicio IVR en los ECOS.....	39
Ilustración 17: Puertos del Router MP.....	40
Ilustración 18: Plan de marcado para llamadas salientes.....	43
Ilustración 19: Zoiper 1.....	47
Ilustración 20: Zoiper 2.....	47
Ilustración 21: Infraestructura WIFI.....	48
Ilustración 22: Redes Ad-Hoc.....	48
Ilustración 23: Topología de Red Mesh.....	49
Ilustración 24: ATA Linksys 1.....	51
Ilustración 25: ATA Linksys 2.....	51

# Índice de Instalación

Instalación 1: Actualización del sistema operativo.....	16
Instalación 2: Dependencias de Asterisk.....	16
Instalación 3: Descarga de paquetes Asterisk.....	16
Instalación 4: Dependencias y módulos Asterisk.....	16
Instalación 5: Instalación de LIBPRI.....	17
Instalación 6: Instalación de DAHDI.....	17
Instalación 7: Instalación de Asterisk.....	17
Instalación 8: Instalación de ejemplos para la configuración en Asterisk.....	17
Instalación 9: Validación y Permisos para ejecutar Asterisk.....	18
Instalación 10: Instalación de MySQL.....	18
Instalación 11: Contraseña root para MySQL.....	18
Instalación 12: Accediendo a consola de MySQL.....	18
Instalación 13: Instalación de unixODBC.....	19
Instalación 14: Festival y dependencias.....	19
Instalación 15: Voces en español para Festival.....	19
Instalación 16: Archivo de configuración para voces en español Festival.....	19
Instalación 17: Wireshark en Ubuntu.....	46
Instalación 18: Instalación de Zoiper.....	47
Instalación 19: Instalación de Zoiper para ejecutar desde consola.....	47

## Índice de configuración

Configuración 1: Tarjeta de red.....	24
Configuración 2: Interfaces eth0 en servidor Asterisk.....	24
Configuración 3: Prueba de conectividad con proveedor VoIP.....	24
Configuración 4: Troncal SIP 1.....	25
Configuración 5: Troncal SIP 2.....	25
Configuración 6: Cargando la configuración de troncal SIP.....	26
Configuración 7: Prueba de estado para la troncal SIP.....	26
Configuración 8: Extensiones internas, archivo sip.conf.....	27
Configuración 9: Tablas en base de datos MySQL.....	28
Configuración 10: Tabla citas.....	28
Configuración 11: Tabla referencias.....	29
Configuración 12: Tabla farmacia.....	29
Configuración 13: Archivo odbcinst.ini para registrar DSN.....	29
Configuración 14: Archivo odbc.ini para conexión con MySQL.....	30
Configuración 15: Prueba de conexión para bases de datos.....	30
Configuración 16: Verificación de módulos odbc.....	30
Configuración 17: Registrando el DSN para crear las funciones de consultas hacia BD.....	31
Configuración 18: Ejemplo de función en módulo de Asterisk para hacer consultas a BD.....	31
Configuración 19: Configuración de Festival.....	32
Configuración 20: Recargando las configuraciones de Asterisk.....	32
Configuración 21: Prueba de conexión Asterisk y MySQL.....	32
Configuración 22: Contexto extensiones.....	33
Configuración 23: Menú principal del servicio IVR.....	34
Configuración 24: Contexto IVR1.....	35
Configuración 25: Contextos IVR[a,b,c,d] para información de cirugía ambulatoria.....	35
Configuración 26: Contexto IVR2.....	36
Configuración 27: Contexto IVR3.....	36
Configuración 28: Contexto IVR4.....	37
Configuración 29: Registro de la troncal MP 01.....	40
Configuración 30: Registro de la troncal MP 02.....	41
Configuración 31: Registro de la troncal MP 03.....	41
Configuración 32: Extensiones SIP MP 01.....	41
Configuración 33: Extensiones SIP MP 02.....	42
Configuración 34: Extensiones SIP MP 03.....	42
Configuración 35: Plan de marcado MP 01.....	42
Configuración 36: Plan de marcado MP 02.....	42
Configuración 37: Plan de marcado MP 03.....	43
Configuración 38: Configuración del Servidor para utilizar IAX.....	46
Configuración 39: Extensión con protocolo IAX.....	47

## **CAPITULO I. INTRODUCCIÓN.**

El hospital público más grande de El Salvador es el Hospital Nacional Rosales el cual atiende anualmente miles de consultas. Diariamente muchas personas acuden a este hospital para recibir diferentes tipos de servicio. Estos servicios muchas veces se ven frustrados por la falta de información, también porque los pacientes no atienden las indicaciones y las recomendaciones que los profesionales de la salud les proporcionan.

En el presente trabajo de graduación se implementa una herramienta de comunicación basada en los más recientes avances tecnológicos en materia de voz sobre protocolo de Internet (VoIP). La herramienta a desarrollar es un sistema de respuesta de voz iterativo (IVR), basado en software libre. Éste será usado por el Hospital Nacional Rosales (HNR). De esta manera el servicio de atención a los pacientes podrá ofrecerse las 24 horas del día, los 7 días de la semana, en todo el año. La idea encaja muy bien con los planes del ministerio de salud (MINSAL) de implementar centrales telefónicas basadas en software libre.

El servicio IVR será capaz de atender a los pacientes del Hospital Nacional Rosales, proporcionando información e indicaciones generales. Desde un principio este trabajo de graduación definió trabajar con aquellos departamentos del HNR que tuvieran interés en mejorar el servicio de atención al paciente. A partir de reuniones mantenidas con personal del MINSAL y con funcionarios del HNR se seleccionaron dos departamentos: Cirugía ambulatoria (comité gestor de camas) y departamento de laboratorio clínico.

También, a partir de reuniones sostenidas con el director de la dirección de tecnologías de la información y la comunicación del MINSAL se propuso incluir la posibilidad de interconectar el servicio IVR con los registros de los pacientes del HNR. Sobre esta propuesta se realizó una extensa investigación y se realizarón muchas pruebas de laboratorio que produjeron resultados satisfactorios.

### ***1.1 Necesidad de la investigación.***

Esta surge del problema que el HNR tiene en cuanto al servicio de información al paciente. Anualmente el HNR atiende cerca de 250,000 consultas y cada día muchos pacientes acuden a sus citas, algunas de ellas se ven frustradas, las pruebas de laboratorio son ejemplos muy claros de la perdida de servicios por parte de los pacientes. Otro ejemplo es la falta de preparación del paciente previo a la ejecución de una cirugía.

Es posible mejorar los flujos de información hacia el paciente mediante la introducción de tecnologías de la información y las comunicaciones. El costo de la implantación de estas tecnologías pueden reducirse si se utilizan herramientas basadas en software libre.

## **1.2 Motivación para realizar el proyecto.**

El HNR no cuenta con herramientas tecnológicas para facilitar la información a sus pacientes. Implementar un servicio IVR podría ayudar a mejorar la atención al público. Por ejemplo, se podría trasladar indicaciones acerca de cuidados antes y después de cirugías. También, se podría mejorar la realización efectiva de pruebas de laboratorio clínico.

Además, la interconexión del servicio IVR con las bases de datos de los pacientes permitiría incrementar el flujo de información desde el HNR hacia los pacientes. Por ejemplo: si una persona necesita saber el día y la hora de su cita programada en el HNR, entonces bastaría con consultar la IVR. Mediante el número de registro, proporcionado por el HNR, se llamaría a un número telefónico y el sistema le proporcionaría de manera automática la información necesitada.

Otra motivación importante es la económica. La implementación de este servicio sobre plataformas de software libre contribuye al ahorro. Además, se cuenta con mucho soporte técnico gratuito en la Internet.

## **1.3 Objetivos y organización.**

Los objetivos planteados en el presente proyecto de graduación son:

### **1.3.1 Objetivo General.**

Contribuir a mejorar la atención de los pacientes del Hospital Nacional Rosales.

### **1.3.2 Objetivos Específicos.**

Implementar un servicio de respuesta de voz iterativo que ayude a canalizar la información de forma más efectiva.

Utilizar herramientas de software libre para disminuir costos.

Contribuir al mejoramiento de la atención al paciente a través de la introducción de servicios telefónicos.

### **1.3.3 Organización.**

En el capítulo I se presenta los intereses del proyecto organización del mismo, además de la motivación para llevar a cabo este servicio y sus objetivos de trabajo.

En el capítulo II se trata la temática de todos los conceptos involucrados para

poder entender el desarrollo del proyecto. Así como también se explica el funcionamiento de protocolos y los codificadores-decodificadores (codecs) más utilizados. Además, se explican las consideraciones de diseño para la implementación del servicio IVR.

En el capítulo III se presenta la instalación del software necesario para implementar el servicio IVR. Se explica la instalación de Asterisk en Ubuntu-linux, el gestor de bases de datos MySQL, el sintetizador de voz Festival.

En el capítulo IV se desarrollan los pasos de configuración del servicio IVR. Este servicio no quedará implantado en el HNR, debido a que aún no se ha realizado la introducción de centrales basadas en software libre en las instalaciones del HNR. Sin embargo, los pasos de configuración y el diseño de red telefónica propuesto en este capítulo serán de fácil implementación una vez el HNR haya cambiado su central telefónica. Dentro de la configuración y el diseño se hace énfasis en la configuración de la troncal SIP, debido a que ésta permite enrutar las llamadas salientes y entrantes. También se realiza la configuración de la conexión entre Asterisk y MySQL para acceder a bases de datos. Finalmente se desarrollan los planes de marcado para las llamadas internas y externas.

En el capítulo V se presenta una propuesta de red inalámbrica que puede servir en un entorno rural. Esta idea pretende ofrecer servicios de telecomunicación a los equipos comunitarios de salud (ECOS) creados por el MINSAL. El concepto de red inalámbrica y de sistema de telefonía propuesto se basan en los desarrollos realizados en la escuela de ingeniería eléctrica en últimos años.

Finalmente, en el capítulo VI se presentan las conclusiones y líneas futuras.

## **CAPITULO II. CONCEPTOS INVOLUCRADOS EN EL SERVICIO ITERATIVO DE VOZ.**

En este capítulo se abordaran todos los conceptos necesarios para diseñar e implementar el servicio IVR. Iniciamos con los conceptos relacionados al software Asterisk. Se hace énfasis en los protocolos de señalización SIP e IAX. También, se explican el concepto de troncal. Su importancia radica en que a través de este se establecen las conexiones entre centrales. Además, se aborda el concepto de codec, involucrado en la digitalización de voz. También, se explica como funcionan los tipos de tecnologías que establecen la conexión entre la IVR y la base de datos.

### **2.1 Protocolos de señalización.**

La señalización en telefonía es muy importante debido a que es la encargada de establecer, mantener y finalizar una conversación entre dos terminales. Además, ofrece funciones de supervisión, marcado de llamada y retorno de tonos.

En la red de telefonía pública (PSTN, del ingles Public Switched Telephone Network) la voz y la señalización están separadas. Esto quiere decir que se dedica un circuito a voz y otro circuito a señalización. El que la voz y la señalización estén separadas se traduce en que éstas toman caminos físicos diferentes. Es decir, la voz podría viajar por un cable y la señalización por otro[1].

De manera similar a la PSTN, la señalización VoIP sigue los mismos principios, la voz y la señalización viajan por caminos diferentes como se muestra en la Ilustración 1. A continuación se describe dos de los protocolos más utilizados en VoIP.

#### **2.1.1 SIP.**

El protocolo de inicio de sesión SIP (del inglés *Session Initiation Protocol*) es utilizado para iniciar y terminar llamadas de los servidores VoIP[2]. SIP es muy parecido al protocolo HTTP porque sigue una estructura de mensajerías, petición y respuesta, con códigos similares al HTTP. Por ejemplo, un código de retorno 200ok significa éxito en la petición. Un código 404 significa no encontrado, es decir que no se pudo establecer la comunicación. Entonces este protocolo se basa en el intercambio de peticiones y respuestas. Los mensajes de sincronización son:

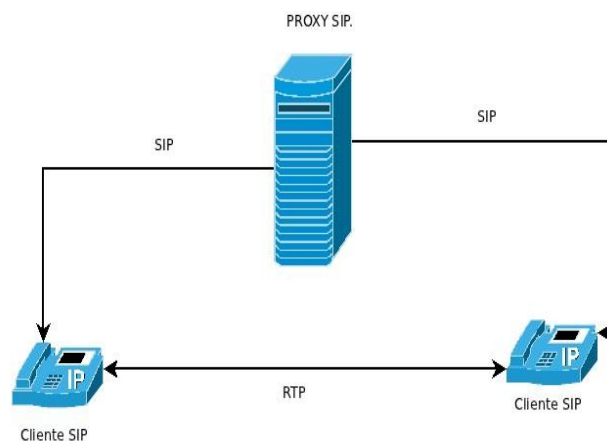
- Invite: utilizados para invitar un usuario a participar en una sesión o simplemente para modificar parámetros.
- Ack: este mensaje confirma el establecimiento de una sesión.
- Option: solicita información de las capacidades del servidor VoIP.
- Bye: indica que la sesión ha sido terminada.
- Cancel: cancela una petición pendiente.
- Register: registra a un usuario o agente.



Las respuestas generadas a partir de una petición devuelven un código en este caso la línea inicial recibe el nombre de "status line". Se puede recibir las siguientes respuestas:

- 1XX: mensaje provisional.
- 2XX: Éxito.
- 3XX: redireccionamiento.
- 4XX: fallos del método.
- 5XX: fallos del servidor VoIP.
- 6XX: fallos globales.

SIP es el protocolo de señalización más utilizado. Las ventajas que se le atribuyen son la transmisión de audio y de vídeo en tiempo real, algo esencial en las telecomunicaciones. Como se puede ver en la Ilustración 1, los clientes SIP pueden comunicarse directamente por medio de los puertos RTP.[1] Normalmente SIP hace uso de elementos provisionales llamados proxies para facilitar el establecimiento de las llamadas. Un proxy, actúa como el apoderado para negociar entre dos partes, entonces queda claro que la señalización SIP y las conversaciones de voz viajan por caminos diferentes es decir se utilizan puertos diferentes.



*Ilustración 1: Proxy SIP.*

Como ejemplo se muestra el establecimiento de una llamada realizada desde la PSTN hacia una extensión de la red VoIP de la UES. La comunicación se estableció entre el número 22261683 (dado de alta en la PSTN) y el número 25112110 (dado de alta en red VoIP de la UES).

Es posible monitorizar llamadas VoIP utilizando software libre. En este trabajo de graduación se utilizó Wireshark<sup>1</sup>. Este programa es un analizador de protocolos telefónicos y de datos. Wireshark resultó de mucha utilidad para entender el proceso de establecimiento de una llamada. Los resultados de utilización de este software se muestra en la Ilustración 2. En ésta se observan tres columnas. La primera contiene las mediciones de tiempo. La segunda columna muestra el intercambio de información entre dos servidores SIP. La tercera columna contiene los comentarios respectivos. El análisis del protocolo SIP realizado por Wireshark se muestra en la segunda columna. En las líneas 1

1 <http://www.wireshark.org/>

y 2 se muestran las direcciones de los dos servidores VoIP involucrados en la comunicación telefónica. La dirección IP 168.243.7.190 corresponde al servidor VoIP de la UES y la dirección IP 10.10.89.99 corresponde al servidor VoIP instalado en el laboratorio de telecomunicaciones en la Escuela de Ingeniería Eléctrica (EIE).

1.	Time	168.243.7.190		
2.			10.10.89.99	
3.	26.095	INVITE SDP (g711A GSM g711U telephone-eventRTP...e-101)	SIP From:"22261683"	
		<:sip:22261683@168.243.7.190 To:<:sip:s@10.10.89.99:5060		
4.		(5060) ----->	(5060)	
5.	26.096	100 Trying		SIP Status
6.		(5060) <-----	(5060)	
7.	30.100	200 OK SDP (GSM g711U g711A telephone-eventRTP...e-101)		SIP Status
8.		(5060) <-----	(5060)	
9.	30.135	ACK		SIP Request
10.		(5060) ----->	(5060)	
11.	30.626	RTP (GSM)		RTP Num packets:970 Duration:21.457s SSRC:0x608DD7DD
12.		(11540) <-----	(19142)	
13.	30.644	RTP (GSM)		RTP Num packets:750 Duration:14.980s SSRC:0x5359ED95
14.		(11540) <-----	(19142)	
15.	45.644	RTP (telephone-event) DTMF Two 2		RTP Num packets:9 Duration:0.120s SSRC:0x5359ED95
16.		(11540) <-----	(19142)	
17.	45.784	RTP (GSM)		RTP Num packets:315 Duration:6.280s SSRC:0x5359ED95
18.		(11540) <-----	(19142)	
19.	52.086	BYE		SIP Request
20.		(5060) ----->	(5060)	
21.	52.086	200 OK		SIP Status
22.		(5060) <-----	(5060)	

*Ilustración 2: Protocolo SIP.*

La línea 3 muestra la petición **INVITE** que es una solicitud realizada por el servidor VoIP de la UES al Servidor VoIP de la EIE. La solicitud requiere trasladar la llamada entrante desde la PSTN dirigida al servidor VoIP de la EIE. Esto se observa cuando el abonado llamante 22261683@168.243.7.190 invita a establecer comunicación con el abonado destino sip:s@10.10.89.99:5060. La línea 4 muestra el puerto a través del cual se lleva a cabo el proceso de señalización SIP, el cual es el número 5060.

En las líneas 5 y 6 se observa como el servidor VoIP de la EIE da una respuesta **100 Trying** al servidor VoIP de la UES, a través del puerto 5060. Las líneas 7 y 8 muestran que el servidor VoIP EIE envía la respuesta **200 OK**, automáticamente después de haber enviado el mensaje **100 Trying**. También puede observarse como el servidor VoIP de la UES advierte que solo puede entender información codificada mediante el estándar GSM, g711u y g711. En las líneas 9 y 10 se observa como el servidor VoIP de la UES envía un mensaje de confirmación **ACK** al servidor VoIP de la EIE, a través del puerto 5060. En las líneas 11-18 se observa como la señal de voz se establece a través de protocolo RTP (puerto 11540 para el servidor UES y puerto 19142 para el servidor EIE) y la información es codificada a través del estándar GSM. Puede observarse que en la línea 15 el servidor VoIP de la EIE reconoce un tono de marcación DTMF. Ese tono de marcación se generó mediante la pulsación de la tecla número 2 del abonado llamante. En las líneas 19-22 puede observarse como la finalización de llamada la genera el abonado llamante y la gestiona el servidor VoIP de la UES mediante el mensaje **BYE**. El servidor VoIP de la EIE responde con el mensaje **200 ok**.

### 2.1.2 IAX.

IAX (de sus siglas en ingles Inter Asterisk eXchange). Fue creado por Mark Spencer (también creador de Asterisk) para solucionar una serie de problemas e inconvenientes al momento de utilizar SIP en VoIP[3].

El principal objetivo de IAX es minimizar el ancho de banda utilizado en la transmisión de voz y vídeo a través de la red IP y proveer un soporte nativo para ser transparente a los NATs (Network Address Translation). La estructura básica de IAX se potencia en la multiplexación de la señalización y del flujo de datos sobre un solo puerto UDP, generalmente el 4569. El protocolo original ha quedado obsoleto por su segunda versión conocida como IAX2. Se caracteriza por ser robusto y simple en comparación con otros protocolos. Permite manejar una gran cantidad de códecs y transportar cualquier tipo de datos.

Una llamada IAX o IAX2 tiene tres fases:

#### A) Establecimiento de la llamada.

El terminal A inicia una conexión y manda un mensaje **NEW**. El terminal llamado responde con un **ACCEPT** y el llamante le responde con un **ACK**. A continuación el terminal llamado da las señales de **RINGING** y el llamante contesta con un **ACK**. Para confirmar la recepción del mensaje. Por último, el llamado acepta la llamada con un **ANSWER** y el llamante confirma ese mensaje.

#### B) Flujo de datos o flujo de audio.

Se mandan los frames M y F en ambos sentidos con la información vocal. Los frames M son mini-frames que contienen solo una cabecera de 4 bytes para reducir el uso en el ancho de banda. Los frames F son frames completos que incluyen información de sincronización.

#### C) Liberación de la llamada o desconexión.

La liberación de la conexión es tan sencillo como enviar un mensaje de **HANGUP** y confirmar dicho mensaje con un **ACK**.

Para poder entender el protocolo IAX se realizó una prueba de laboratorio en el servidor VoIP EIE. Este consistió en el establecimiento de una llamada entre dos softphone (Zoiper) como se muestra en la Ilustración 3. Los Softphone están, preparados para sincronizar sus llamadas con el protocolo IAX. En las líneas 3 y 4 se observa, que la computadora con dirección IP 10.10.89.253 envía un mensaje **NEW** hacia la computadora con dirección IP 10.10.89.252. Y a su vez responde con una serie de mensajes **ACCEPT** como se aprecia en las líneas 19, 20, 25 y 26. En las líneas 29 - 32 se observa que el softphone llamante, envía una serie de mensajes **ACK** para confirmación y a su vez el softphone llamado responde con un mensaje **RINGING** para generar tonos de marcado. En la línea 39 y 40 se observa que la terminal llamada responde con un mensaje **ANSWER** para indicar que llamada telefónica se ha establecido.

1	Time	10.10.89.253	10.10.89.252
2			
3	35.104	NEW	
4		(4569) ----->	(4569)
5	35.104	unknown (0x28)	
6		(4569) <-----	(4569)
7	35.105	NEW	
8		(4569) ----->	(4569)
9	35.106	ACK	
10		(4569) <-----	(4569)
11	35.106	AUTHREQ	
12		(4569) <-----	(4569)
13	37.104	AUTHREQ	
14		(4569) <-----	(4569)
15	37.105	AUTHREP	
16		(4569) ----->	(4569)
17	37.105	ACK	
18		(4569) <-----	(4569)
19	37.105	ACCEPT	
20		(4569) <-----	(4569)
21	37.106	unknown (0x0e)	
22		(4569) <-----	(4569)
23	37.107	VNAK	
24		(4569) ----->	(4569)
25	37.107	ACCEPT	
26		(4569) <-----	(4569)
27	37.107	unknown (0x0e)	
28		(4569) <-----	(4569)
29	37.108	ACK	
30		(4569) ----->	(4569)
31	37.108	ACK	
32		(4569) ----->	(4569)
33	37.126	RINGING	
34		(4569) <-----	(4569)
35	37.127	ACK	
36		(4569) ----->	(4569)
37	37.149	ACK	
38		(4569) <-----	(4569)
39	41.839	ANSWER	
40		(4569) <-----	(4569)
41	41.839	stop sounds	
42		(4569) <-----	(4569)
43	41.839	unknown (0x14)	
44		(4569) <-----	(4569)
45	41.839	TXREQ	
46		(4569) <-----	(4569)
47	41.840	ACK	
48		(4569) ----->	(4569)
49	41.840	ACK	
50		(4569) ----->	(4569)
51	41.840	ACK	
52		(4569) ----->	(4569)
53	41.840	ACK	
54		(4569) ----->	(4569)
55	42.642	TXCNT	
56		(4569) ----->	(4569)
57	43.441	TXCNT	
58		(4569) ----->	(4569)
59	43.840	TXCNT	
60		(4569) <-----	(4569)
61	43.841	TXACC	
62		(4569) ----->	(4569)
63	43.841	TXCNT	
64		(4569) ----->	(4569)
65	43.843	TXREADY	
66		(4569) ----->	(4569)
67	43.843	ACK	
68		(4569) <-----	(4569)
69	43.844	TXREL	
70		(4569) <-----	(4569)
71	45.843	TXREL	
72		(4569) <-----	(4569)
73	45.843	ACK	
74		(4569) ----->	(4569)
75	45.848	Voice	
76		(4569) ----->	(4569)
77	45.850	ACK	
78		(4569) <-----	(4569)
79	45.852	Voice	
80		(4569) <-----	(4569)
81	45.852	ACK	
82		(4569) ----->	(4569)
83	51.407	HANGUP	
84		(4569) ----->	(4569)
85	51.408	ACK	
86		(4569) <-----	(4569)

Ilustración 3: Protocolo IAX.

Las líneas 83 y 84 indican que un mensaje **HANGUP** es enviado, hacia la terminal llamada y su vez responde con un mensaje **ACK** de confirmación como se muestra en las líneas 85 y 86 para terminar la llamada telefónica.

### **2.1.3 Comparación entre IAX Y SIP.**

Cabe mencionar que todo el flujo de datos y sincronización del protocolo IAX, se hace por un único puerto, normalmente el 4569. El utilizar un único puerto presenta algunas ventajas sobre el protocolo SIP[4].

#### **Ancho de banda.**

IAX utiliza un menor ancho de banda, ya que los mensajes son codificados de forma binaria mientras que en SIP son mensajes de texto.

#### **NAT.**

En IAX la señalización y los datos viajan conjuntamente con lo cual se evitan los problemas de NAT que frecuentemente aparecen en SIP. En SIP la señalización y los datos viajan de manera separada y por eso aparecen problemas de NAT en el flujo de audio. Cuando este flujo debe superar los routers y firewalls.

#### **Estandarización y uso.**

SIP es un protocolo estandarizado por la IETF, IAX está aún siendo estandarizado y es por ello que no se encuentra en muchos dispositivos existentes en el mercado.

#### **Utilización de puertos.**

IAX utiliza un solo puerto (4569) para mandar la información de señalización y los datos de todas sus llamadas. SIP, sin embargo utiliza un puerto (5060) para señalización y 2 puertos RTP por cada conexión de audio (como mínimo 3 puertos)

#### **Flujo de audio al utilizar un servidor.**

En SIP la señalización de control pasa siempre por el servidor pero la información de audio (flujo RTP) puede viajar extremo a extremo sin tener que pasar necesariamente por el servidor SIP. En IAX al viajar la señalización y los datos de forma conjunta todo el tráfico de audio debe pasar obligatoriamente por el servidor IAX. Esto produce un aumento en el uso del ancho de banda que deben soportar los servidores IAX sobretodo cuando hay muchas llamadas simultaneas.

#### **Otras funcionalidades.**

IAX es un protocolo pensado para VoIP y transmisión de vídeo y presenta funcionalidades interesantes como la posibilidad de enviar o recibir planes de marcado (dialplans) que resultan muy interesante al usarlo conjuntamente con servidores Asterisk. SIP es un protocolo de propósito general y podría transmitir sin dificultad cualquier información y no sólo audio o vídeo.

## 2.2 Codec de voz utilizados por Asterisk.

Los codecs son conjuntos de transformaciones utilizadas para digitalizar la voz. Estos convierten tanto la voz en datos como los datos en voz[5]. La elección de un codec depende de varios factores a tener en cuenta, como lo es la calidad de llamada, costo de licenciamiento, uso de banda ancha, resistencia a pérdidas de paquetes y necesidad de procesamiento basado en MIPS (millones de instrucciones por segundo), disponibilidad en Asterisk y en los teléfonos.

### 2.2.1 G711u y G729a.

Existen dos estándares los que se conocen como el micro-law (u-law) y el a-law. A estos estándares se les conoce también como G711u y G711a respectivamente[5].

El micro-law se usa normalmente en Norte-américa y el a-law en Europa. La familia de codecs G711 no requieren de gran procesamiento y por eso están disponibles en la mayoría de los equipos de VoIP.

El codec g.711 utiliza un ancho de banda de 64 kbps y no tiene un mecanismo de resistencia a pérdidas de paquetes por otra parte el g.729 reduce su ancho a 8 kbps pero tiene un costo de US \$10 por canal y tiene un mecanismo que reduce al 3% la pérdidas de paquetes, mientras que el g.711 es gratuito.

### 2.2.2 GSM .

El codec GSM (del inglés Global System of Mobile communications) proviene del sistema de comunicaciones móviles y es del tipo RPE-LTP (Regular Pulse Excitation Long-Term Prediction) Proporciona una tasa de 13kbps ofreciendo una buena calidad con gran simpleza de proceso para aplicaciones de tiempo real[6].

### 2.2.3 Método simplificado para el Cálculo de ancho de banda utilizado por Codecs.

Aunque los codecs utilizan muy poco ancho de banda, existe una sobrecarga causada por las cabeceras de paquetes IP, UDP y RTP de los paquetes de voz. Se puede decir que la necesidad de ancho de banda varía de acuerdo con los tipos de cabeceras requeridos. Por ejemplo, Si estamos en una red Ethernet tenemos que adicionar el encabezado Ethernet al cálculo, si estamos en una red WAN normalmente vamos a sumarle el encabezamiento Frame-Relay o PPP. Esto aumenta la cantidad de banda ancha utilizada en hasta incluso tres veces.

```
Ejemplo.  
  
codec g.711 (64 Kbps)  
Red Ethernet (Ethernet+ip+UDP+RTP+G.711)      = 87.2 Kbps  
Red PPP (PPP+ip+UDP+RTP+G.711)                = 82.4 Kbps  
Red Frame-Relay (FR+ip+UDP+RTP+G.711)         = 82.8 Kbps  
  
codec G.729 (8 Kbps)  
Red Ethernet (Ethernet+ip+UDP+RTP+G.729)      = 31.2 Kbps  
Red PPP (PPP+ip+UDP+RTP+G.729)                = 26.4 Kbps  
Red Frame-Relay (FR+ip+UDP+RTP+G.729)         = 26.8 Kbps
```

Ilustración 4: Ejemplo de cálculo del ancho de banda para codec.

Los cálculos de la Ilustración 4 fueron realizados con la herramienta que proporciona la comunidad de usuarios Asterisk guide<sup>2</sup> estos pueden ayudarnos para estimar el ancho de banda necesario en la implementación de una central Asterisk.

Una de las simplificaciones más usadas es estimar el número de llamadas simultáneas por tipo de usuario. Por ejemplo:

PBX en Empresas (1 llamada simultánea por 5 ramales).  
 PBX para Hotel (1 llamada simultánea para 10 ramales).

*Ilustración 5: Consideraciones para simplificar la cantidad de líneas telefónicas utilizadas.*

estas simplificaciones que se muestran en la Ilustración 5 tienen sentido solo si se ha estudiado el tráfico telefónico. En base a lo anterior podemos usar como ejemplo lo siguiente:

Empresa A:	30 ramales.
Empresa B:	15 ramales.
Hotel A:	20 ramales.

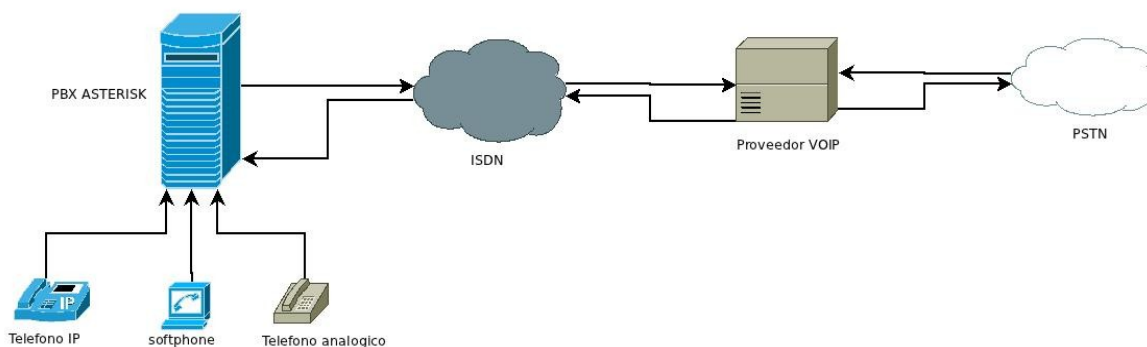
*Ilustración 6: Ejemplo para calcular el ancho de banda necesario.*

La Ilustración 6 muestra el comportamiento típico de dos PBX montadas en empresas y una PBX para dar servicio en un Hotel, con el objetivo realizar cálculos de ancho de banda. Por la simplificación tendríamos 2 llamadas simultáneas para Hotel A, 6 llamadas simultáneas en la Empresa A y 3 en la Empresa B. ahora suponiendo que se utiliza el codec g.729 en las filiales se puede estimar el ancho de banda necesario:

Ancho de banda necesaria para Empresa A (Frame-Relay):  $26,8 * 6 = 160,8$  Kbps.

Banda necesaria para a Empresa B (Frame-Relay): 80,4 Kbps.

## 2.3 Troncales SIP.



*Ilustración 7: funcionamiento de las troncales SIP*

Una troncal sip es el equivalente moderno de las troncales E1/T1 que en el pasado y aun en la actualidad se pueden comprar de un proveedor de telecomunicaciones para conectar PBX tradicionales. Con los nuevos avances

<sup>2</sup> <http://www.asteriskguide.com/bandcalc/bandcalc.php>

tecnológicos en materia VoIP es posible conectar PBX Asterisk hacia un proveedor VoIP con el objetivo de enrutar todas las llamadas hacia un destino fuera de la red interna[5].

En la Ilustración 7 se puede observar como una central Asterisk crea una ruta para el tráfico de llamadas por medio de la ISDN para registrarse a un proveedor VoIP, con el objetivo de tener acceso a la red telefónica pública PSTN. Las troncales SIP están basadas en el estándar SIP. El término "troncal", se refiere a un grupo de líneas telefónicas interconectadas con un proveedor con el objetivo de enrutar hacia dentro y hacia afuera todas las llamadas telefónicas y así lograr una conexión entre la red pública telefónica.

En las PBX Asterisk, una troncal SIP se registra como si fuera un cliente para otro servidor SIP y luego la recepción de llamadas es enrutada hacia una extensión Asterisk logrando así una buena calidad de servicio (QoS).

## **2.4 Tipos de tecnologías en interfaces de voz.**

en las interfaces de voz existen diversas tecnologías que ayudan a crear una interacción entre el usuario y una contestadora automática a continuación se citan algunas.

### **2.4.1 Detección de tonos DTMF.**

El esquema de marcado DTMF (del inglés Dual tone multi frequency signaling) fue diseñado por los laboratorios Bell e introducido en los Estados Unidos a mediados de los años 60's como una disposición del marcado fácil por pulsos mejorando la fiabilidad en la sincronización de un enlace punto a punto.

básicamente la utilización de este método funciona así, el usuario escucha una voz previamente grabada que le emite instrucciones, luego se pulsa una tecla del terminal telefónico para elegir una de las opciones, finalmente el sistema reconoce la opción y crea una acción previamente definida.

### **2.4.2 Reconocimiento de voz ASR.**

El reconocimiento ASR (del inglés Automatic Speech Recognition) permite convertir la voz humana en texto por medio de un software específico, para ser traducido a un comando y ejecutar una acción. Así por ejemplo cuando el usuario se conecta a una IVR este será atendido para escoger las opciones.

### **2.4.3 Síntesis de voz TTS.**

TTS (del inglés Text to Speech) es un proceso en el cual un texto es convertido a voz artificial por medio de software, en tiempo real. A diferencia de las grabaciones, los TTS son ejecutados en tiempo real.

La voz que oye el usuario no está grabada, es voz sintetizada que además es muy útil para usarla con respuestas que son variables en los sistemas IVR, útil para dar respuestas con valores variables.



## **2.5 Definición de una IVR.**

Permite la comunicación con datos de un servidor a través de los tonos generados por el teléfono y además, permite la administración de llamadas creando un nuevo medio de información interactivo.

Un IVR montado en Asterisk nos da la facilidad de poder conectarnos remotamente o internamente hacia base de datos tales como MySQL o PostgreSQL, Oracle, SQL Server. También podemos crear conexiones con líneas analógicas FXO o troncales VoIP ya sea SIP o IAX. Esta plataforma de hardware y software libre nos permite solventar con mucho margen de ganancia las necesidades de comunicación, por ser tan robusta y adaptable a las redes de computadoras existentes como por ejemplo en algún ministerio de operaciones publicas o alguna empresa privada.

## **2.6 Principios generales en el diseño.**

algunos de los principios generales para el diseño de las IVR podrían ser los que a continuación se presentan, cabe mencionar que no existe una bibliografía o manuales de diseño que nos puedan, guiar en la construcción de estos sistemas por lo tanto las siguientes recomendaciones han sido una recolección de las experiencias de una empresa dedicada a este rubro<sup>3</sup>.

- Proporcionar solo la información que se necesita y de la forma más simple.
- Diseñar para la mayoría de los usuarios y evitar que tengan que recorrer opciones que sólo interesan a pocos.
- Hacer que el sistema trabaje para el usuario, por ejemplo, no pidiendo el mismo dato varias veces.
- Ser consistente y no cambiar a menudo las opciones de los menús.
- El saludo inicial debe ser breve y no repetirlo si el usuario vuelve al inicio.
- Evitar mensajes promocionales, se deben poner solo si el usuario lo puede encontrar justificado y además estos deben ser muy breves.
- El menú principal debe durar como máximo 20 segundos contando el saludo y el menú principal.
- Pedir identificación solo si es necesario, además debe ser el más fácil por ejemplo Registro clínico o DUI normalmente utilizados para acceder a información en servidores de datos.
- Los Menús Debe estar provistos de un breve recordatorio, título del menú que indica al usuario donde ha ido a parar y de qué tratan las opciones que escuchó Ejemplo: "Estado de su Cuenta". Máximo de 3 pasos para hacer una tarea. poner las opciones de navegación al final de todo.

---

3 El Salvador Network SALNET.

## **2.7 Principales errores de ergonomía.**

- Menús demasiado largos. Existen algunas aplicaciones de contestadoras automáticas que prolongan sus menús tanto que el usuario, puede llegar a perder la noción del lugar donde se encuentra y lo que es peor olvidar el objetivo de su llamada.
- Los usuarios se pierden o no encuentran lo que necesitan. Esto debido a la mala redacción de los menús o por saturación de los mismos.
- No comprenden como usar el sistema. Debido a que el sistema no cuenta con recordatorios o indicaciones de su buen uso.
- Lógica confusa y "callejones sin salida". Por falta de opciones de navegación y validación de datos.
- Ineficiente recuperación ante fallos. El sistema debe ser provisto de validación de datos que le permita reaccionar a eventos que no están establecidos en los menús.
- Falta de canales de atención al cliente. Muchas veces la falta de atención personalizada hacia el usuario por parte de operadores humanos, hace que el sistema se vuelva ineficiente.

## **2.8 Cantidad óptima de menús a presentar al usuario.**

Un estudio realizado por Georgiano Millar en el año 1956 llamado, "*El número mágico siete, más o menos dos: Algunos límites sobre nuestra capacidad de procesar información*", describe el hecho que la mayoría de las personas sólo pueden "procesar" alrededor de siete elementos de información en forma simultánea. Dependiendo del tipo de información y de características del propio individuo, a veces esta habilidad se reduce a cinco y en algunos pocos casos se eleva hasta nueve. Las limitaciones de la memoria a corto plazo son más evidentes ante estímulos auditivos y presentados en forma secuencial. más adelante casi un poco más de 50 años después, este mismo principio crea influencia de manera importante el diseño de servicios o sistemas que utilizan la inserción de menús tales como la programación de paginas web y por supuesto en los IVR.

Ahora, *¿de qué manera este número afecta el diseño de sistemas con menús.?* Si tenemos un sistema cuyo menú principal cuenta con 6 opciones no tendremos problemas, ya que según esta regla, el usuario los podrá memorizar rápidamente. Pero si tenemos un sistema cuyo menú principal, por ejemplo, lo forman 15 opciones, estaríamos frente a un problema. Los usuarios no podrán memorizar cada opción del menú rápidamente haciendo que su experiencia en el sistema sea dificultosa.

En casos como el anterior, cuando el menú cuenta con demasiadas opciones, lo aconsejable es agruparlas dejando un máximo de 9 (7 más o menos 2) y el resto dejarlas como sub-opciones de algunas de las 9 principales.

La regla del "7 +-2" con el tiempo ha pasado a ser parte de la mayoría de las pruebas heurísticas que se realizan al momento de verificar la calidad de un diseño de servicios los cuales incluyen menús al usuario final.

## **2.9 Opciones de navegación recomendables.**

- Volver atrás. Utilizada cuando se a llegado a un sub-menu el cual no presenta la información que el usuario desea y este debe retornar al menú anterior.
- Volver a menú principal. Utilizada cuando el usuario desea regresar directamente al menú principal.
- Repetir. Esta opción puede usarse cuando el sistema proporciona información al usuario y el desea que se le sea repetida.
- Opciones de desvió hacia un operador. Si existe una opción de paso al operador, debe ser la última
- validación de datos. No hay que cortar nunca la comunicación si el usuario se equivoca. Es preciso dar caminos alternativos.
- Preguntas si/no. Se usan para recuperar errores y para confirmar acciones estas tienen que ser preguntas directas que induzcan a decir de forma natural sólo Sí o No.

## CAPITULO III. INSTALACIÓN DEL SOFTWARE.

### 3.1 Instalación de Asterisk 1.8 en Ubuntu 11.04 LTS.

En este proyecto de graduación, se trabajo con Ubuntu 11.04 LTS debido a que es la versión mas estable en este momento.[7] La actualización se realizo mediante la ejecución de los siguientes comandos.

```
sudo apt-get update
sudo apt-get dist-upgrade
```

*Instalación 1: Actualización del sistema operativo.*

Luego se debe reiniciar el servidor para que los cambios se hagan efectivos. Las dependencias necesarias de Asterisk 1.8 y todos sus componentes se instalan por medio de la ejecución del siguiente comando.

```
sudo apt-get install build-essential linux-headers-`uname -r` subversion libncurses5-dev libssl-dev libxml2-dev
vim-nox gawk wget apache2
```

*Instalación 2: Dependencias de Asterisk.*

Con las dependencias ya instaladas, se necesita descargar los archivos necesarios para la instalación correcta. El primer archivo a descargar sera la versión 1.8. de Asterisk, se crea una carpeta en donde la descarga se realiza y seguidamente se descomprime.

```
mkdir -p ~/src/asterisk-complete/asterisk
cd ~/src/asterisk-complete/asterisk
wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-1.8-current.tar.gz
tar zxvf asterisk-1.8-current.tar.gz
```

*Instalación 3: Descarga de paquetes Asterisk.*

Las siguientes lineas de comandos sirven para satisfacer las dependencias de los módulos y aplicaciones que se desea incluir posteriormente a la instalación, como por ejemplo los módulos res\_odbc.so y func\_odbc.so que nos permitirá crear conexiones con bases de datos MySQL.

Nota : durante la instalación el sistema podría pedir introducir el codigo telefónico del país el cual para nuestro país es el 503.

```
cd ~/src/asterisk-complete/asterisk/asterisk-1.8.4.1/
sudo ./contrib/scripts/install_prereq install
sudo ./contrib/scripts/install_prereq install-unpackaged
```

*Instalación 4: Dependencias y módulos Asterisk.*

La Instalación 1,2,3 y 4 realizan la actualización del sistema operativo, instalación de dependencias de Asterisk, descarga de software Asterisk e instalación de las dependencias para los módulos, respectivamente. Hasta este punto aun no se ha instalado Asterisk. Previo a ello se instalaran las aplicaciones LIBPRI que es una implementación escrita en Lenguaje C, que contiene la especificación ISDN, y permite configurar canales T1/E1 utilizando hardware digium y DAHDI que contiene controladores, librerías, programas y herramientas para interactuar con hardware digium.

Por lo antes mencionado se debe tener en cuenta que la secuencia de

instalación sera primeramente LIBPRI, luego DAHDI y finalmente Asterisk. Se hace en este orden para garantizar que los script no tengan ningún problema a la hora de verificar las dependencias de los módulos integrados.

### 3.1.1 LIBPRI.

Para instalar LIBPRI creamos una carpeta, descargamos los paquetes previamente compilados para Ubuntu mediante la ejecución de los siguientes comandos.

```
cd ~/src/asterisk-complete/  
mkdir libpri  
cd libpri/  
svn co http://svn.asterisk.org/svn/libpri/tags/1.4.11.5  
cd 1.4.11.5  
make  
sudo make install
```

*Instalación 5: Instalación de LIBPRI.*

### 3.1.2 DAHDI.

Para instalar DAHDI al igual que el paso anterior creamos una carpeta y descargamos los paquetes previamente compilados con la ejecución de los siguientes comandos.

```
cd ~/src/asterisk-complete/  
mkdir dahdi  
cd dahdi/  
svn co http://svn.asterisk.org/svn/dahdi/linux-complete/tags/2.4.1.2+2.4.1/  
cd 2.4.1.2+2.4.1  
make  
sudo make install  
sudo make config
```

*Instalación 6: Instalación de DAHDI.*

### 3.1.3 Asterisk.

Ahora llegamos a la instalación de Asterisk para lograrlo se debe ejecutar los siguientes comandos en el servidor.

```
cd ~/src/asterisk-complete/asterisk/asterisk-1.8.4.1/  
./configure  
contrib/scripts/get_mp3_source.sh  
make  
sudo make install  
sudo make config
```

*Instalación 7: Instalación de Asterisk.*

Opcionalmente se puede instalar los ejemplos de archivos para la configuración y documentación .

```
sudo make samples  
sudo make progdocs
```

*Instalación 8: Instalación de ejemplos para la configuración en Asterisk.*

Finalmente se debe cambiar los permisos de algunos directorios para el buen funcionamiento y seguridad de los directorios que Asterisk utiliza como configuración del sistema.

```
sudo chown asterisk. /var/run/asterisk
sudo chown -R asterisk. /etc/asterisk
sudo chown -R asterisk. /var/{lib,log,spool}/asterisk
sudo chown -R asterisk. /var/www/
sudo /etc/init.d/asterisk start
```

### *Instalación 9: Validación y Permisos para ejecutar Asterisk.*

En la Instalación 9 podemos ver en la primera línea el directorio `/var/run/asterisk` es cambiado de propietario mediante el comando `chown` con el objetivo de que el sistema pueda acceder a ese directorio sin tener privilegios de super usuario "root", al contrario del los siguiente comandos que modifican los propietarios para ser usados, solo por el súper usuario root y el ultimo comando `sudo /etc/init.d/asterisk start` sirve para iniciar el servicio Asterisk siempre que el equipo encienda o reinicie.

## **3.2 Instalación de MySQL.**

La instalación de MySQL es necesaria debido a que se pretende, proporcionar información de los pacientes tales como: citas, referencias médicas y servicio de farmacia. Estos datos serán almacenados en bases de datos. Y se trasladaran a los pacientes del HNR por medio de la IVR en el Menú 3. Registro clínico.

Para instalar MySQL en Ubuntu 11.04 [8] se debe instalar los siguientes paquetes `mysql-server` y `mysql-client`. Mediante el siguiente comando.

```
sudo apt-get install mysql-server mysql-client
```

### *Instalación 10: Instalación de MySQL.*

Pedirá crear la contraseña de administrador root cuando se instale el programa, pero si no hace la petición, se puede crear de la siguiente manera.

```
sudo /usr/bin/mysqladmin -u root password clavenueva
```

### *Instalación 11: Contraseña root para MySQL.*

Luego de configurar la contraseña debemos comprobar la conexión como se muestra en la Instalación 12 esto para verificar que la instalación de MySQL fue efectiva. Las bases de datos definidas en el sistema contendrán información de cada paciente del HNR y estas podrán ser consultadas con el plan de marcado de Asterisk por medio del numero de registro proporcionado por el HNR.

```
mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.1.62-0ubuntu0.11.04.1 (Ubuntu)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

### *Instalación 12: Accediendo a consola de MySQL.*

### 3.3 Instalación de unixODBC.

Como ya antes se menciona, existe la intención de proporcionar la información que esta almacenada en bases de datos a los pacientes, para lograr esto, después de haber instalado MySQL (3.2 Instalación de MySQL.) se necesita crear una conexión entre el plan de marcado de Asterisk y el gestor de bases de datos. Por lo que es necesario la instalación de unixODBC.

unixODBC nos permite conexiones hacia BD, en este caso se necesita tener una conexión hacia MySQL, por lo tanto se instalara el software y el driver libmyodbc de la siguiente manera[9].

```
sudo apt-get install unixodbc unixodbc-dev libmyodbc
```

*Instalación 13: Instalación de unixODBC.*

unixODBC permite conexiones con muchos tipos de bases de datos por ejemplo con postgresQL, simplemente con la instalación del driver especifico. Si se desea crear conexiones hacia bases de datos del tipo postgresQL el driver que debe instalarse es psqldbca.

### 3.4 Instalación de Festival.

Festival es un software text-to-speech (TTS) o sintetizador de voz que permite convertir texto a voz humana de manera sintética, por lo tanto es de mucha utilidad para poder proporcionar datos extraídos desde una base de MySQL. Su manera correcta de instalación es la siguiente[10].

```
sudo apt-get install festival festival-dev
```

*Instalación 14: Festival y dependencias.*

Festival viene configurado por defecto con el idioma ingles para poder utilizarlo con el idioma español tenemos que seguir los siguientes pasos.

```
cd /usr/share/festival/voices
wget http://www.voztovoice.org/tmp/festival-spanish.zip
sudo apt-get install unzip
sudo unzip festival-spanish.zip
```

*Instalación 15: Voces en español para Festival.*

Y modificar el siguiente archivo de Festival vi /usr/share/festival/festival.scm añadimos estas lineas.

```
;(language__spanish)
(set! voice_default 'voice_el_diphone)
(define (tts_textasterisk string mode)
(let ((wholeutt (utt.synth (eval (list 'Utterance 'Text string))))))
(utt.wave.resample wholeutt 8000)
(utt.wave.rescale wholeutt 5)
(utt.send.wave.client wholeutt)))
```

*Instalación 16: Archivo de configuración para voces en español Festival.*

## CAPITULO IV. IMPLEMENTACIÓN DEL SERVICIO IVR PARA EL HNR.

### 4.1 Servicio IVR para HNR.

Para el diseño de la IVR que dará el servicio en el HNR, se ha tomado en cuenta todas las consideraciones que se mencionaron en la pag 13 . Con el objetivo de brindar un buen servicio a los pacientes. A continuación se describe su funcionamiento.

#### Menú principal.

El servicio cuenta con un saludo de bienvenida para presentar la institución, luego presenta un menú principal con cuatro opciones:

- Menú 1. Cirugía ambulatoria.
- Menú 2. Laboratorio clínico.
- Menú 3. Registro clínico.
- Menú 4. Atención personalizada por operadores.

Es de esta forma que el usuario a través de un tono DTMF generado desde su teléfono puede acceder a las opciones del menú principal, marcando la tecla desde la la opción 1 hasta la 4. En este menú principal como en los otros menús, existe una validación de datos, es decir si el usuario marca una opción que no esta presentada en el menú, el sistema envía un mensaje de alerta al usuario, retornando de nuevo a la presentación de las opciones, sin mostrar el saludo de bienvenida y dándole la oportunidad de elegir una valida.

En la Ilustración 8. Podemos observar que el primer mensaje es el saludo de bienvenida, luego se presenta el **Menú principal**. Teniendo en cuenta que su duración debe ser breve y que si el usuario proporciona datos erróneos, debe contar con un bloque de validación, permitiendo regresar al menú principal sin presentar el saludo de bienvenida.

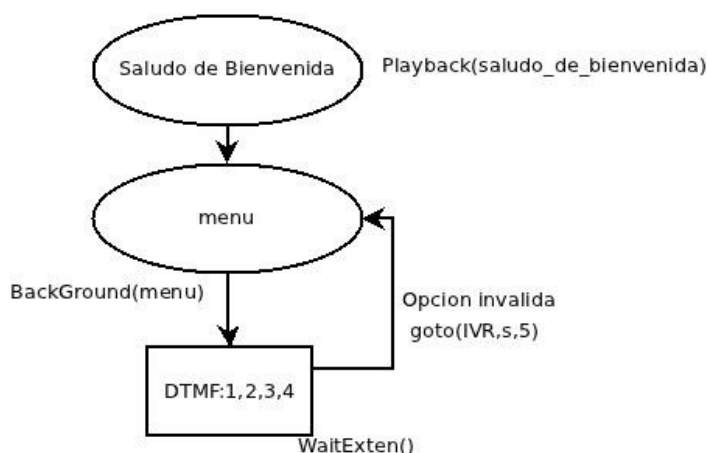


Ilustración 8: Menú principal del servicio IVR.



## Menú 1. Cirugía ambulatoria.

Como se muestra en la Ilustración 9 el menú 1 tiene cuatro opciones acerca de información de cirugía ambulatoria tales como:

1. Cuidados previos a la cirugía.
2. Normas después de la operación.
3. Cuidados en casa.
4. Acerca de signos de alarma.

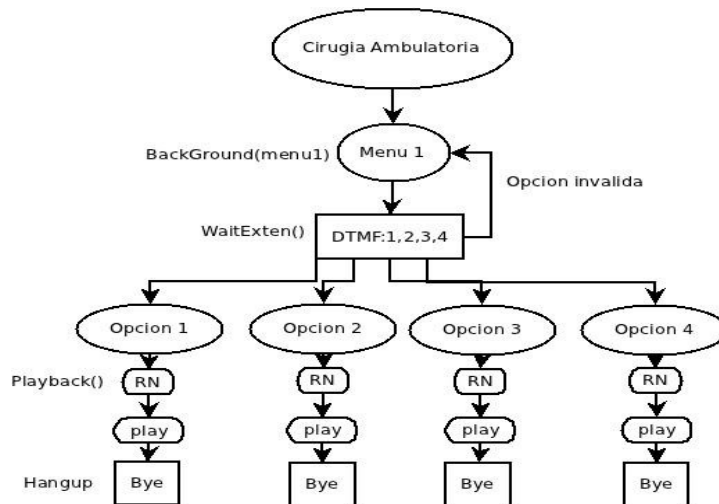


Ilustración 9: Menú 1 para proporcionar información de cirugía ambulatoria.

Al igual que el menú principal también cuenta con una validación de datos, es decir cuando el paciente marca una opción no válida el sistema retorna al menú 1 dando la oportunidad de elegir una opción válida. Entonces si el usuario quiere acceder a una de las opciones basta con marcar un tono DTMF indicando la opción de su preferencia (bloque DTMF en la Ilustración 9) seguidamente le será proporcionada la información que solicita, a través de una grabación. Cabe mencionar que el sistema le indicara hacia donde ha llegado con un pequeño recordatorio (bloque RN) y luego de esto, un archivo previamente grabado empezara a reproducirse (bloque play). Cuando la reproducción termine, un pequeño mensaje de despedida indicara que la información a terminado (bloque bye).

## Menú 2. Laboratorio clínico.

Como se muestra en la Ilustración 10 El menú 2 presenta nueve opciones acerca de información para realizar efectivamente pruebas del laboratorio clínico:

1. Muestra de sangre.
2. Uro-cultivo, o examen general de orina.
3. Creatinina y proteínas en orina.
4. Curva de tolerancia a la glucosa.
5. Sangre oculta en heces.

6. Cultivo de hongos en uñas.
7. Muestra, para baciloscopia, o esputo.
8. Cultivo bacteriológico de lesión.
9. Cultivo faríngeo.

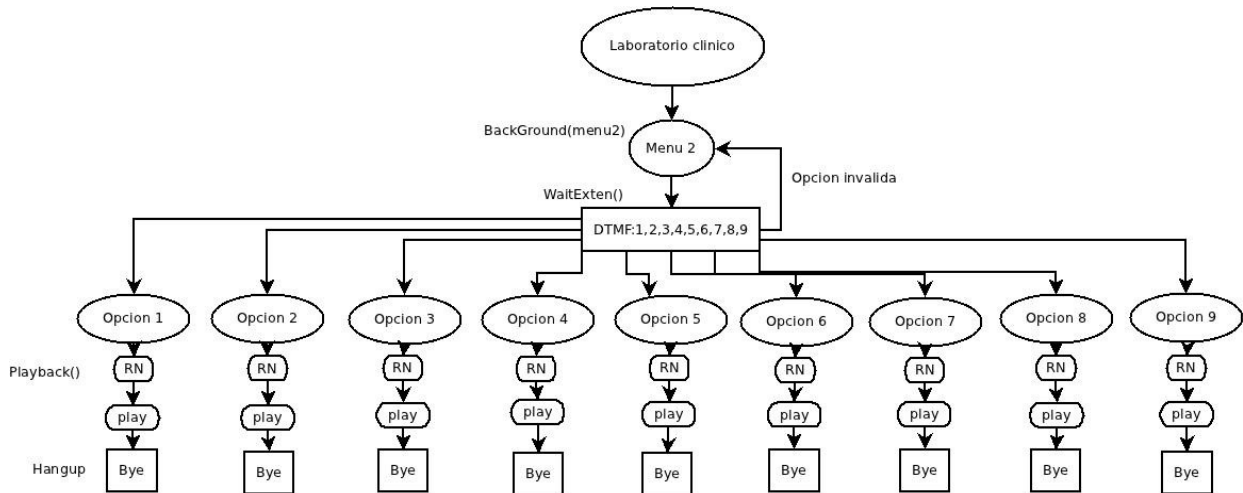


Ilustración 10: Menú 2 para proporcionar información de Laboratorio Clínico.

Este menú 2 es similar al anterior, también cuenta con validación de datos, recordatorios (bloque RN) y mensaje de despedida (bloque Bye).

### Menú 3. Registro clínico.

El menú 3, permite al usuario acceder a su registro clínico por medio de consultas a bases de datos definidas en el servidor. Esto se logra proporcionando su número de registro. al igual que en los menús anteriores también cuenta con validación de datos y en el se puede tener acceso a información como: referencias clínicas, citas programadas y servicio de farmacia. Este tipo de datos son variables es decir cambian según el registro de cada paciente por lo tanto, se hace uso de la tecnología "TTS" antes mencionada en la pagina 12, para sintetizar de texto a voz humana, las consultas hechas por el plan de marcado de Asterisk a las bases de datos.

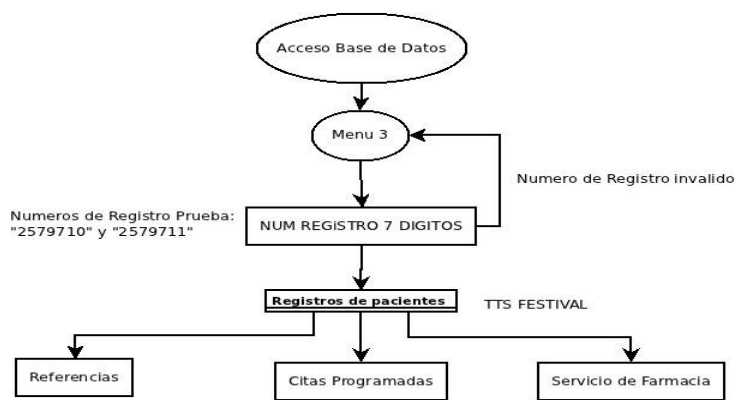
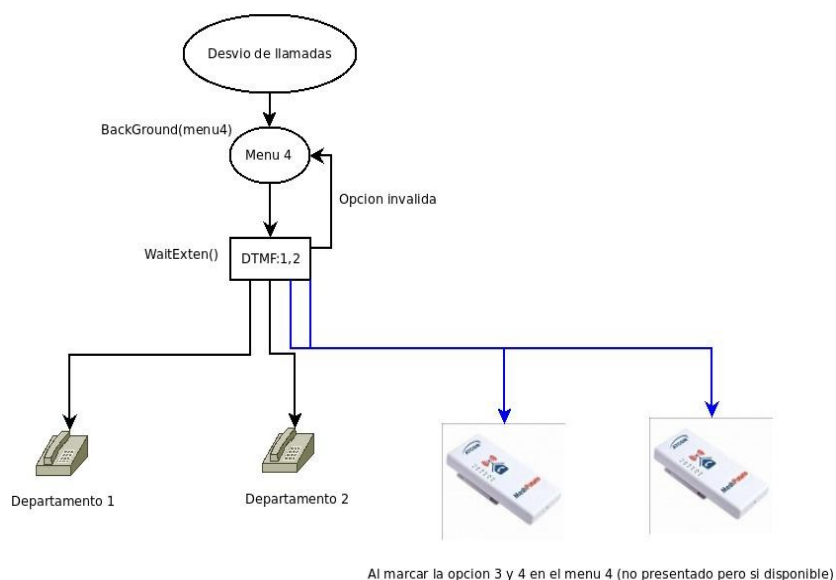


Ilustración 11: Menú 3 para proporcionar información de registro clínico.

La Ilustración 11 muestra el proceso cuando el usuario se encuentra en el menú 3 de la IVR. Primeramente se le pedirá proporcionar su numero de registro, luego se hará la consulta hacia las bases de datos definidas en el servidor y si es correcto el registro proporcionado, la IVR trasladara la información respectiva. Si no es correcto el bloque indicara que el dato es erróneo y permitirá al usuario ingresar un dato valido.

#### **Menú 4. Atención personalizada por operadores.**

El menú 4 como se muestra en la Ilustración 12, es un servicio de desvío para las llamadas hacia un operador. El cual es implementado para dar atención especializada a los pacientes, en este menú el sistema estará a la espera de dos tipos de tonos DTMF 1 y 2. Para luego, desviar la llamada telefónica al “departamento 1” o al “departamento 2” según sea el caso, también facilita opción al usuario para que marque una extensión si sabe el numero del departamento.



Al marcar la opcion 3 y 4 en el menu 4 (no presentado pero si disponible)  
 Ilustración 12: menú 4 para desvío de llamada a operadores internos.

### **4.2 Diseño de la Red Telefónica.**

En la Ilustración 13. vemos un diagrama de la red telefónica. Este consiste, en montar un servidor VoIP Asterisk sobre una red local LAN, 10.10.89.0/24 de la red privada en la Escuela de Ingeniería Eléctrica (EIE). Este servidor tiene una dirección IP 10.10.89.99. y una IP virtual 10.30.1.3 que servirá para tener conexión hacia una red inalámbrica WIFI<sup>4</sup> que más adelante se explicara. Esto se logra modificando el archivo /etc/network/interfaces con el código que se muestra en la Configuración 1.

4 Esta red sera utilizada para extender el servicio en los equipos comunitarios de salud (ECOS) ubicados en zonas rurales.

```

auto eth0:1
iface eth0:1 inet static
address 10.30.1.3
netmask 255.255.255.0
gateway 10.30.1.1

;luego de modificar el archivo interfaces ejecutar el siguiente comando
sudo /etc/init.d/networking restart

```

### Configuración 1: Tarjeta de red.

Y para comprobar que se ha configurado bien la tarjeta de red, podemos ejecutar el comando `ifconfig` como se muestra en la Configuración 2 de la siguiente manera.

```

ifconfig

eth0:1  Link encap:Ethernet direcciónHW 00:1a:92:fb:28:b2
        Direc. inet:10.30.1.3 Difus.:10.255.255.255 Másc:255.0.0.0
        ACTIVO DIFUSIÓN MULTICAST MTU:1500 Métrica:1
        Interrupción:16 Dirección base: 0xd800

```

### Configuración 2: Interfaces eth0 en servidor Asterisk.

Luego hay que asegurarse que tenemos conexión con un proveedor de VoIP, en este proyecto se utilizó el servidor Asterisk de la UES, el cual posee la dirección IP pública 168.243.7.190. La forma de verificar la conexión es por medio de un comando ping como se muestra en la Configuración 3.

```

# ping 168.243.7.190
PING 168.243.7.190 (168.243.7.190) 56(84) bytes of data.
64 bytes from 168.243.7.190: icmp_req=1 ttl=57 time=34.3 ms
64 bytes from 168.243.7.190: icmp_req=2 ttl=57 time=33.7 ms
64 bytes from 168.243.7.190: icmp_req=3 ttl=57 time=32.1 ms
64 bytes from 168.243.7.190: icmp_req=4 ttl=57 time=29.9 ms
64 bytes from 168.243.7.190: icmp_req=5 ttl=57 time=28.8 ms

```

### Configuración 3: Prueba de conectividad con proveedor VoIP.

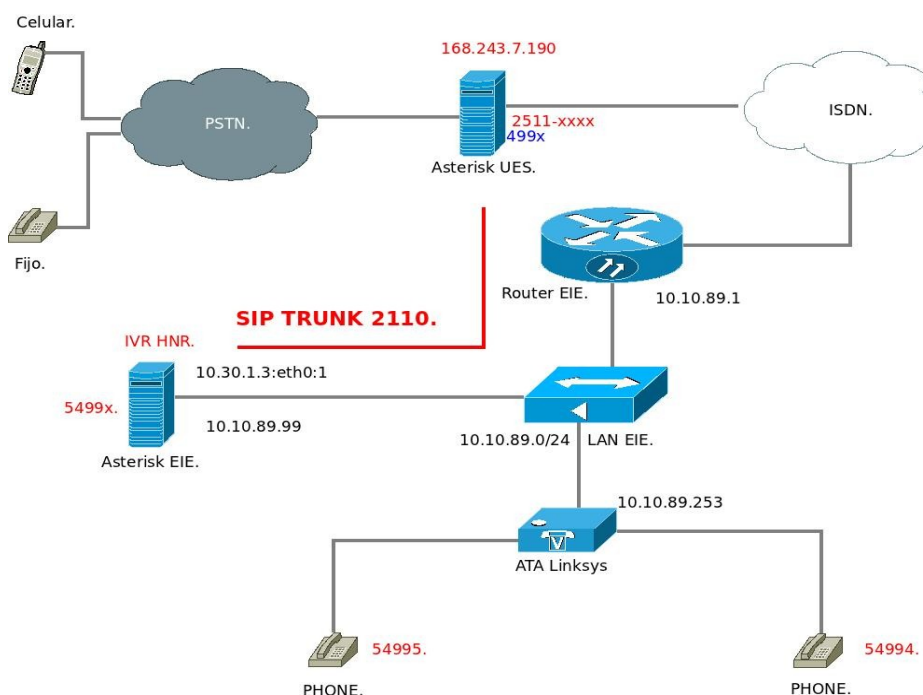


Ilustración 13: Diseño de la red telefónica.

Hay que aclarar que el proveedor VoIP (en este caso el servidor Asterisk de la UES.) asigno para este trabajo de graduación dos extensiones internas (4996, 4998) y una línea troncal con acceso a la red publica PSTN (2511-2110).

En la Ilustración 13 podemos observar una conexión troncal SIP con la extensión 2110 hacia el servidor VoIP UES permitiendo enrutar las llamadas telefónicas dentro y fuera de la red hacia la PSTN. También se instalo un ATA (de sus siglas en ingles Analog Telephony Adapter) para registrar dos extensiones internas (54994 y 54995).

### **4.3 Configuración de la troncal SIP en servidor Asterisk UES.**

Como se menciona antes en la pagina 11 acerca del concepto de troncales SIP. Estas nos permiten configurar un canal, en el cual se encaminara el trafico telefónico del sistema. Para configurar una troncal SIP en Asterisk primero es necesario seguir cierto procedimiento.[5]

**Paso 1.** Dentro del archivo sip.conf en la sección “general” se coloca una línea de registro del proveedor SIP.

**Paso 2.** Crear una entrada de tipo friend es decir que acepta y realiza llamadas siempre en el archivo sip.conf.

**Paso 3.** Crear un plan de marcado que de especifique una ruta para las extensiones internas.

1. Siguiendo el procedimiento anterior en el archivo (etc/asterisk/sip.conf) dentro del contexto [general] en la sección OUTBOUND SIP REGISTRATIONS agregar la siguiente línea.

```
register => 2110:ie22110@168.243.7.190
```

#### *Configuración 4: Troncal SIP 1.*

En la Configuración 4. La línea que registra la troncal SIP, contiene primeramente la extensión 2110 y seguidamente su contraseña ie22110 finalmente seguido del carácter @ tenemos la dirección IP 168.243.7.190 correspondiente al proveedor VoIP.

2. Siempre dentro del archivo sip.conf al final de este agregar el siguiente código.

```
[2110]
host=168.243.7.190
secret=ie22110
username=2110
fromuser=2110
insecure=port,invite
type=friend
disallow=all
allow=gsm,ulaw,alaw
dtmfmod=rfc2833
qualify=yes
canreinvite=no
nat=yes
context=IVR
```

#### *Configuración 5: Troncal SIP 2.*

En la Configuración 5. registramos la troncal SIP con los siguientes parámetros:

host 168.243.7.190, luego secret es la contraseña, username y fromuser es 2110.

Nota: el contexto IVR contiene todo el plan de marcado, para las llamadas entrantes, éste sera explicado con más detalle en la pagina 33 sección 4.7 correspondiente al plan de marcado de las llamadas entrantes.

3. luego dentro de una consola de Linux digitar los siguientes comandos para recargar los archivos sip.conf en donde se define los números telefónicos para las extensiones internas de la red.

```
# asterisk -rrrr
CLI> sip reload
Reloading SIP
== Parsing '/etc/asterisk/sip.conf': == Found
== Parsing '/etc/asterisk/users.conf': == Found
[May  3 15:27:52] NOTICE[1761]: chan_sip.c:20794 handle_response_peerpoke: Peer '2110' is now Reachable. (2ms / 2000ms)
```

#### *Configuración 6: Cargando la configuración de troncal SIP.*

La Configuración 6 muestra como cargar al servidor las configuraciones antes registradas, luego con el comando sip reload dentro del CLI de Asterisk pasamos las nuevas configuraciones, la ultima línea nos indica que la troncal SIP se ha registrado con el proveedor VoIP (“Peer '2110' is now Reachable”).

Para verificar podemos ejecutar el comando sip show peers siempre dentro del CLI de Asterisk.

```
CLI> sip show peers
Name/username      Host                Dyn Forcerport ACL Port      Status
2110/2110          168.243.7.190      N                5060      OK (2 ms)
```

#### *Configuración 7: Prueba de estado para la troncal SIP.*

En la Configuración 7 vemos la respuesta al comando anterior, la primera columna nos muestra el username 2110 se ha registrado(ok) al host 168.243.7.190 por el puerto 5060.

## **4.4 Configuración de extensiones SIP internas.**

En la Ilustración 13 podemos observar que se utiliza un ATA con la dirección IP 10.10.89.253 con dos extensiones internas (54994 y 54995). En este trabajo de graduación se utilizo el ATA PAP2T de linksys, para mas referencias consultar el apéndice C en la pag 50. Para registrar las extensiones en el servidor Asterisk (EIE), se debe modificar el archivo etc/asterisk/sip.conf. Las extensiones internas son cinco: 54994, 54995, 54996, 54997, 54998. Las últimas tres son utilizadas por los router inalámbricos MP (Mesh Potato) que nos permitirá adaptar una red inalámbrica para un escenario rural<sup>5</sup>. Entonces para registrar todas las extensiones, debemos agregar un bloque parecido al de la Configuración 8 por cada número telefónico interno. Donde “X” se debe cambiar por 4, 5, 6, 7, 8, 9 respectivamente. En total tendremos 5 bloques parecidos al anterior, al final del archivo sip.conf. Hasta el momento se ha configurado la troncal SIP y las extensiones internas de la red telefónica.

5 Los routers MP01 son una iniciativa de la fundación Village Telco [11].

Nota: el contexto extensiones de la Configuración 8 contiene el plan de marcado para la red interna así como también permitirá enrutar las llamadas dentro y fuera de la red este contexto se explicara en la pagina 32 que refiere al plan de marcado en la red interna.

```
[5499X]
type=friend ; puede recibir y realizar llamadas
username=5499X ;
secret=54994X ; contraseña para cada extensión
nat=yes ; El teléfono está nateado
host=dynamic ; la IP de cada cliente SIP puede cambiar
canreinvite=no ;
qualify=200 ; Tiempo de 200 ms para recibir respuesta
disallow=all ; deshabilita todos los codec
allow=ulaw ; habilita codec del tipo ulaw
allow=alaw ; habilita codec del tipo alaw
allow=g729 ; habilita el codec g729
context=extensiones ; plan de marcado
```

Configuración 8: Extensiones internas, archivo sip.conf.

#### 4.5 Configuración de MySQL, unixODBC y Festival para acceder a bases de datos.

En la Ilustración 9 mostré el menú principal de la IVR. Éste contiene una opción que proporciona información de registro clínico de los pacientes. Por ejemplo, referencias, citas programadas y servicio de farmacia. Para lograr acceder a esos datos se debe crear una conexión entre Asterisk y MySQL por medio de unixODBC[12].

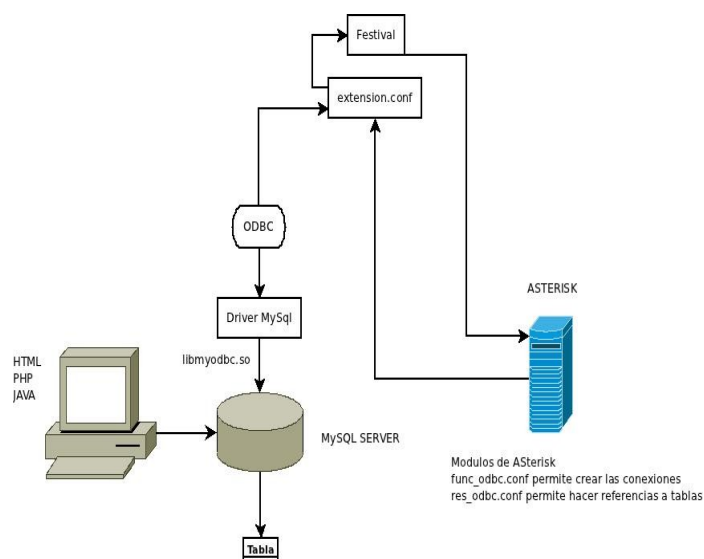


Ilustración 14: Conexión entre Asterisk y MySQL.

En la Ilustración 14 se muestra el proceso para hacer consultas hacia la bases de datos. Asterisk ejecuta consultas hacia MySQL a través de la conexión

unixODBC. Esto se realiza por medio del plan de marcado definido en el archivo `extension.conf`.

Luego esas consultas se pasan en modo de caracteres de texto hacia el sintetizador Festival, para ser convertidos en voz humana y así el usuario podrá escuchar la información de su registro clínico.

Los módulos que permiten a Asterisk interactuar con unixODBC son **res\_odbc** en el cual se registran los DSN (data source name) y **func\_odbc** que permite programar funciones para hacer las consultas a las bases de datos.

#### 4.5.1 Base de datos para los registros de pacientes.

Como ejemplo, para analizar la iteracción entre MySQL y Asterisk se contruye una bases de datos con tres tablas. La primera tabla contiene las citas programadas. La segunda contiene las recetas médicas, diagnosticadas al paciente por su médico de cabecera o médico especialista. La tercera tabla contiene las referencias médicas del paciente.

A continuación se muestra la base de datos de MySQL que contiene los registros clínicos con los cuales se realizaron pruebas en el servicio IVR. Estas tablas pueden ser almacenadas en otro servidor o estar en el mismo junto al servidor Asterisk. Como podemos ver son tres tablas dentro de la base de datos. Éstas son: citas, farmacia y referencias.

```
$ mysql -u root -p
Enter password:
mysql> use hospital_rosales;
Database changed
mysql> show tables
+-----+
| Tables_in_hospital_rosales |
+-----+
| citas                       |
| farmacia                   |
| referencias                 |
+-----+
3 rows in set (0.00 sec)
```

*Configuración 9: Tablas en base de datos MySQL.*

La Configuración 9 muestra el contenido de la base de datos `hospital_rosales` la cual contiene las tres tablas antes mencionadas. En la Configuración 10, en la configuración 11 y en la configuración 12 se pueden observar las estructuras de estas tablas.

```
mysql> use hospital_rosales
Database changed
mysql> SELECT* FROM citas;
+-----+-----+-----+-----+
| Registro | Nombres                                     | Hora | Fecha           |
+-----+-----+-----+-----+
| 2579710 | Jesyca Pamela Sanchez Morales           | 2 p m | 4 Diciembre 2012 |
| 2579711 | Raul Fernando Alvarenga Aquino          | 4 p m | 4 Septiembre 2012 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

*Configuración 10: Tabla citas.*

La Configuración 10 muestra la tabla `citas`. Ésta contiene las columnas `registro`, `nombres` de los pacientes, `hora` y `fecha` que el paciente tiene una cita programada en el hospital. Esta información es proporcionada por el HNR



mediante la tarjeta “CONTROL DE CITAS” que es proporcionada a cada paciente cuando recibe un servicio.

```
mysql> SELECT* FROM referencias;
+-----+-----+-----+-----+
+-----+
| Registro | Nombres                | Servicio                | Diagnostico                | Referencia
+-----+-----+-----+-----+
| 2579710 | Yesica Pamela Sanchez Morales | Cirugia Plastica Mujeres | Granulona a cuerpo extranio | Marcos
Vindel
| 2579711 | Raul Fernando Alvarenga Aquino | Medicina General        | Dolor de Oidos            | Miguel
Sandoval
+-----+-----+-----+-----+
2 rows in set (0.02 sec)
mysql>
```

*Configuración 11: Tabla referencias.*

La Configuración 11 contiene la descripción de la tabla referencias, extraída de la “TARJETA DE REFERENCIAS” otorgada a los pacientes por el HNR. Esta tabla tiene las columnas registro, nombre, servicio, diagnóstico y referencia médica. Esta información es proporcionada a cada paciente con el objetivo de brindar el servicio de referencias médicas.

```
mysql> SELECT* FROM farmacia;
+-----+-----+-----+-----+-----+-----+
+-----+
| Registro | Nombres                | Sexo | Medicamento                | Cantidad | Disponibilidad |
Fecha_de_Entrega |
+-----+-----+-----+-----+-----+-----+
| 2579710 | Jesyca Pamela Sanchez Morales | Femenino | Clorfeniramina 4 mg | 20      | No disponible | 4
Diciembre 2012 |
| 2579711 | Raul Fernando Alvarenga Aquino | Maculino | Dextrometorfano      | 20      | No Disponible | 4
Septiembre 2012 |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

*Configuración 12: Tabla farmacia.*

La Configuración 12 muestra las columnas de la tabla farmacia. En sus primeras dos columnas, al igual que las tablas anteriores, cuenta con el número de registro y el nombre del paciente. Además, tiene las columnas sexo, medicamento, cantidad, disponibilidad y fecha de entrega del medicamento a los pacientes.

#### 4.5.2 Configuración de unixODBC.

La configuración del archivo `odbcinst.ini` nos permite tener una conexión con MySQL. En este archivo se crea un DSN (del inglés Data Source Name) dentro de él se registra el driver. Para lograrlo se tiene que editar el archivo `/etc/odbcinst.ini` agregando las siguientes líneas de código.

```
[MySQL]
Description      = Mysql Connector 3.51
Driver           = /usr/lib/odbc/libmyodbc.so
UsageCount       = 1
```

*Configuración 13: Archivo odbcinst.ini para registrar DSN.*

En la Configuración 13 la primer línea de código que se muestra es la cabecera

del DSN [MySQL]. Ésta permite acceder al driver cada vez que lo necesitemos. La segunda línea es una pequeña descripción. La tercera línea indica la ruta donde se encuentra el driver `libmyodbc.so` para tener conexiones hacia bases de datos del tipo MySQL.

Configuración del archivo `/etc/odbc.ini` en este archivo se registra el DSN y también se indican los parámetros de la conexión.

```
[hospital_rosales]
Driver      = MySQL
Description = Mi base Mysql
SERVER     = localhost
PORT      = 3306
USER      = root
Password  = proyecto12
Database  = hospital_rosales
OPTION    = 3
```

*Configuración 14: Archivo odbc.ini para conexión con MySQL.*

En la Configuración 14 la primera línea es la cabecera `hospital_rosales`. En la segunda línea se indica el DSN definido en la Configuración 13. La tercera línea es una pequeña descripción. En la cuarta línea, “server” indica la dirección del servidor para las bases de datos, en este caso es `localhost`. También, se indica el puerto, normalmente 3306 para MySQL. En la líneas restantes se proporciona las credenciales tales como usuario, contraseña y el nombre de la base de datos `hospital_rosales`.

Finalmente, se comprueba la conexión por medio del siguiente comando:

```
$ isql hospital_rosales
+-----+
| Connected!
|
| sql-statement
| help [tablename]
| quit
|-----|
SQL> quit
$
```

*Configuración 15: Prueba de conexión para bases de datos.*

En la Configuración 15 podemos ver que la conexión entre ODBC y MySQL. La conexión efectiva se puede verificar por el mensaje `connected!`. Para salir de la conexión basta con ejecutar el comando `quit`.

### 4.5.3 Configuración de los módulos `res_odbc` y `func_odbc`.

Para poder acceder a una base de datos desde Asterisk mediante `unixODBC`, necesitamos tener cargado el módulo `res_odbc.so` y `func_odbc.so`. Para verificar si estos módulos están cargados se ejecuta el siguiente comando.

```
CLI> module show like odbc
Module          Description          Use Count
res_odbc.so     ODBC resource        0
cdr_adaptive_odbc.so Adaptive ODBC CDR backend 0
func_odbc.so    ODBC lookups         0
res_config_odbc.so Realtime ODBC configuration 0
cdr_odbc.so     ODBC CDR Backend     0
cel_odbc.so     ODBC CEL backend     0
6 modules loaded
```

*Configuración 16: Verificación de módulos odbc.*

Como se puede apreciar en la Configuración 16 se tienen cargados 6 módulos

para ODBC. Estos permiten la conexión entre el plan de marcado de Asterisk y las bases de datos Mysql.

### Configuración de modulo `res_odbc`.

El módulo `res_odbc.so` nos permite registrar los DSN antes definidos en `unixODBC`, según Configuración 14. Para ello se debe modificar el archivo `/etc/asterisk/res_odbc.conf` con el siguiente segmento de código:

```
[hospital_rosales]
enable => yes
dsn => hospital_rosales
username => root
password => proyecto12
pre-connect => yes

[referencias]
enable => yes
dsn => hospital_rosales
username => root
password => proyecto12
pre-connect => yes

[citas]
enable => yes
dsn => hospital_rosales
username => root
password => proyecto12
pre-connect => yes

[farmacia]
enable => yes
dsn => hospital_rosales
username => root
password => proyecto12
pre-connect => yes
```

*Configuración 17: Registrando el DSN para crear las funciones de consultas hacia BD.*

En la Configuración 17 se tienen cuatro contextos `hospital_rosales`, `referencias`, `citas` y `farmacia`. Éstos permiten crear referencias para hacer consultas en las bases de datos. La primera línea de estos bloques es `enable`. Ésta se habilita con `yes`. Luego, en los cuatro bloques se indica el **DSN**, para este caso es `hospital_rosales` según Configuración 14. Luego, a cada bloque se le proporciona el nombre de usuario y la contraseña, siendo en todas igual a `root` y `proyecto12`. Por ultimo todos los bloques tienen la línea `pre-connect`.

### Configuración de módulo `func_odbc`.

Este módulo permite crear funciones de lectura para la tablas definidas en la Configuración 10,11 y 12. Para ello se debe modificar el archivo `/etc/asterisk/func_odbc.conf` y registrar los comandos que permiten hacer las consultas en MySQL.

```
[REFERENCIAS_NOMBRE]
dsn=referencias
readsql=SELECT Nombres FROM referencias WHERE Registro='${SQL_ESC(${ARG1})}'
```

*Configuración 18: Ejemplo de función en módulo de Asterisk para hacer consultas a BD.*

En la Configuración 18 se ha definido la función `REFERENCIAS_NOMBRE` que contiene dos líneas. La primera línea es el DSN `referencias` definido en la Configuración 17. La segunda línea es un comando MySQL, que realiza una consulta hacia la base de datos. La consulta se realiza según el registro almacenado en la

variable `${SQL_ESC(${ARG1})}`. Ese registro contiene una serie de siete dígitos generados mediante tonos DTMF desde una terminal telefónica. Estas funciones son utilizadas por el plan de marcado de Asterisk definidas en la Configuración 27.

Para hacer las diferentes consultas se ha definido una función por cada tipo de referencia en las bases de datos. Para más detalle consultar APÉNDICES.

#### 4.5.4 Configuración de Festival en Asterisk.

Para configurar el sintetizador de voz Festival simplemente hay que modificar/activar el siguiente archivo `/etc/asterisk/festival.conf`, como se muestra a continuación:

```
[general]
host=localhost
port=1314
festivalcommand=(tts_textasterisk "%s" 'file')(quit)\n
```

*Configuración 19: Configuración de Festival.*

Después de todas las configuraciones anteriores solo queda recargar todos los archivos de configuración de Asterisk por medio del siguiente comando en consola:

```
$/etc/init.d/asterisk reload
Reloading Asterisk PBX configuration files.
```

*Configuración 20: Recargando las configuraciones de Asterisk.*

Luego podemos verificar la configuración dentro del CLI de Asterisk.

```
# asterisk -r
CLI> odbc show
ODBC DSN Settings
-----
Name:  hospital_rosales
DSN:  hospital_rosales
Name:  referencias
DSN:  referencias
Name:  citas
DSN:  citas
Name:  farmacia
DSN:  farmacia
Last connection attempt: 1969-12-31 18:00:00
Pooled: No
Connected: Yes
```

*Configuración 21: Prueba de conexión Asterisk y MySQL.*

## 4.6 Configuración del plan de marcado de la red interna.

El sistema IVR propuesto para el HNR tiene dos contextos. Al primero se le ha llamado extensiones, contiene el plan de marcado de la red interna. El segundo contexto se ha llamado IVR, contiene el plan de marcado para las llamadas entrantes.

En la Configuración 8 se definieron las extensiones internas para la red telefónica del HNR y se hizo referencia al contexto extensiones. A continuación se modifica el archivo `/etc/asterisk/extension.conf`, agregando el siguiente

segmento de código:

```
[extensiones]
exten => 54994,1,Dial(SIP/54994,20,rt); contestar la llamada cuando se marque 54994.
exten => 54995,1,Dial(SIP/54995,20,rt); contestar la llamada cuando se marque 54995.
exten => 54996,1,Dial(SIP/54996,20,rt); contestar la llamada cuando se marque 54996.
exten => 54997,1,Dial(SIP/54997,20,rt); contestar la llamada cuando se marque 54997.
exten => 54998,1,Dial(SIP/54998,20,rt); contestar la llamada cuando se marque 54998.
exten => 50000,1,goto(IVR,s,1) ; acceder a la ivr desde las red interna.
exten => _2XXXXXXX,1,Dial(SIP/2110/${EXTEN}); en rutar la llamada por troncal 2110.
```

#### *Configuración 22: Contexto extensiones.*

En la Configuración 22 se muestra el contexto extensiones para la red del HNR. Tal como se ha configurado, éste permite tener comunicación dentro de la red interna y además tener acceso a la red publica PSTN por medio de la troncal sip 2110. Las líneas 2 a 6 configuran las extensiones de 5 dígitos 54994, 54995, 54996, 54997 y 54998. Todas las extensiones están configuradas de la misma manera. Asi por ejemplo, para la extensión 54994 el código `Dial(SIP/54994,20,rt)` significa generación de tonos de marcado durante un tiempo igual a 20 segundos.

En la línea siete se configura la extensión 50000 para acceder al servicio IVR desde la red interna el código `goto(IVR,s,1)` significa un salto al contexto IVR cuando desde la red interna se maque el numero 5000.

En la última línea de la Configuración 22 se muestra el código `Dial(SIP/2110/${EXTEN})` para generar llamada a números de ocho dígitos pertenecientes a la PSTN. El primer dígito de `_2XXXXXXX` indica que solo se podrán realizar llamadas a teléfonos de línea fija, pues el plan de marcación de la PSTN asigna como primer dígito el número 2 a los abonados fijos.

## **4.7 Configuración del plan de marcado para llamadas entrantes.**

Las llamadas entrantes son atendidas por una contestadora automática cuando el usuario marca desde su teléfono dado de alta en la PSTN, hacia el numero **25112110**. Esta contestadora presenta un saludo de bienvenida y los respectivos menús. Para que, el usuario puedan elegir las opciones de su preferencia. El menú principal y los siguientes menús son definidos en el archivo `/etc/asterisk/extension.conf` y para las aplicaciones que reproducen audio, se debe guardar los archivos con extensión **gsm** en `/var/lib/asterisk/sounds/en`.

### **4.7.1 Contexto IVR.**

En el siguiente contexto [IVR] se define, el menú principal antes descrito en la pagina 20 y presentado en la Ilustración 8.

En la Configuración 23 podemos ver que se tiene 6 prioridades del tipo **“s”** la primera `wait(4)`, indica una espera de 4 segundos, la segunda `set(digit=7)` establece un tiempo de 7 segundos entre la espera del primer dígito y los sucesivos. La tercera `Set(TIMEOUT(response)=10)` establece 10 segundos en espera para que el usuario presione una tecla. La cuarta utiliza la aplicación

playback(saludo\_de\_bienvenida) para reproducir un archivo de audio saludo de bienvenida. La quinta prioridad s utiliza la aplicación Background(menu) para reproducir el menú principal y al mismo tiempo capturar un tono DTMF. La ultima prioridad s utiliza la aplicación waitExten() para detectar los tonos DTMF marcados por el usuario.

```
[IVR]
exten => s,1,Wait(4) ; espera 4 segundos
exten => s,2,Set(TIMEOUT(digit)=7) ; tiempo entre deteccion de tonos
exten => s,3,Set(TIMEOUT(response)=10); tiempo de espera sin orpimir teclas
exten => s,4,Playback(saludo_de_bienvenida) ;reproduce el saludo de bienvenida
exten => s,5,Background(menu); presenta el menu principal
exten => s,6,WaitExten() ; detecta tonos dtmf
exten => 1,1,goto(IVR1,s,1) ; saltos hacia IVR[1,2,3,4]
exten => 2,1,goto(IVR2,s,1)
exten => 3,1,goto(IVR3,s,1)
exten => 4,1,goto(IVR4,s,1)
exten => i,1,Playback(opcion_invalida) ;validación de datos
exten => i,2,Playback(opcion_correcta)
exten => i,3,goto(IVR,s,5)
exten => t,1,goto(IVR,s,5)
exten => h,1,Hangup ; cuelga la llamada
```

### *Configuración 23: Menú principal del servicio IVR.*

Luego de las prioridades “s” se tiene cuatro líneas de código que utilizan la función goto() estas son utilizadas para hacer referencias a los contextos **IVR[1,2,3,4]** respectivamente. Seguidamente tenemos las prioridades “i” que son utilizadas para la validación de datos, es decir si el usuario oprime una tecla fuera de las validas entraran en función las prioridades “i”. La primera y segunda prioridad i utilizan las aplicaciones Playback(opcion\_invalida), Playback(opcion\_correcta) respectivamente, para reproducir un archivo de audio previamente grabado y así indicar que se marco una opción no valida, la tercera prioridad “i” hace un salto con la función goto(IVR,s,5) hacia la prioridad 5 “s” la cual se encarga de presentar el menú principal.

Finalmente si el usuario no presiona ninguna tecla dentro de 10 segundos la prioridad “t” goto(IVR,s,5) realizara un salto para presentar de nuevo el menú principal. Como se puede observar en el contexto “IVR” de la configuración 23 se hace referencia hacia a cuatro contextos llamados **IVR[1, 2 , 3 , 4]** los cuales presentan los cuatro menús del servicio. Estos servicios son: información de cirugía ambulatoria, indicaciones para pruebas de laboratorios, acceso a registro clínico y atención de operadoras internas, respectivamente.

#### **4.7.2 Contexto IVR1.**

El contexto IVR1 presenta el menú 1 y el contexto IVR2 presenta el menú 2 definidos en la pag. 21 estos dos contextos son similares, proporcionan información al usuario mediante grabaciones de audio, principalmente se hace uso de las aplicaciones background() y waitExten() de Asterisk para detectar los tonos DTMF y goto() para hacer saltos entre otros contextos. A continuación se presenta el contexto IVR1 en donde se aprecian las líneas de comando que permiten crear el menú 1 para el servicio IVR.

En la Configuración 24 se tiene tres prioridades “s” la primera wait(2) espera 2 segundos para pasar a la segunda prioridad para presentar el menu1 mediante la aplicación Background(menu1) y reproducir el archivo de audio

menu1. Al mismo tiempo espera que el usuario oprima una tecla. La tercera prioridad “s” utiliza waitExten() para detectar el evento DTMF. Luego de las prioridades “s” se tiene una serie de cuatro sentencias las cuales utilizan la aplicación goto() para hacer saltos hacia los contextos **IVR[a, b, c, d]** respectivamente. El contexto **IVR1** al igual que el contexto principal **IVR** tiene validación de datos mediante la prioridad “i”.

```
[IVR1]
exten => s,1,wait(2); espera 2 segundos
exten => s,2,BackGround(menu1); reproduce el archivo menu1
exten => s,3,waitExten() ; detecta tonos dtmf
exten => 1,1,goto(IVRa,s,1) ; saltos hacia contextos IVR[a,b,c,d]
exten => 2,1,goto(IVRb,s,1)
exten => 3,1,goto(IVRc,s,1)
exten => 4,1,goto(IVRd,s,1)
exten => t,1,goto(IVR1,s,1)
exten => i,1,Playback(opcion_invalida) ; validación de datos
exten => i,2,Playback(opcion_correcta)
exten => i,3,goto(IVR1,s,2)
exten => i,4,hangup ;cuelga la llamada
```

#### *Configuración 24: Contexto IVR1.*

En la Configuración 25 se tiene los cuatro contextos para proporcionar la información de cirugía ambulatoria utilizados por el contexto IVR1. Como se puede ver, todos son similares. Los contextos Cuentan con 5 prioridades s. La primera prioridad s wait(2) realiza una espera de dos segundos, la segunda y tercera reproducen los archivos de audio que contienen la información por medio de la aplicación Playback() y la cuarta reproduce un archivo de que indica la finalizacion de las indicaciones y una despedida. La quinta termina la llamada por medio de Hangup.

```
[IVRa]
exten => s,1,wait(2)
exten => s,2,Playback(cuidados_previos_a_la_cirugia_indicaciones)
exten => s,3,Playback(cuidados_previos_antes_de_la_cirugia)
exten => s,4,Playback(gracias_por_su_atencion)
exten => s,5,Hangup ;cuelga la llamada.
[IVRb]
exten => s,1,wait(2)
exten => s,2,Playback(normas_despues_de_la_operacion_indicaciones)
exten => s,3,Playback(Normas_despues_de_la_operacion)
exten => s,4,Playback(gracias_por_su_atencion)
exten => s,5,Hangup
[IVRc]
exten => s,1,wait(2) ; espera 2 segundos
exten => s,2,Playback(cuidados_en_casa_inidicaciones)
exten => s,3,Playback(cuidados_en_casa)
exten => s,4,Playback(gracias_por_su_atencion)
exten => s,5,Hangup
[IVRd]
exten => s,1,wait(2)
exten => s,2,Playback(signos_de_alarma_indicaciones)
exten => s,3,Playback(signos_de_alarma)
exten => s,4,Playback(gracias_por_su_atencion)
exten => s,5,Hangup
```

#### *Configuración 25: Contextos IVR[a,b,c,d] para información de cirugía ambulatoria.*

### **4.7.3 Contexto IVR2.**

El contexto IVR2 sirve para definir el menú 2 presentado en la Ilustración 10 permite proporcionar indicaciones para realizar pruebas de laboratorio de manera efectiva. Además es muy similar al IVR1, con la diferencia que este presenta 9 opciones y hace referencia a los contextos **IVR[e,f,g,h,i,j,k,l,m]** de la misma manera utilizando la aplicación goto().

En la Configuración 26 se muestra el código para definir el menú 2. Este contexto tiene 3 prioridades **s**, nueve saltos goto(), una prioridad **t** y un bloque para la validación de datos mediante las prioridades **i**.

```
[IVR2]
exten => s,1,Wait(2): espera 2 segundos
exten => s,2,BackGround(menu2) ; reproduce el archivo menu2
exten => s,3,WaitExten() ; obtiene los tonos DTMF
exten => 1,1,goto(IVRe,s,1);salto a IVRe
exten => 2,1,goto(IVRf,s,1);salto a IVRf
exten => 3,1,goto(IVRg,s,1);salto a IVRg
exten => 4,1,goto(IVRh,s,1);salto a IVRh
exten => 5,1,goto(IVRi,s,1);salto a IVRi
exten => 6,1,goto(IVRj,s,1);salto a IVRj
exten => 7,1,goto(IVRk,s,1);salto a IVRk
exten => 8,1,goto(IVRl,s,1);salto a IVRl
exten => 9,1,goto(IVRm,s,1);salto a IVRm
exten => t,1,goto(IVR2,s,1);salto a IVR2
exten => i,1,Playback(opcion_invalida); validacion de datos
exten => i,2,Playback(opcion_correcta)
exten => i,3,goto(IVR2,s,2) ;salto a IVR2
exten => i,4,hangup ; cuelga la llamada
```

*Configuración 26: Contexto IVR2.*

#### 4.7.4 Contexto IVR3.

En este contexto se define las líneas de código que nos permitirá acceder a las bases de datos de MySQL y así definir el menú 3 que se presento en la Ilustración 11. Este sirve para proporcionar la información de registro clínico al usuario por medio del sintetizador de voz Festival. En este contexto hay que mencionar tres aplicaciones de Asterisk muy importantes las cuales son **Read**, **Set** y **Festival** las cuales permiten proporcionar los datos que el paciente necesita. A continuación se muestra el contexto IVR3 para acceder a registro clínico.

```
[IVR3]
exten => s,1,Answer
exten => s,2,Wait(2)
exten => s,3,Festival("Por favor introduzca su numero de registro, de 7 digitos.")
exten => s,n,Read(REGISTRO,beep,7)
exten => s,n,NoOp(REGISTRO: ${REGISTRO})
exten => s,n,Set(NOMBRE=${ODBC_REFERENCIAS_NOMBRE(${REGISTRO})})
exten => s,n,Set(SERVICIO=${ODBC_REFERENCIAS_SERVICIO(${REGISTRO})})
exten => s,n,Set(DIAGNOSTICO=${ODBC_REFERENCIAS_DIAGNOSTICO(${REGISTRO})})
exten => s,n,Set(REFERENCIA=${ODBC_REFERENCIAS_MEDICO(${REGISTRO})})
exten => s,n,Set(FECHA=${ODBC_CITAS_FECHA(${REGISTRO})})
exten => s,n,Set(HORA=${ODBC_CITAS_HORA(${REGISTRO})})
exten => s,n,Set(MEDICAMENTO=${ODBC_FARMACIA_MEDICAMENTO(${REGISTRO})})
exten => s,n,Set(DISPONIBILIDAD=${ODBC_FARMACIA_DISPONIBILIDAD(${REGISTRO})})
exten => s,n,Set(ENTREGA=${ODBC_FARMACIA_FECHA_ENTREGA(${REGISTRO})})
exten => s,n,GotoIf("${NOMBRE}foo" = "foo"?notfound)
exten => s,n,Festival("El nombre del paciente es ${NOMBRE}.")
exten => s,n,Festival("El tipo de servicio del paciente es ${SERVICIO}.")
exten => s,n,Festival("Su diagnostico del es ${DIAGNOSTICO}.")
exten => s,n,Festival("Su medico de referencia es ${REFERENCIA}.")
exten => s,n,Festival("Usted tiene una cita el dia ${FECHA} a la hora ${HORA} .")
exten => s,n,Festival("su Receta medica es ${MEDICAMENTO}. y esta, ${DISPONIBILIDAD} a la fecha en Farmacia ")
exten => s,n,Festival("Gracias por su atencion.")
exten => s,n,NoOp(Employee: ${NOMBRE})
exten => s,n,goto(IVR,s,5)
exten => s,n(notfound),Festival("El registro del paciente no esta registrado.")
exten => s,n,goto(IVR3,s,3)
```

*Configuración 27: Contexto IVR3.*

En la Configuración 27 podemos apreciar el contexto que permite acceder a las tablas de MySQL. Para ilustrar el proceso se explicaran las sentencias mas importantes, utilizadas para hacer las consultas. primero por medio del



sintetizador de voz se pide al usuario marcar su numero de registro de siete dígitos mediante el comando `Festival("Por favor introduzca su numero de registro, de 7 digitos.")` luego con la ayuda de la aplicación `Read(REGISTRO,beep,7)` se guarda ese numero en la variable `REGISTROS`, además se reproducirá un archivo de audio llamado `beep`.

Ya que se tiene almacenado el numero de registro en la variable `REGISTROS` se hace la consulta por medio de la sentencia `Set(NOMBRE=${ODBC_REFERENCIAS_NOMBRE(${REGISTRO})})` esta sentencia significa que el sistema llamara a la función "REFERENCIAS\_NOMBRE" antes definida en Configuración 18 para que Asterisk haga una lectura en la base de datos MySQL con el valor almacenado en "REGISTRO" y todo eso se almacena en la variable llamada "NOMBRE".

Luego se pasa la variable llamada `NOMBRE` a `Festival` por medio de la sentencia `Festival("El nombre del paciente es ${NOMBRE}.")` finalmente la cadena de texto almacenada en "NOMBRE" es convertida a voz sintetica en el comando de `Festival` y asi el sintetizador proporciona la información.

#### 4.7.5 Contexto IVR4.

Este contexto permite definir el menú 4 presentado en la Ilustración 12 el cual sirve para trasladar la llamada a operador, desviando la llamada hacia una extensión interna, Esta representa la atención del departamento de cirugía ambulatoria (ext 54994) y laboratorio clínico (ext 54995).

```
[IVR4]
exten => s,1,wait(2); espera 2 segundos
exten => s,2,BackGround(menu4) ;reproduce el archivo de audio menu4
exten => s,3,waitExten() ;capta los tonos DTMF
exten => 1,1,goto(IVR11,s,1) ; salto a IVR11
exten => 2,1,goto(IVR12,s,1) ; salto a IVR12
exten => 3,1,goto(IVR13,s,1) ; salto a IVR13
exten => 4,1,goto(IVR14,s,1) ; salto a IVR14
exten => i,1,Playback(opcion_invalida) ; reproduce archivo de audio
exten => i,2,Playback(opcion_correcta) ; reproduce archivo de audio
exten => i,3,goto(IVR4,s,2) ;salto a IVR4
exten => i,4,hangup ; cuelga la llamada
```

*Configuración 28: Contexto IVR4.*

En la Configuración 28 se puede observar que el contexto tiene tres prioridades "s", la primera `wait(2)` es una espera de 2 segundos, la segunda utiliza la aplicación `BackGround(menu4)` para reproducir el archivo de audio `menu4` y la tercera `waitexten()` para detectar los tonos DTMF.

Luego se tiene cuatro sentencias `goto()` que hacen referencia a los contextos **IVR1[1,2,3,4]** dentro de los cuales se hace uso de la aplicación "dial" para generar tonos de llamada a las respectivas extensiones mediante `Dial(SIP/5499X,20,rt)`. También cuenta con sentencias para validar los datos por medio de las prioridades "i", realizando un `goto()` hacia la segunda `s,2,BackGround(menu4)`.

## **CAPITULO V. IMPLEMENTACIÓN DEL SERVICIO IVR PARA UN ENTORNO RURAL.**

Existen organizaciones internacionales tales como : OPS (organización panamericana de la salud) y la OMS (organización mundial para la salud) que insta a los países latinoamericanos a impulsar establecimientos para crear redes integrales de servicio a la salud con un enfoque de salud comunitaria. Dada esta iniciativa el MINSAL introduce desde julio del 2010 los equipos comunitarios de salud ECOS para asegurar que el sistema de salud sea mas accesible en aquellos lugares donde la atención es muy necesaria. Los ECOS integran la reforma de salud promovida por el MINSAL esta se centra en el reforzamiento del sistema de atención para descongestionar el flujo de pacientes en los hospitales de tercer nivel (Rosales, Maternidad y Bloom)

La Universidad de El Salvador a través de su departamento de Telecomunicaciones ha desarrollado diversas investigaciones sobre como implantar redes inalámbricas de bajo costo en lugares de acceso remoto. Realizando así estudios que han llevado a obtener una infraestructura de red dinámica basada en las REDES MESH (de su español redes maya), estas redes se basan en protocolos para enrutar paquetes de datos de una manera inteligente con un protocolo llamado B.A.T.M.A.N. (de sus siglas en ingles better approach to mobile ad-hoc networking). La tecnología utilizada para implementar estas redes son los novedosos routers inalámbricos MESH POTATO que son potenciados por la el proyecto "Village Telco". Un proyecto que se encarga de fomentar redes MESH para ser usada en sistemas telefónicos y de datos de bajo costo basado en los estándares WIFI.



*Ilustración 15: Redes de Bajo costo basadas en estándar WIFI.*

El servicio IVR para el HNR puede ser implementado en una red inalámbrica WIFI, con el objetivo de brindar la información e indicaciones a pacientes en donde no hay acceso a las redes de telefonía tradicionales, como por ejemplo en zonas rurales y así poder integrar los equipos comunitarios de salud implantados por el MINSAL. Estas redes pueden ser utilizadas en un futuro para instruir a la población en casos de desastres o epidemia. Siendo la población la usuaria de este sistema.

### 5.1 Configuración de Red inalámbrica para integrar los ECOS.

A continuación se muestra el diseño de una red comunitaria de bajo costo, implementada para dar servicio donde se encuentran los ECOS (Equipos Comunitarios de Salud). Esta red se construye gracias a los router MP de la comunidad Village Telco.

En la Ilustración 16 se puede ver que la red WIFI se integra fácilmente al servidor VoIP EIE, por medio de el router buellet ubiquiti. Este router permite el encaminamiento de los paquetes de datos en la red y también crea una conexión entre la red cableada y la red inalámbrica.

Para integrar la red WIFI MESH al servicio VoIP se debe registrar una extensión en cada uno de ellos las cuales se definieron en Configuración 8.

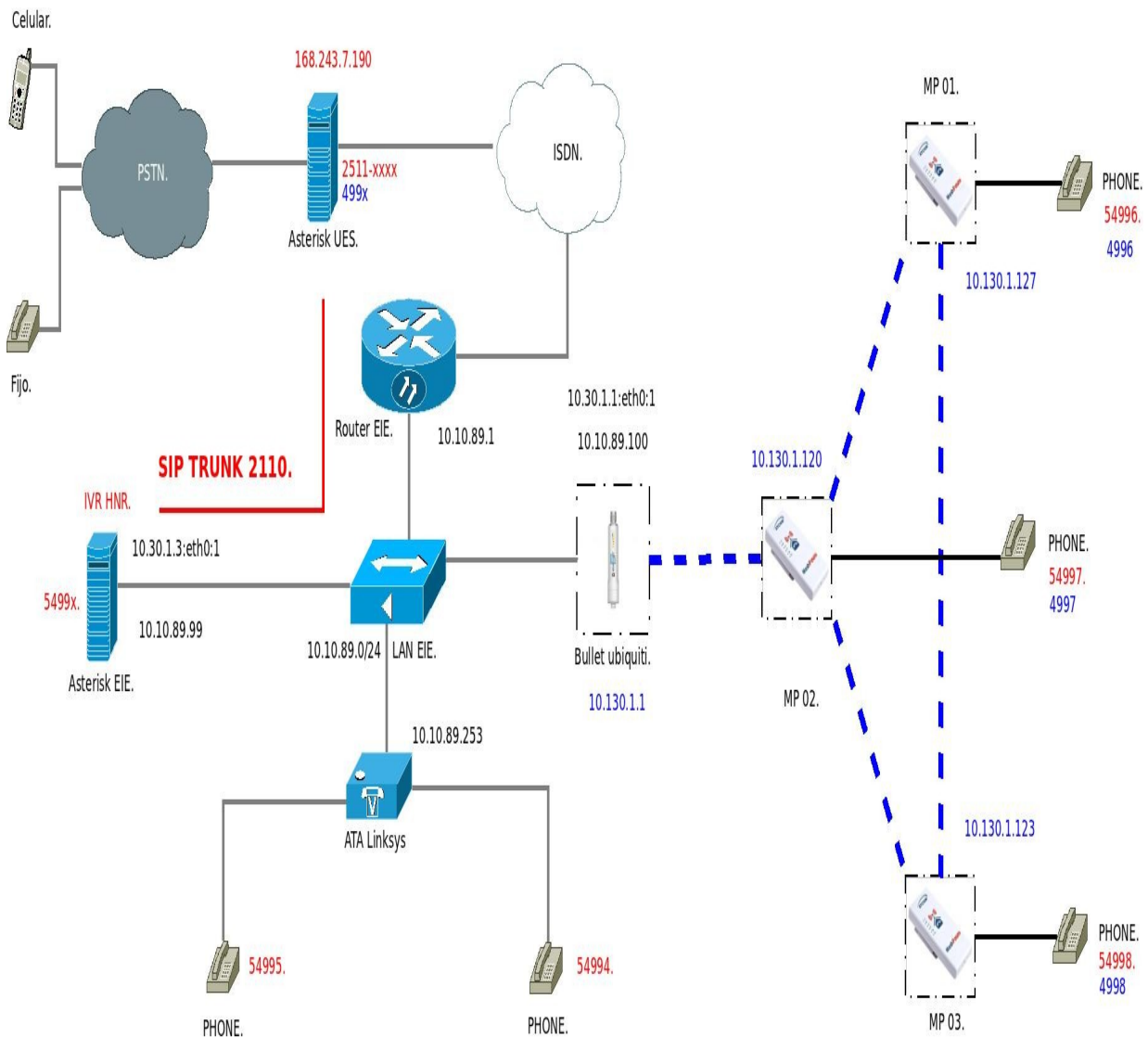


Ilustración 16: Diseño de la Red para dar servicio IVR en los ECOS.

Los routers MP son registrados al servidor VoIP EIE como clientes es decir se

configura una extensión del tipo 5499x dentro de ellos con el objetivo de darles acceso a la red interna y también a la PSTN.



*Ilustración 17: Puertos del Router MP*

En la Ilustración 17 se observa los dos tipos de puertos que poseen los MP. El puerto Rj11 sirve para conectar un teléfono analógico y el Rj45 para conectores Ethernet.

## **5.2 Configuración de las Troncales SIP en routers MP.**

Para integrar la red WIFI MESH al servicio IVR se debe registrar cada uno de los router MP como clientes del servidor Asterisk UES, por medio de una Troncal SIP. Esta configuración es muy parecida a la que se hace con la troncal 2110 de la pag. 25 En la sección 4.3 referente a la troncal en servidor VoIP EIE.

Para registrar los MP al servidor VoIP de la UES se debe modificar el archivo /etc/asterisk/sip.conf con los siguientes bloques.

### **5.2.1 troncal SIP MP 01.**

en /etc/asterisk/sip.conf después de la cabecera [general]

```
register = 54996@54996/54996  
register = 4996@4996/4996
```

*Configuración 29: Registro de la troncal MP 01.*

## 5.2.2 troncal SIP MP 02.

en /etc/asterisk/sip.conf después de la cabecera [general]

```
register = 54997@54997/54997
register = 4997@4997/4997
```

*Configuración 30: Registro de la troncal MP 02.*

## 5.2.3 troncal SIP MP 03.

en /etc/asterisk/sip.conf después de la cabecera [general]

```
register = 54998@54996/54998
register = 4998@4996/4998
```

*Configuración 31: Registro de la troncal MP 03.*

En las configuraciones 29, 30 y 31 se registran las troncales SIP de cada MP. La primera línea `register = 5499x@5499x/5499x` registra la extensión interna con el servidor VoIP EIE y la segunda `register = 499x@499x/499x` registra la extensión con el servidor VoIP UES. Como se puede ver existe redundancia en el sistema, es decir este es capas de registrarse a varios servidores VoIP en simultaneo. Esta característica ayuda a los router a buscar una ruta alternativa para encaminar sus llamadas.

## 5.3 Configuración de las extensiones SIP en routers MP.

Como antes se menciona el sistema posee redundancia en dos servidores VoIP y también dentro de su propia red es decir se pueden hacer llamadas internas marcando la ultima serie de su dirección IP inalámbrica. Por lo tanto es necesario crear dos tipos de extensiones dentro del archivo `sip.conf` para cada router MP.

La configuración de las extensiones se logra modificando el archivo `/etc/asterisk/sip.conf` agregando los contextos respectivos en cada uno de los router MP, Cabe mencionar que cada MP cuenta con un plan de marcado definido en el contexto `default` más adelante en un apartado se explica el funcionamiento de estos.

### 5.3.1 Extensiones en MP 01.

al final del archivo `/etc/asterisk/sip.conf`

```
[4996]
host=168.243.7.190
secret=xxxxxx
username=4996
fromuser=4996
insecure=port,invite
type=friend
disallow=all
allow=gsm,ulaw,alaw
dtmfmod=rfc2833
qualify=yes
canreinvite=no
nat=yes
context=default

[54996]
host=10.30.1.3
secret=xxxxxx
username=54996
fromuser=54996
insecure=port,invite
type=friend
disallow=all
allow=gsm,ulaw,alaw
dtmfmod=rfc2833
qualify=yes
canreinvite=no
nat=yes
context=default
```

*Configuración 32: Extensiones SIP MP 01.*

### 5.3.2 Extensiones en MP 02.

```
[4997]
host=168.243.7.190
secret=xxxxxx
username=4997
fromuser=4997
insecure=port,invite
type=friend
disallow=all
allow=gsm,ulaw,alaw
dtmfmod=rfc2833
qualify=yes
canreinvite=no
nat=yes
context=default

[54997]
host=10.30.1.3
secret=xxxxxx
username=54997
fromuser=54997
insecure=port,invite
type=friend
disallow=all
allow=gsm,ulaw,alaw
dtmfmod=rfc2833
qualify=yes
canreinvite=no
nat=yes
context=default
```

Configuración 33: Extensiones SIP MP 02.

### 5.3.3 Extensiones en MP 03.

```
[4998]
host=168.243.7.190
secret=xxxxxx
username=4998
fromuser=4998
insecure=port,invite
type=friend
disallow=all
allow=gsm,ulaw,alaw
dtmfmod=rfc2833
qualify=yes
canreinvite=no
nat=yes
context=default

[54998]
host=10.30.1.3
secret=xxxxxx
username=54998
fromuser=54998
insecure=port,invite
type=friend
disallow=all
allow=gsm,ulaw,alaw
dtmfmod=rfc2833
qualify=yes
canreinvite=no
nat=yes
context=default
```

Configuración 34: Extensiones SIP MP 03.

## 5.3 Configuración del plan de marcado en routers MP.

El plan de marcado define como serán procesadas las llamadas telefónicas dentro de cada MP, por lo tanto se modifican los contexto default en los archivos extension.conf de cada MP de la siguiente manera.

Plan de marcado en MP 01. En archivo /etc/asterisk/extension.conf dentro del contexto default.

```
[default]
exten => _XXXXX,1,Dial(SIP/54996/${EXTEN}) ;para llamadas salientes.
exten => 54996,1,Dial(MP/1,80) ;para llamadas entrante.
exten => _2XXXXXXX,1,Dial(SIP/54996/${EXTEN}) ;para llamadas a la pstn.
exten => _XXXX,1,Dial(SIP/4996/${EXTEN}) ;para llamadas salientes.
exten => 4996,1,Dial(MP/1,80) ;para llamadas entrante.
```

Configuración 35: Plan de marcado MP 01.

Plan de marcado en MP 02. En archivo /etc/asterisk/extension.conf dentro del contexto default.

```
[default]
exten => _XXXXX,1,Dial(SIP/54997/${EXTEN}) ;para llamadas salientes.
exten => 54997,1,Dial(MP/1,80) ;para llamadas entrante.
exten => _2XXXXXXX,1,Dial(SIP/54997/${EXTEN}) ;para llamadas a la pstn.
exten => _XXXX,1,Dial(SIP/4997/${EXTEN}) ;para llamadas salientes.
exten => 4997,1,Dial(MP/1,80) ;para llamadas entrante.
```

Configuración 36: Plan de marcado MP 02.

Plan de marcado en MP 03. En archivo /etc/asterisk/extension.conf dentro del

contexto default.

```
[default]
exten => _XXXXX,1,Dial(SIP/54998/${EXTEN}) ;para llamadas salientes.
exten => 54998,1,Dial(MP/1,80) ;para llamadas entrante.
exten => _2XXXXXXX,1,Dial(SIP/54998/${EXTEN}) ;para llamadas a la pstn.
exten => _XXXX,1,Dial(SIP/4998/${EXTEN}) ;para llamadas salientes.
exten => 4998,1,Dial(MP/1,80) ;para llamadas entrante.
```

### Configuración 37: Plan de mercado MP 03.

como se puede ver en los contextos default, de las configuraciones 35, 36, 37 se definen dos líneas para las llamadas entre extensiones del tipo 5499X y otras dos para extensiones del tipo 499X.

la línea `exten => _XXXXX,1,Dial(SIP/5499x/${EXTEN})` atiende las llamadas del tipo 5499X e indica que si el usuario marca un numero de 5 dígitos este sera enviado por el canal SIP 5499x, la segunda línea `5499x,1,Dial(MP/1,80)` indica que si una terminal marca el numero 5499x el router correspondiente atenderá la llamada generando tonos de marcado. También se define como sera tratado un numero marcado de 8 dígitos, en la línea `_2XXXXXXX,1,Dial(SIP/5499x/${EXTEN})`. De igual manera se definen las líneas para procesar las llamas de las extensiones del tipo 499x mediante la línea `exten => _XXXX,1,Dial(SIP/499x/${EXTEN})` para ser enviadas por el canal SIP y `exten => 4998,1,Dial(MP/1,80)` para ser atendidas.

Un ejemplo de como se procesan las llamadas telefónicas para números de ocho dígitos se muestra en la ilustración 17.

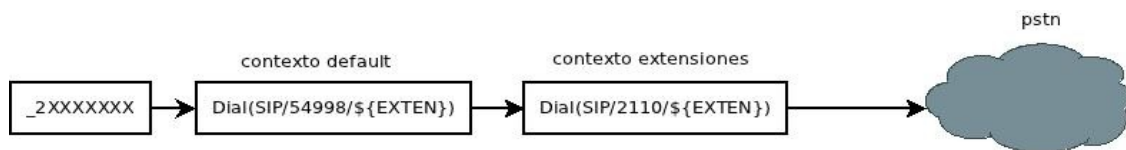


Ilustración 18: Plan de marcado para llamadas salientes

Cuando un numero de ocho dígitos es marcado desde la terminal conectada en un MP. La llamada es procesada por el contexto default y enviada por el canal SIP 54998 hacia el servidor VoIP EIE. Luego la misma llamada es procesada por el contexto extensiones y enviada por el canal SIP 2110 hacia la PSTN mediante la troncal SIP con el servidor VoIP UES.

## **CAPITULO VI. CONCLUSIONES Y LINEAS FUTURAS.**

En este trabajo de graduación se implemento un servicio de voz iterativo IVR para facilitar la información a los pacientes del HNR utilizando Asterisk como servidor VoIP.

Luego de configurar e implementar este servicio se llego a las siguientes conclusiones:

### **6.1 Conclusiones.**

- **Protocolo:** el protocolo SIP, es el mas utilizado en la VoIP pero presenta algunos inconvenientes tales como el consumo de banda ancha y los problemas de NAT's debido a que la voz y la señalización viajan por circuitos diferentes mas sin embargo es el protocolo mas utilizado en VoIP. En este proyecto de graduación se crearon Troncales y extensiones SIP para implementar el servicio IVR. Sin embargo también se podría haber utilizado el protocolo IAX2 que consume menor ancho de banda y es mucho más robusto aunque aun no esta muy utilizado en redes VoIP. Debido a eso la integración de la IVR podría ser de menor escala.
- **Codec:** dada la importancia del ancho de banda sin dejar de lado la calidad del procesamiento de voz, es necesario tener en cuenta que tipo de codec puede ser mas fiable para el sistema. En este servicio IVR se deja configurado los tres principales codecs utilizados por Asterisk (g.729, g.711 y GSM) para que pueda ser compatible con cualquier otra PBX. Algunas centrales PBX utilizan el codec GSM que sobresale por su simpleza y calidad. Sin embargo es recomendable utilizar codec mas frecuentemente en las redes VoIP.
- **Sistemas Operativos:** Para la implementación del servicio se utilizo Ubuntu/linux dada la buena fiabilidad y la administración de recursos que provee. Además de ser un sistema de código abierto posee un gran soporte en la Internet. Esto es de gran ventaja, debido a que los responsables de dar mantenimiento al servicio IVR cuentan con mucha información gratuita para poder solventar los problemas que puedan surgir en la puesta en marcha y operación del servicio.
- **Servicio y operación:** El servicio puede llegar a ser cambiado y modificado durante la operación. Esto debido a que no hay una receta efectiva para garantizar un buen diseño, mas sin embargo se puede obtener una buena calidad en el servicio creando y modificando los componentes.



- Interfaces de voz : la utilización de sintetizadores para la voz con software libre como Festival facilita al sistema interactuar con los usuarios, pero cabe mencionar que la calidad de voz debe ser mejorada. Por lo tanto si se desea obtener un mejoramiento en las voces sintéticas se debe seguir realizando estudios de como incorporar otras herramientas para lograr trasladar la información almacenada en gestores para bases de datos.

## **6.2 Lineas futuras.**

- Luego de haber realizado este trabajo de graduación. El laboratorio de telecomunicaciones de la EIE estará listo para seguir desarrollando nuevos servicios relacionados a la redes VoIP tales como: Servicio de Call-Center debido a que estos también se pueden implementar en una red telefónica similar a la de este proyecto. Sin embargo esto conlleva a realizar investigaciones de los funcionamientos de esos servicios.
- Se debe seguir potenciando el uso de protocolos de código abierto tales como IAX debido a que su funcionalidades provee características muy peculiares y eficientes en el uso de ancho de banda. Por lo tanto se deben realizar investigaciones acerca de su implementación en centralitas Asterisk.
- Además estos tipos de servicios pueden ser de gran utilidad para solventar las necesidades de comunicación a la población en zonas rurales por eso es necesario seguir potenciando las redes inalámbrica WIFI con el objetivo de crear nuevos y diferentes tipos de servicios.

## APÉNDICES.

### APÉNDICE A. Software complementario utilizado en el servicio.

#### Wireshark.

Wireshark es un analizador de protocolos open-source diseñado por Gerald Combs y que actualmente está disponible para plataformas Windows y Unix.

su principal objetivo es el análisis de tráfico además de ser una excelente aplicación didáctica para el estudio de las comunicaciones y para la resolución de problemas de red. Wireshark implementa una amplia gama de filtros que facilitan la definición de criterios de búsqueda para los más de 1100 protocolos soportados actualmente (versión 1.4.3), y todo ello por medio de una interfaz sencilla e intuitiva que permite desglosar por capas cada uno de los paquetes capturados. Gracias a que Wireshark “entiende” la estructura de los protocolos, podemos visualizar los campos de cada una de las cabeceras y capas que componen los paquetes monitorizados, proporcionando un gran abanico de posibilidades al administrador de redes a la hora de abordar ciertas tareas en el análisis de tráfico. para instalar Wireshark en Ubuntu 11.04 se debe ejecutar el siguiente comando:

```
sudo apt-get install wireshark
```

*Instalación 17: Wireshark en Ubuntu.*

#### Configuración de un cliente IAX con ZOIPER.

Zoiper es un software telefónico, el cual permite configurar un cliente SIP y otro IAX2 para poder hacer llamados desde la PC o Notebook hacia otros usuarios de la red. A continuación se muestra como registrar una extensión IAX en Asterisk con este Softphone.

En la sección [general] del archivo iax.conf activar las siguientes líneas.

```
[general]
bindport=4569 ;el puerto UDP usado por este protocolo
bindaddr=10.10.89.99 ;el ip que usará para "escuchar" los pedidos de conexiones
delayreject=yes ; Mejora la seguridad contra "brute force password attacks" retrasando el envío de los rechazos de autenticación
srvlookup=yes ; permite una gestión optimizada de los DNS
accountcode=lss0101 ;genera una cuenta general con la cual grabar las llamadas.
language=en ; lenguaje predefinido de la voces (ingles)

disallow=all ; desactivamos todos los codecs (audio y video)
allow=ulaw ; definimos unos cuantos codecs audio predefinido's para todas las extensiones.
allow=alaw
allow=gsm
```

*Configuración 38: Configuración del Servidor para utilizar IAX.*

Luego para configurar una extensión IAX se debe agregar al final del archivo /etc/asterisk/iax.conf un segmento parecido al siguiente:

```
[marko]
type=friend ; tipo friend es peer y user a la vez
host=dynamic ; si el cliente no se conecta siempre desde un IP determinado hay que indicar dynamic
secret=password ; contraseña
context=default ; contexto asociado a este usuario en extensions.conf
```

### Configuración 39: Extensión con protocolo IAX.

Luego se debe descargar Zoiper de <http://www.zoiper.com/softphone/> una vez se descarga en un archivo en un tar.gz se debe descomprimir.

```
tar -xzf Archivo.tar.gz
```

### Instalación 18: Instalación de Zoiper.

dentro de la carpeta solo debe de haber un archivo llamado zoiper ese archivo se debe pasar a un directorio mas apropiado, este puede ser /usr/sbin con el siguiente comando:

```
$ mv zoiper /usr/sbin
$ zoiper ;para lanzar el softphone
```

### Instalación 19: Instalación de Zoiper para ejecutar desde consola.

Una vez instalado lo corremos y vamos en la ventana de opciones. Escogemos Cuenta nueva de iax y ponemos un nombre de la cuenta.

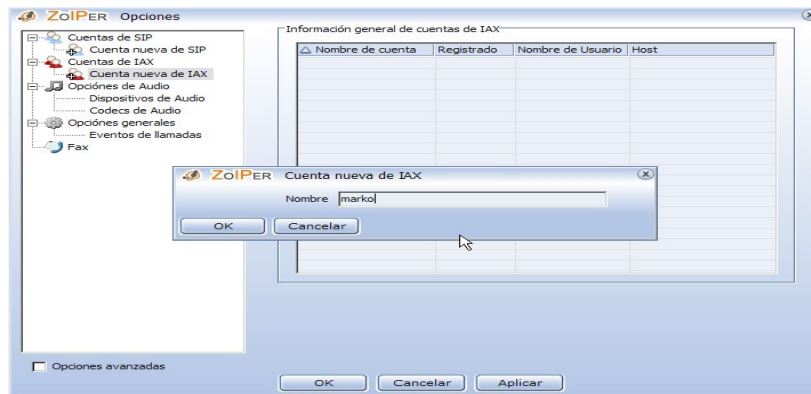


Ilustración 19: Zoiper 1.

En la siguiente pantalla insertamos los datos de la cuenta como definidos en iax.conf. en la barra de la cuenta aparecerá: marko (Registrado)

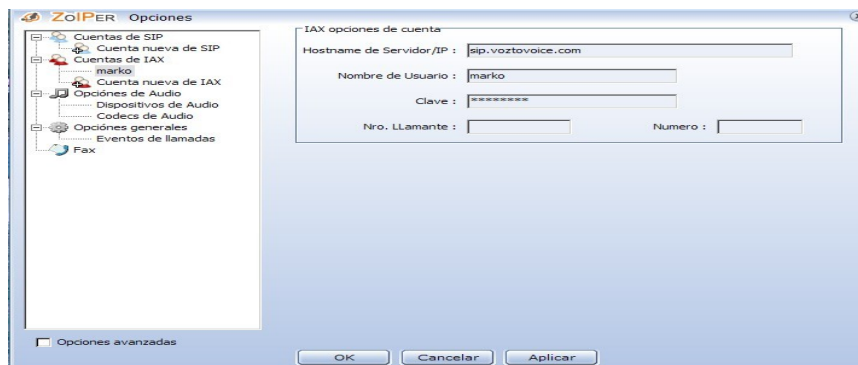


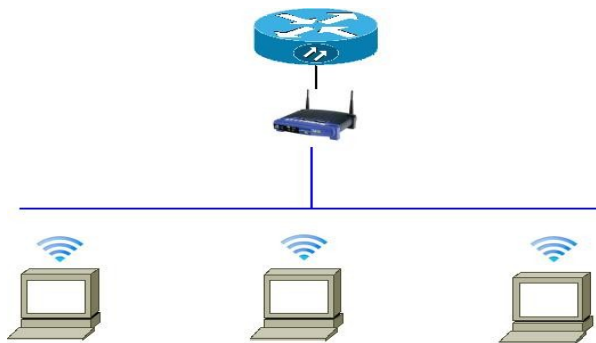
Ilustración 20: Zoiper 2.

## **APÈNDICE B. Tipos de Infraestructura en redes inalámbricas.**

Las redes de computadores inalámbricas (Wireless) pueden clasificarse en dos grandes grupos.

### **Redes con infraestructura.**

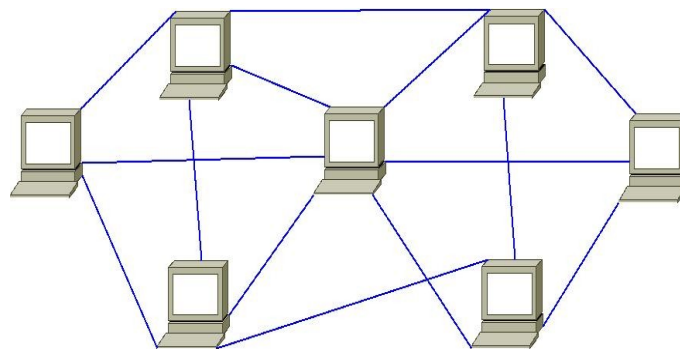
Constan de un número fijo de enlaces cableados entre sí como se muestra en la Ilustración 21. Cada host móvil debe comunicar con uno de estos enlaces dentro de su radio de acción. El nodo puede moverse libremente pero si sale fuera del rango de su enlace, debe conectar con otro para asegurar que la información llegue a su destino. Un ejemplo de este tipo de redes es la red de telefonía móvil formada por numerosas estaciones y antenas dispersas por todas las ciudades.



*Ilustración 21: Infraestructura WIFI.*

### **Redes sin infraestructura (Ad-Hoc).**

Formadas por hosts móviles y que pueden estar conectados entre sí arbitrariamente y de manera dinámica. Es decir, no hay ningún elemento fijo y la topología de la red puede adoptar múltiples formas siendo igual de funcional como se muestra en la Ilustración 22. En este tipo de redes, todos los nodos funcionan como encaminadores (routers) y se ven involucrados tanto en el descubrimiento como en el mantenimiento de rutas.



*Ilustración 22: Redes Ad-Hoc.*

Algunos ejemplos de uso de las redes Ad-Hoc son: Operaciones de emergencia de búsqueda y rescate, convenciones y análisis de datos en terrenos catastróficos.

## APÈNDICE C. Redes WIFI MESH.

### Funcionamiento de la Red.

En un principio la redes Mesh fueron utilizadas en aplicaciones militares. Para comunicarse con aquellas estaciones que se encontraban lejos de la zona de cobertura. El principio de funcionamiento de estas aplicaciones radicaba en un método para enviar paquetes entre estaciones bases cercanas con el objetivo de crear rutas hacia aquellas otras estaciones lejanas a al base principal. Estas a su vez debían irse pasando la información en forma de saltos hasta llegar al objetivo.

Una definición sencilla de las redes Mesh podría ser: son aquellas redes en las que se mezclan las dos topologías de las redes inalámbricas( infraestructura y ad-hoc ). Básicamente son redes con topología de infraestructura, pero que permiten unirse a la red a dispositivos que a pesar de estar fuera del rango de cobertura de los puntos de acceso están dentro del rango de cobertura de algún nodo móvil que directamente o indirectamente está dentro del rango de cobertura del punto de acceso (topología sin infraestructura o ad-hoc).

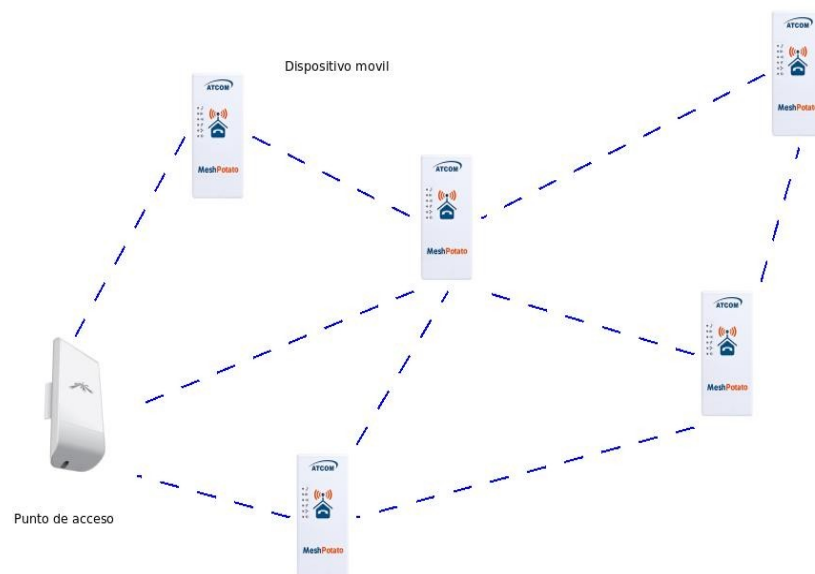


Ilustración 23: Topología de Red Mesh.

En la Ilustración 23 muestra que estas redes también permiten que los nodos móviles se comuniquen independientemente del punto de acceso entre sí. Actúan como repetidores para transmitir la información de los nodos alcanzables hasta los extremos que quieren comunicarse y que están demasiado lejos para alcanzarse, dando como resultado una red que permite abarcar grandes distancias. Además son un tipo de red muy confiable, ya que cada nodo esta conectado a varios nodos. Si uno de ellos falla o abandona la red, sus vecinos simplemente buscarán otra ruta. Para ampliar la capacidad y la confiabilidad de la red basta con añadir más nodos.

## **Protocolo de Enrutamiento B.A.T.M.A.N.**

De sus siglas en ingles Better Approach To Mobile Ad-hoc Networking B.A.T.M.A.N. es un protocolo de ruteo pro activo para Redes Mesh (Ad-hoc) Inalámbricas . Se utiliza para mantener pro activamente la información sobre la existencia de todos los nodos en la red y también para saber cuales son accesibles a través de los enlaces de comunicación de uno o múltiples saltos. La estrategia de este protocolo es determinar para cada destino en la malla un vecino para realizar saltos sucesivos y así poder realizar un ruteo de múltiple saltos basado en IP[13].

Algunas características son :

- Soporte de múltiples interfaces es decir puede ser titulizado en más de una tarjeta inalámbrica.
- Soporte de interfaces alias, lo que permite correr en paralelo con otros protocolos.
- Interfaz IPC (Inter-Process Communication), sirve para conectarse al demonio batmand.

Permite realizar consultas sobre:

- Información de debug .
- Que nodos vecinos posee la red .
- Que nodos GW existen al alcance.

## ***APÈNDICE D. Configuración del Adaptador PAP2T.***

El adaptador de teléfono Linksys PAP2T permite conectar un teléfono analógico a una red de área local LAN para poder usarlo con un servidor VoIP. A continuación se muestran los pasos a seguir para su correcta configuración.

PASO 1.

determinar la dirección IP que recibió. Para conseguir la dirección IP, levante el teléfono asignado al conector de la línea 1 y Marque: \*\*\*\* (**4 asteriscos**) Después marque: **110 #** y recibirá la dirección IP de su dispositivo, por ejemplo: 192.168.0.100.

PASO 2.

Use un navegador en su red e ingrese la dirección: **http://<IP ADDRESS>/** donde <IP ADDRESS> se reemplaza con la dirección que recibió en el PASO 1.

PASO 3.

Haga clic en el botón de "Admin Login" en la esquina superior derecha para

iniciar una sesión y después haga clic en la pestaña "Line 1" para seleccionar la línea 1.



*Ilustración 24: ATA Linksys 1.*

PASO 4.

Sólo necesitará modificar unos cuantos parámetros establecidos de fábrica. Estos son:

**Proxy:** dirección IP del servidor VoIP. **Display Name:** Ingrese su nombre completo. **User ID:** Ingrese el número de teléfono de la extensión. **Password:** Ingrese la contraseña **Register Expires:** 3600.

PASO 6.

Haga clic en el boto Save Setting.



*Ilustración 25: ATA Linksys 2.*

PASO 7.

Hacer llamadas.

## BIBLIOGRAFÍA.

- [1] Alberto Escudero Pascual y Louise Berthilson, VoIP para el Desarrollo Diciembre 2006 .  
[www.it46.se](http://www.it46.se)
- [2] Travis Russell, Session Initiation Protocol SIP, 2008.
- [3] Wiley, Inter Asterisk Exchange IAX, 2009.
- [4] 3CX, Protocolos VoIP, 2009.  
<http://www.voipforo.com/protocolosvoip.php>
- [5] flavio e.gonçalves, como construir y configurar un pbx con software libre, 2005.
- [6] 3CX, Codecs, 2009.  
<http://www.voipforo.com/codec/codecs.php>
- [7] David Gómez, Cómo instalar paso a paso Asterisk 1.8, 23 mayo, 2011.  
<http://emslinux.com/>
- [8] Ubuntu ES, MySQL, 2007.  
<http://www.guia-ubuntu.org/index.php?title=MySQL>
- [9] Greg, UnixODBC example setup, 2011.
- [10] voztovoice, Instalar y configurar Festival en Asterisk 1.6.X, Septiembre 2009.  
<http://www.voztovoice.org/?q=node/97>
- [11] Village telco, village telco, 2012  
<http://villagetelco.org/>
- [12] VozToVoice, Asterisk Func\_odbc.conf - ODBC y Base de datos, Septiembre 2009.  
<http://www.voztovoice.org/?q=node/187>
- [13] B.A.T.M.A.N. protocol concept  
<http://www.open-mesh.org/wiki/open-mesh/BATMANConcept>