





**ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO**

**FACULTAD DE INFORMATICA Y ELECTRONICA**

**ESCUELA DE INGENIERIA EN SÍSTEMAS**

***“ESTUDIO DEL ESTANDAR VOICEXML PARA EL ACCESO A DATOS VIA  
TELEFONICA. CASO PRÁCTICO: PROTOTIPO PARA CONSULTAS DE  
PLANILLAS TELEFÓNICAS”***

**TESIS DE GRADO**

**PREVIA LA OBTENCION DEL TITULO DE INGENIERO EN SÍSTEMAS INFORMÁTICOS**

**EDGAR ROLANDO MORALES CALUÑA**

**JEANNETH ALEGRIA CONCHA TRUJILLO**

**RIOBAMBA-ECUADOR**

**-2010-**

## **AGRADECIMIENTO**

Nuestro infinito agradecimiento a Dios, por habernos dado fuerza y conocimiento necesario para culminar nuestras metas. También la confianza y el apoyo de nuestros padres y hermanos, porque han contribuido positivamente para llevar a cabo esta jornada.

Queremos dejar constancia de nuestro profundo agradecimiento y sentimiento de gratitud a los señores Dr. Julio Santillán e Ing. Wladimir Castro director y miembro del proyecto de tesis como también a los señores Sixto Martin Borja (CTO y co-fundador de i6net) y Juan Pablo Godoy Macari (Ing. Conectividad y Redes) quienes a través del internet nos brindaron una ayuda importante en el desarrollo del proyecto.

## **DEDICATORIA**

A Dios, a mis padres quienes han sabido formarme con buenos sentimientos, hábitos y valores, de amar y respetar el lugar que me vio nacer, lo cual me han ayudado a salir adelante buscando siempre el mejor camino, además a mis hermanos, abuelitos, tíos y primos.

**GRACIAS**

*Edgar Morales Caluña*

## **DEDICATORIA**

A Dios, a mis padres por su comprensión y ayuda en momentos buenos y malos, quienes me enseñan a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento. Me han dado todo lo que soy como persona, mis valores, mis principios, mi perseverancia y mi empeño. También a mis hermanos, mis cuñadas y sobrinos quienes siempre confiaron en mí y me brindaron su apoyo incondicional.

**GRACIAS**

*Jeanneth Concha Trujillo.*

## FIRMAS RESPONSABLES Y NOTAS

ING. IVAN MENES

---

**DECANO DE LA FACULTAD DE  
INFORMÁTICA Y ELECTRÓNICA**

ING. RAÚL ROSERO

---

**DIRECTOR DE LA ESCUELA  
DE INGENIERÍA EN SISTEMAS**

DR. JULIO SANTILLAN

---

**DIRECTOR TESIS**

ING. WLADIMIR CASTRO

---

**MIEMBRO DE TESIS**

LCDO. CARLOS RODRIGUEZ

---

**DIRECTOR DEL CENTRO**

**DE DOCUMENTACIÓN**

## **RESPONSABILIDAD DEL AUTOR**

Nosotros, Edgar Rolando Morales Caluña y Jeanneth Alegría Concha Trujillo, somos los responsables de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo.

---

Edgar Rolando Morales Caluña

---

Jeanneth Alegría Concha Trujillo

## ABREVIATURAS

<b>ADSL:</b>	Asymmetric Digital Subscriber Line
<b>ASR:</b>	Automatic Speech Recognition
<b>ATA:</b>	Analog Telephone Adapter.
<b>DAHDI:</b>	Digium Asterisk Hardware Device Interface
<b>FXO:</b>	Foreign eXchange Office4
<b>FXS:</b>	Foreign eXchange Subscriber
<b>FIA:</b>	Form interpretation algorithm
<b>DTMF:</b>	Dual Tone Multi-Frecuency
<b>GPL:</b>	General Public License
<b>HTML:</b>	HyperText Markup Language
<b>IVR:</b>	Interactive Voice Response
<b>ISDN:</b>	Integrated Services Digital Network
<b>PSTN:</b>	Public Switched Telephone Network
<b>PBX:</b>	Private Branch Exchange
<b>PHP:</b>	Hypertext Pre-processor
<b>SIP:</b>	Session Initiation Protocol
<b>SRGS:</b>	Speech Recognition Grammar Specification
<b>TTS:</b>	Text To Speech
<b>VoIP:</b>	Voice Over Internet Protocol
<b>VXML:</b>	VoiceXML

**WWW:** World Wide Web

**XML:** Extensible Markup Language

# INDICE GENERAL

**AGRADECIMIENTO**

**DEDICATORIA**

**ABREVIATURAS**

**INTRODUCCIÓN**

## **CAPÍTULO I MARCO REFERENCIAL**

1.1.	INTRODUCCIÓN .....	- 27 -
1.2.	FORMULACION GENERAL DEL PROYECTO DE TESIS .....	- 28 -
1.2.1.	ANTECEDENTES .....	- 28 -
1.3.	JUSTIFICACION.....	- 29 -
1.4.	OBJETIVOS .....	- 32 -
1.4.1.	OBJETIVO GENERAL .....	- 32 -
1.4.2.	OBJETIVOS ESPECIFICOS .....	- 32 -
1.5.	HIPOTESIS .....	- 33 -
1.6.	METODOS Y TECNICAS .....	- 33 -

## **CAPÍTULO II MARCO TEÓRICO**

2.1	INTRODUCCIÓN .....	- 34 -
2.2	EVOLUCIÓN DE LA TELEFONÍA.....	- 35 -
2.2.1	TELEFONÍA FIJA .....	- 35 -
2.2.1.1	FUNCIONAMIENTO.....	- 35 -
2.2.1.2	CARACTERÍSTICAS.....	- 36 -
2.2.1.3	ARQUITECTURA .....	- 36 -
2.2.2	TELEFONÍA MÓVIL.....	- 37 -
2.2.2.1	FUNCIONAMIENTO.....	- 38 -
2.3	VOZ SOBRE IP.....	- 39 -

2.3.1	PROTOCOLO SIP .....	- 39 -
2.4	TECNOLOGÍAS ANEXAS AL ESTÁNDAR VOICEXML.....	- 40 -
2.4.1	PBX (Private Branch Exchange) .....	- 40 -
2.4.1.1	CONCEPTO .....	- 40 -
2.4.1.2	FUNCIONAMIENTO.....	- 40 -
2.4.1.3	FUNCIONES BÁSICAS DE UNA PBX .....	- 41 -
2.4.1.4	INTERCONEXIÓN DE LA RED TELEFÓNICA A LA PBX.....	- 41 -
2.4.1.5	ADAPTADORES ANALÓGICOS .....	- 41 -
2.4.1.6	TARJETAS ANALÓGICAS DE LA SERIE X400P .....	- 42 -
2.4.1.7	TELÉFONOS IP.....	- 42 -
2.4.1.7.1	ASPECTOS IMPORTANTES: .....	- 43 -
2.4.1.7.2	CARACTERÍSTICAS:.....	- 44 -
2.4.1.8	TELÉFONOS ANALÓGICOS.....	- 45 -
2.4.1.9	TELÉFONOS CELULARES.....	- 45 -
2.4.2	SISTEMAS DE RECONOCIMIENTO DE VOZ .....	- 46 -
2.4.2.1	CARACTERÍSTICAS PRINCIPALES.....	- 47 -
2.4.2.2	LIMITANTES.....	- 47 -
2.4.2.3	ARQUITECTURA DE UN SISTEMA DE RECONOCIMIENTO DE VOZ .....	- 48 -
2.4.2.4	SISTEMAS DEPENDIENTES DEL LOCUTOR.....	- 49 -
2.4.2.5	SISTEMAS INDEPENDIENTES DEL LOCUTOR .....	- 49 -
2.4.3	SISTEMAS TEXT TO SPEECH .....	- 50 -
2.4.3.1	FUNCIONAMIENTO.....	- 51 -
2.4.3.2	UTILIDAD DE LOS SINTETIZADORES DE VOZ .....	- 52 -
2.4.3.3	TIPOS DE SINTETIZADORES EXISTENTES .....	- 53 -
2.5	EL ESTÁNDAR VOICEXML.....	- 55 -
2.5.1	EL WORLD WIDE WEB Y VOICEXML .....	- 55 -

2.5.2	CARACTERÍSTICAS GENERALES .....	- 57 -
2.5.3	MODELO ARQUITECTÓNICO .....	- 57 -
2.5.3.1	APPLICATION HOSTING ENVIROMENT .....	- 58 -
2.5.3.2	VOICEXML INTERPRETER .....	- 58 -
2.5.3.3	VOICEXML INTERPRETER CONTEXT.....	- 59 -
2.5.3.4	IMPLEMENTATION PLATAFORM .....	- 59 -
2.5.4	METAS DE VOICEXML .....	- 59 -
2.5.5	ALCANCE DE VOICEXML.....	- 59 -
2.5.6	PRINCIPIOS DE DISEÑO DE VOICEXML. ....	- 60 -
2.5.7	REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DE LA PLATAFORMA .....	- 61 -
2.5.8	ELEMENTOS DE LA PROGRAMACIÓN VOICEXML.....	- 63 -
2.5.8.1	DIALOGOS Y SUBDIALOGOS.....	- 63 -
2.5.8.2	SESIONES.....	- 64 -
2.5.8.3	APLICACIONES.....	- 64 -
2.5.8.4	GRAMÁTICAS.....	- 65 -
2.5.8.4.1	GRAMÁTICAS DE VOZ.....	- 65 -
2.5.8.4.2	GRAMÁTICAS DTMF .....	- 67 -
2.5.8.4.3	GRAMÁTICAS BULTIN.....	- 67 -
2.5.8.4.4	ESTRUCTURA DE ARCHIVOS DE GRAMÁTICA.....	- 69 -
2.5.9	ESTRUCTURA DE UN DOCUMENTO VOICEXML .....	- 70 -
2.5.9.1	ELEMENTOS DEL VOICEXML .....	- 71 -
2.5.9.2	EJEMPLOS DE VOICEXML .....	- 82 -

### **CAPÍTULO III IMPLEMENTACIÓN DE LA CENTRALITA**

3.1	INTRODUCCIÓN .....	- 89 -
3.2	ELEMENTOS HARDWARE Y SOFTWARE .....	- 90 -
3.3	DESCRIPCIÓN DE LOS ELEMENTOS SOFTWARE.....	- 91 -

3.3.1	SISTEMA OPERATIVO.....	- 91 -
3.3.1.1	REQUERIMIENTOS DEL SISTEMA.....	- 91 -
3.3.1.2	CARACTERÍSTICAS PRINCIPALES.....	- 92 -
3.3.2	VXI Asterisk.....	- 93 -
3.3.3	DAHDI.....	- 93 -
3.3.4	ASTERISK.....	- 94 -
3.3.5	VERBIO TTS.....	- 94 -
3.3.6	VERBIO ASR.....	- 96 -
3.3.7	SERVIDOR WEB APACHE.....	- 96 -
3.3.7.1	CARACTERÍSTICAS.....	- 98 -
3.3.7.2	MÓDULOS.....	- 98 -
3.3.8	SERVIDOR DE BASE DE DATOS MYSQL.....	- 99 -
3.3.8.1	CARACTERÍSTICAS.....	- 99 -
3.3.9	PHP (Hypertext Preprocessor).....	- 99 -
3.3.9.1	CARACTERÍSTICAS.....	- 100 -
3.3.9.2	FRAMEWORK.....	- 101 -
3.3.9.3	ENTORNOS DE DESARROLLO PARA PHP.....	- 101 -
3.4	ELEMENTOS HARDWARE.....	- 102 -
3.4.1	TARJETA TELEFÓNICA DE COMUNICACIONES.....	- 102 -
3.4.1.1	TARJETA OPENVOX A400P.....	- 102 -
3.4.1.2	CARACTERÍSTICAS.....	- 103 -
3.5	IMPLEMENTACIÓN DE LA PLATAFORMA DE PRUEBAS.....	- 104 -
3.5.1	INSTALACIÓN DE CENTOS 5.4.....	- 104 -
3.5.2	INSTALACIÓN Y CONFIGURACIÓN DEL MÓDULO DAHDI Y TARJETA OPENVOX A400P.....	- 105 -
3.5.2.1	INSTALACIÓN DE DAHDI-LINUX.....	- 105 -

3.5.2.2	INSTALACIÓN DE DAHDI-TOOLS.....	- 106 -
3.5.2.3	DETECCIÓN DE LA TARJETA.....	- 107 -
3.5.2.4	DAHDI START/STOP .....	- 108 -
3.5.3	INSTALACIÓN Y CONFIGURACIÓN DE MODULO ASTERISK.....	- 108 -
3.5.3.1	INSTALACIÓN DE ASTERISK .....	- 108 -
3.5.3.2	COMPROBACIÓN DEL PROGRAMA DE INSTALACIÓN DE ASTERISK.....	- 109 -
3.5.3.3	ASTERISK START/STOP .....	- 109 -
3.5.4	INSTALACIÓN DE VXI* VOICEXML BROWSER .....	- 110 -
3.5.4.1	INSTALACIÓN DE PAQUETES VXI * .....	- 110 -
3.5.4.2	COMPROBAR INSTALACIÓN.....	- 110 -
3.5.4.3	AYUDA EN LÍNEA .....	- 111 -
3.5.4.4	ACTIVACIÓN DE LICENCIA COMERCIAL .....	- 112 -
3.5.4.5	DESINTALACIÓN DE VXI* .....	- 113 -
3.5.4.6	ACTUALIZACIÓN DE LA VERSION.....	- 113 -
3.5.4.7	CONFIGURACIÓN DE VXI * .....	- 113 -
3.5.4.8	FUNCIONES DE CONTROL .....	- 120 -
3.5.4.9	FUNCIONES DE LA CUENTA.....	- 121 -
3.5.4.10	VXI* START / STOP .....	- 123 -
3.5.4.11	USANDO VXI* .....	- 123 -
3.5.4.12	PAGINAS VOICEXML.....	- 124 -
3.5.4.13	USO DE LA SINTAXIS.....	- 124 -
3.5.4.14	COMANDOS DE ADMINISTRACIÓN.....	- 125 -
3.5.5	CONFIGURACIÓN DE LA PBX.....	- 125 -
3.5.6	INSTALACIÓN Y CONFIGURACIÓN DE LA PLATAFORMA DE PRUEBAS .	- 128 -
3.5.6.1	INSTALACIÓN DEL MODULO VERBIO-ENGINES-8.0-0.I386.RPM .....	- 128 -
3.5.6.2	INSTALACIÓN DEL MODULO VERBIO-CLIENTS-8.0-0.I386.RPM .....	- 128 -

3.5.6.3	INSTALACIÓN DEL MODULO VERBIO-TTS-CARLOS-ESEB-8.0-0.I386.RPM .....	129 -
3.5.6.4	INSTALACIÓN DEL MODULO VERBIO-ASR-ES_CA_EU_GA-8.0-0.I386.RPM ....	129 -
3.5.6.5	INSTALACIÓN DEL MODULO VERBIO_V8-0_2010-02-23_I686.tar .....	129 -
3.5.6.6	CONFIGURACIÓN.....	129 -
3.5.6.7	CONFIGURACIÓN DE TEXT TO SPEECH .....	130 -
3.5.6.8	CONFIGURACIÓN DEL SERVIDOR AUTOMATIC SPEECH RECOGNITION .	131 -
3.5.6.9	CONFIGURAR MODULES.CONF .....	132 -
3.5.6.10	CONFIGURAR VERBIO.CONF.....	133 -
3.5.6.11	INSTALACIÓN DEL ADMINISTRADOR WEB (WEBMIN).....	133 -
3.5.6.11.1	CONFIGURACIÓN DEL SERVIDOR WEB APACHE .....	134 -
3.5.6.11.2	INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR DE BASE DE DATOS MYSQL	- 136 -
3.5.7	PRUEBAS INICIALES DE LA PLATAFORMA .....	138 -
3.5.7.1	FUNCIONAMIENTO DE LA PBX .....	138 -
3.5.7.1.1	PROBAR EL SERVIDOR TEXT TO SPEECH .....	138 -
3.5.7.1.2	PROBAR EL SERVIDOR ASR .....	139 -

## **CAPITULO IV DISEÑO E IMPLEMENTACION DEL PROTOTIPO**

4.1	INTRODUCCIÓN .....	140 -
4.2	DISEÑO DE LA SOLUCIÓN .....	141 -
4.2.1	DESCRIPCIÓN DEL SERVICIO .....	141 -
4.2.2	CONSULTA DE PLANILLA TELEFÓNICA.....	141 -
4.2.3	SERVICIOS .....	141 -
4.2.4	AYUDA.....	141 -
4.2.5	DIAGRAMAS DE FLUJO DE LA SOLUCIÓN .....	141 -

4.2.5.1	APLICACIÓN TELEFÓNICA .....	- 142 -
4.2.5.2	APLICACIÓN WEB .....	- 143 -
4.2.5.3	MODELO ENTIDAD RELACIÓN .....	- 143 -
4.2.5.3.1	DICCIONARIO DE DATOS .....	- 144 -
4.2.6	LIMITACIONES DEL SISTEMA .....	- 147 -
4.2.7	DISEÑO DEL SISTEMA .....	- 147 -
4.2.7.1	APLICACIÓN TELEFÓNICA .....	- 147 -
4.2.7.1.1	INGRESO AL SISTEMA.....	- 147 -
4.2.7.1.2	ELECCIÓN DEL MENÚ .....	- 148 -
4.2.7.1.3	CONSULTA DE PLANILLAS.....	- 149 -
4.2.7.1.4	CONSULTA DE PLANILLAS POR VALOR TOTAL.....	- 149 -
4.2.7.1.5	CONSULTA DE PLANILLAS A DETALLE .....	- 149 -
4.2.7.1.6	CONSULTA DE SERVICIOS.....	- 150 -
4.2.7.1.7	AYUDA.....	- 150 -
4.2.7.2	MODELO FUNCIONAL DE LA APLICACIÓN TELEFÓNICA .....	- 150 -
4.2.7.2.1	CASO DE USO GENERAL ASOCIADO A LA APLICACIÓN TELEFÓNICA..	- 150 -
4.2.7.2.2	CASO DE USO ASOCIADO A LA CONSULTA DE PLANILLA TELEFÓNICAS .....	- 151 -
4.2.8	APLICACIÓN WEB .....	- 152 -
4.2.8.1	ARQUITECTURA DE LA APLICACIÓN WEB .....	- 153 -
4.2.8.2	MODELO FUNCIONAL DE LA APLICACIÓN WEB .....	- 154 -
4.2.8.2.1	CASO DE USO GENERAL ASOCIADO A LA APLICACIÓN WEB.....	- 154 -
4.3	IMPLEMENTACIÓN DEL PROTOTIPO.....	- 154 -
4.3.1	DESARROLLO DE LA APLICACIÓN WEB.....	- 156 -
4.3.1.1	PANTALLA DE INICIO .....	- 156 -
4.3.1.2	PANTALLA DE INGRESOS.....	- 156 -

4.3.1.3	PANTALLA DE INGRESOS DE OPCIONES .....	- 157 -
4.3.1.3.1	INGRESO DE CLIENTES.....	- 157 -
4.3.1.3.2	INGRESO DE LÍNEAS TELEFÓNICAS .....	- 158 -
4.3.1.3.3	INGRESO DE SERVICIO.....	- 158 -
4.3.1.4	PANTALLA DE GENERACIÓN DE FACTURA.....	- 159 -
4.3.1.4.1	FACTURA TOTAL .....	- 159 -
4.3.1.5	PANTALLA PARA MOSTRAR DATOS.....	- 161 -
4.3.1.5.1	PANTALLA PARA MOSTRAR DATOS DE FACTURA .....	- 161 -
4.3.1.5.2	PANTALLA PARA MOSTRAR DATOS DEL CLIENTE.....	- 162 -
4.3.1.5.3	PANTALLA PARA MOSTRAR DATOS DE SERVICIOS .....	- 162 -
4.3.2	DESARROLLO DE LA APLICACIÓN TELEFÓNICA .....	- 163 -
4.3.2.1	SCRIPT DE PRESENTACIÓN.....	- 163 -
4.3.2.2	SCRIPT DE SELECCIÓN DEL MENÚ PRINCIPAL.....	- 164 -
4.3.2.3	SCRIPT DE GENERACIÓN DE FACTURA .....	- 165 -
4.3.2.3.1	SCRIPT DE GENERACIÓN DE FACTURA TOTAL.....	- 166 -
4.3.2.3.2	SCRIPT DE GENERACIÓN DE FACTURA A DETALLE .....	- 168 -
4.3.2.4	CONSULTA A SERVICIOS .....	- 171 -
4.3.2.5	NAVEGACIÓN POR LA AYUDA .....	- 173 -
4.3.2.6	SALIR DEL SISTEMA.....	- 175 -

## **CAPITULO V COMPROBACIÓN DE LA HIPOTESIS**

5.1	INTRODUCCIÓN .....	- 176 -
5.2	DEMOSTRACIÓN DE LA HIPOTESIS .....	- 177 -
5.3	ACCESO A LA INFORMACIÓN VÍA WEB .....	- 177 -
5.3.1	WEB MÓVIL .....	- 177 -
5.3.2	WEB FIJA .....	- 178 -
5.4	ACCESO A LA INFORMACION VIA VOZ .....	- 179 -

5.4.1	REQUERIMIENTOS MINIMOS PARA CONNECTARSE A INTERNET .....	- 180 -
5.4.2	REQUERIMIENTOS HARDWARE Y SOFTWARE DE LOS CLIENTES PARA ACCESO A LAS APLICACIONES VOICEXML .....	- 181 -
5.5	EVALUACIÓN DE RECURSOS HARDWARE .....	- 181 -
5.6	EVALUACIÓN DE RECURSOS SOFTWARE.....	- 184 -
5.7	RESULTADOS OBTENIDO .....	- 186 -

**CONCLUSIONES**

**RECOMENDACIONES**

**RESUMEN**

**SUMMARY**

**GLOSARIO**

**BIBLIOGRAFÍA**

**ANEXOS**

## INDICE DE GRÁFICOS

Gráfico I.1: Arquitectura de Prototipo .....	- 32 -
Gráfico II.2: Public Switched Telephone Network .....	- 35 -
Gráfico II.3: Arquitectura de la PSTN .....	- 37 -
Gráfico II.4: Redes Celulares .....	- 38 -
Gráfico II.5: Representación de la red de telefonía celular .....	- 39 -
Gráfico II.6: Tarjeta X400P .....	- 42 -
Gráfico II.7: Central de Telefónica IP .....	- 43 -
Gráfico II.8: Sofphone X-lite .....	- 44 -
Gráfico II.9: Teléfono Analógico .....	- 45 -
Gráfico II.10: Teléfono Celular .....	- 46 -
Gráfico II.11: Arquitectura de Reconocimiento de Voz .....	- 47 -
Gráfico II.12: Componentes del Reconocimiento de Voz .....	- 48 -
Gráfico II.13: Estructura general de un sistema TTS .....	- 51 -
Gráfico II.14: VoiceXML vs Web .....	- 55 -
Gráfico II.15: Características de VoiceXML .....	- 57 -
Gráfico II.16: Arquitectura VoiceXML .....	- 58 -
Gráfico II.17: Interacción Hombre-Maquina .....	- 60 -
Gráfico II.18: Formato de un documento Voicexml. ....	- 63 -
Gráfico II.19: Transición entre documentos VoiceXML .....	- 64 -
Gráfico II.20: Ejemplo de <vxml> .....	- 71 -
Gráfico III.21: Escenario de la plataforma a implementar .....	- 104 -
Gráfico III.22: Instalación Satisfactoria de Dahdi Linux .....	- 106 -
Gráfico III.23: Instalación Satisfactoria de Dahdi Tools .....	- 107 -
Gráfico III.24: Reconocimiento de la Tajeta OpenVox A400P .....	- 107 -
Gráfico III.25: Instalación correcta de Asterisk .....	- 109 -
Gráfico III.26: Comprobación de Instalación de VXI* .....	- 111 -
Gráfico III.27: Ayuda en Línea .....	- 111 -
Gráfico III.28: Licencia Comercial .....	- 112 -
Gráfico III.29: Archivo vxml.conf .....	- 114 -
Gráfico III.30: Archivo chan_dahdi.conf .....	- 126 -

Gráfico III.31: Archivo dahdi-channels.conf .....	- 127 -
Gráfico III.32: Configuración del Servidor TTS Original.....	- 130 -
Gráfico III.33: Configuración del Servidor TTS Configurado.....	- 131 -
Gráfico III.34: Instalación del Webmin.....	- 133 -
Gráfico III.35: Inicio de Sesión Webmin .....	- 134 -
Gráfico III.36: Servidor Web Apache.....	- 135 -
Gráfico III.37: Archivo httpd.conf .....	- 136 -
Gráfico III.38: Servidor de Base de Datos MySQL.....	- 137 -
Gráfico III.39: Base de Datos Telefónica .....	- 137 -
Gráfico IV.40: Diagrama de Flujo de la Aplicación Telefónica.....	- 142 -
Gráfico IV.41: Diagrama de Flujo del comportamiento de la aplicación web .....	- 143 -
Gráfico IV.42: Diagrama de Flujo del Diccionario de Datos .....	- 144 -
Gráfico IV.43: Interacción de los módulos en el Ingreso al Sistema .....	- 148 -
Gráfico IV.44: Interacción de los Módulos para Elección del Menú .....	- 148 -
Gráfico IV.45: Interacción de los Módulos para la Consulta de Planillas Telefónicas	- 150
-	
Gráfico IV.46: Caso de uso general de la aplicación telefónica.....	- 151 -
Gráfico IV.47: Caso de uso para la Consulta de Planillas Telefónicas.....	- 152 -
Gráfico IV.48: Arquitectura de la aplicación web .....	- 153 -
Gráfico IV.49: Caso de Uso General correspondiente a la Aplicación Web.....	- 154 -
Gráfico IV.50: Pantalla de Inicio.....	- 156 -
Gráfico IV.51: Pantalla del Menú Principal .....	- 156 -
Gráfico IV.52: Pantalla del Ingreso de Opciones .....	- 157 -
Gráfico IV.53: Pantalla del Ingreso de Cliente .....	- 157 -
Gráfico IV.54: Pantalla del Confirmación .....	- 157 -
Gráfico IV.55: Pantalla del Ingreso de Línea Telefónica .....	- 158 -
Gráfico IV.56: Pantalla del Confirmación .....	- 158 -
Gráfico IV.57: Pantalla del Ingreso de Servicios .....	- 158 -
Gráfico IV.58: Pantalla del Confirmación .....	- 159 -
Gráfico IV.59: Pantalla de Generación de Factura.....	- 159 -
Gráfico IV.60: Pantalla de Ingreso de Datos en la Factura.....	- 159 -
Gráfico IV.61: Pantalla de Ingreso de Servicios en la Factura .....	- 160 -

Gráfico IV.62: Pantalla de Resultados de la Factura .....	- 160 -
Gráfico IV.63: Pantalla de Búsqueda de Factura .....	- 161 -
Gráfico IV.64: Pantalla de muestra de resultados de Factura.....	- 161 -
Gráfico IV.65: Pantalla de Búsqueda de Cédula del Cliente.....	- 162 -
Gráfico IV.66: Pantalla de muestra de Datos de Cliente.....	- 162 -
Gráfico IV.67: Pantalla de muestra de Datos de Cliente.....	- 162 -
Gráfico V.68: Representación Gráficas de las alternativas de Acceso a la Información ...	-
180 -	
Gráfico V.69: Evaluación de los Recursos Hardware .....	- 184 -
Gráfico V.70: Evaluación de los Recursos Software .....	- 185 -
Gráfico V.71: Resumen de Recursos .....	- 186 -
Gráfico V.72: Resumen Porcentual .....	- 187 -

## INDICE DE TABLAS

Tabla II.1: Voicexml vs Web.....	- 56 -
Tabla II.2: atributos del < vxml > .....	- 70 -
Tabla II.3: Etiquetas de VoiceXML .....	- 72 -
Tabla II.4: Atributos de los formularios. ....	- 74 -
Tabla II.5: Elementos de Entrada de un Formulario .....	- 76 -
Tabla II.6: Elementos de Control del Formulario.....	- 76 -
Tabla II.7: Atributos de < menú > .....	- 78 -
Tabla II.8: Atributos <option>.....	- 79 -
Tabla III.9: Elementos Hardware .....	- 90 -
Tabla III.10: Elementos Software .....	- 91 -
Tabla IV.11: Tabla Administrador .....	- 144 -
Tabla IV.12: Tabla Cliente.....	- 145 -
Tabla IV.13: Tabla Factura .....	- 145 -
Tabla IV.14: Tabla Items.....	- 146 -
Tabla IV.15: Tabla Línea Telefónica .....	- 146 -
Tabla IV.16: Tabla Servicios .....	- 146 -
Tabla V.17: Valorización para el Análisis.....	- 177 -
Tabla V.18: Alternativas para el acceso a las Aplicaciones Web y VoiceXML .....	- 179 -
Tabla V.19: Valores Cuantificados de los Recursos Hardware .....	- 179 -
Tabla V.20: Recursos hardware para el acceso a Aplicaciones .....	- 182 -
Tabla V.21: Valores Cuantificados de los Recursos Hardware .....	- 183 -
Tabla V.22: Recursos Software para el Acceso de Aplicaciones.....	- 184 -
Tabla V.23: Valores Cuantificados de los Recursos Software .....	- 185 -
Tabla V. 24: Tabla Porcentual de Recursos .....	- 187 -

## INDICE DE ESPACIO DE CODIGOS

Espacio Código II.1: Ejemplo de una gramática ABNF .....	- 65 -
Espacio Código II.2: Ejemplo gramatica.xml .....	- 66 -
Espacio Código II.3: Gramática Inline .....	- 67 -
Espacio Código II.4: Sintaxis Gramática Builtin .....	- 68 -
Espacio Código II.5: Ejemplos de Gramática Builtin .....	- 69 -
Espacio Código II.6: Estructura de un Archivo de Gramática .....	- 69 -
Espacio Código II.7: Sintaxis de la Etiqueta <prompt> .....	- 73 -
Espacio Código II.8: Uso de la etiqueta <prompt> .....	- 73 -
Espacio Código II.9: Reproducción de archivos de audio.....	- 74 -
Espacio Código II.10: Ejemplo de un formulario .....	- 77 -
Espacio Código II.11: Ejemplo de un Menú.....	- 80 -
Espacio Código II.12: Uso de Variables .....	- 81 -
Espacio Código II.13: Hola Mundo.....	- 82 -
Espacio Código II.14: Ejemplo de gramáticas locales e if-else .....	- 83 -
Espacio Código II.15: Script menú .....	- 84 -
Espacio Código II.16: Script temperature.php .....	- 85 -
Espacio Código II.17: Script store.php .....	- 86 -
Espacio Código II.18: Gramáticas por Referencia.....	- 87 -
Espacio Código II.19: Archivo “vdemo.txt” .....	- 88 -
Espacio Código III.20: Archivo extensions.conf .....	- 127 -
Espacio Código III.21: hola.vxml .....	- 128 -
Espacio Código III.22: Verificación de Instalación de Verbio .....	- 129 -
Espacio Código III.23: Archivo vxml.conf .....	- 131 -
Espacio Código III.24: Archivo modules.conf .....	- 132 -
Espacio Código III.25: Inicio de la PBX.....	- 138 -
Espacio Código III.26: TTS.....	- 138 -
Espacio Código III.27: ASR .....	- 139 -
Espacio Código IV.28: Archivo bienvenida.vxml.....	- 163 -
Espacio Código IV.29: Archivo menu.vxml.....	- 164 -
Espacio Código IV.30: Archivo planilla.vxml.....	- 165 -

Espacio Código IV.31: Archivo planillauno.vxml.....	- 166 -
Espacio Código IV.32: Archivo planuno.php .....	- 167 -
Espacio Código IV.33: Archivo planillados.vxml .....	- 168 -
Espacio Código IV.34: Archivo plandos.php .....	- 169 -
Espacio Código IV.35: Archivo planillatres.vxml.....	- 170 -
Espacio Código IV.36: Archivo servicios.vxml.....	- 171 -
Espacio Código IV.37: Archivo servicios.php .....	- 172 -
Espacio Código IV.38: Archivo ayuda.vxml.....	- 173 -
Espacio Código IV.39: Archivo opcion_ayuda.vxml .....	- 174 -
Espacio Código IV.40: Archivo salir.vxml.....	- 175 -

## INTRODUCCIÓN

Día a día somos mas las personas que necesitamos tener acceso a la información mediante la red global conocida como internet, actualmente esta red se encuentra saturado por el alto número de usuarios concurrentes a nivel mundial.

Esta situación ha motivado la búsqueda de nuevas formas de acceso a la información, en este caso mediante el uso de un recurso utilizado desde el inicio mismo de la comunicación entre las personas “la voz”, que utilizando las redes de telefonía publica, la telefonía móvil y la telefonía IP permitan el acceso a la información que se encuentra en el internet.

En el capitulo I Marco Referencial, presenta la realización del estudio del estándar VoiceXML, este estándar permite el acceso a la información mediante el uso de la voz, apoyado por motores de reconocimiento de voz y síntesis de voz, utilizando la infraestructura de la redes de telefonía fija y celular.

En el capitulo II Marco Teórico, presenta un estudio básicos de los conceptos de telefonía móvil, fija e IP, motores de reconocimiento de voz, conversores de texto a voz, gramáticas, además se presenta un estudio un poco mas profundo en aspectos del estándar VoiceXML como arquitectura, finalidades, metas y estructuras básicas de programación, en la parte final de este capitulo se presentan ejemplos básicos relacionados a la programación en el estándar de estudio.

En el capitulo III Implementación de la centralita, se describe las herramientas hardware y software utilizados en el proceso de instalación y configuración de la centralita sobre la cual funcionará el prototipo a implementarse en el siguiente capitulo.

En el capitulo IV Parte Aplicativa, se realiza el diseño e implementación de la aplicación VoiceXML para consultas de planillas telefónicas mediante la red de telefonía fija o celular.

En el capítulo V Comprobación de la hipótesis, se realiza un análisis para determinar las alternativas de acceso a las aplicaciones tanto Web como VoiceXML y al mismo tiempo indicar la cantidad de recursos hardware y software que requieren los clientes para su acceso. Basados en los análisis anteriores se comprobará la validez de la hipótesis de investigación.

# **CAPÍTULO I**

## **MARCO REFERENCIAL**

### **1.1. INTRODUCCIÓN**

En este capítulo se plantea el estudio del estándar VoiceXML, esta nueva tecnología propone nuevas formas de acceso a datos mediante la voz.

Se detalla los lineamientos y directrices que ayudarán a desarrollar el proyecto de una forma eficaz y objetiva para evitar contratiempos y redundancias en las actividades y tareas planificadas para encaminar a la investigación correctamente.

Se definirá las metas principales de este proyecto que se deberán cumplir de acuerdo a una ordenada planificación de recursos como: recursos financieros, recursos humanos, recurso tiempo.

## **1.2. FORMULACION GENERAL DEL PROYECTO DE TESIS**

### **1.2.1. ANTECEDENTES**

En los últimos años se ha tenido un gran avance en los sistemas de comunicación a tal punto que ha cambiado la forma de interactuar con los sistemas tradicionales que las empresas utilizan para prestar servicios informáticos. Ya no es suficiente que una empresa preste algún tipo de servicio utilizando un portal web, sino que se ha visto la necesidad de prestar este mismo tipo de servicios de otra forma, de ahí el avance tecnológico como es la “navegación por voz”. En la actualidad muchas empresas y organizaciones en el Ecuador poseen la necesidad de obtener este nuevo tipo de tecnología que ayude al progreso institucional.

Los diálogos y el intercambio de información entre un usuario, es un servicio muy útil y necesario que debe ser implementado y estar al alcance de todas las personas con bajos costos y de fácil uso. Es por ello que se observa la necesidad de desarrollar una aplicación con capacidad de reconocimiento de habla que permita a través de la voz humana y empleando reconocimiento de lenguaje natural, atender automáticamente las peticiones que una persona posea. Lógicamente el desarrollo de cada una de las aplicaciones para satisfacer dicha necesidad deben ser manejadas en énfasis a los requerimientos de cada usuario y a las limitaciones que posea, por ejemplo personas no videntes.

Uno de los estándares más ampliamente usado es Voice Extended Markup Language (VoiceXML), el cual ha tenido muy buena aceptación en el mercado debido a sus múltiples ventajas y la posibilidad de integrarse a muchos otros sistemas, por ejemplo los sistemas de diálogos interactivos, es decir aquellos que brindan información de forma dinámica (consulta de cuentas bancarias, acceso al correo electrónico, etc). Estos sistemas de diálogos se conocen como IVRs (Interactive Voice Response), los cuales son programas informáticos que se diseñen con la finalidad de emular a un ser humano en un dialogo oral con otra persona, por lo tanto se apoyan en otros sistemas o estándares, tal es el caso de TTS (text to speech) y ASR (Speech Recognition).

VXML es un lenguaje de gestión de diálogos muy potente, que ofrece una gran variedad de posibilidades de control sobre los servicios de reconocimiento y generación de voz. Es tan grande que obliga a realizar un diseño muy fino de la implementación de la plataforma para conseguir admitir todos los servicios ofrecidos. Sin embargo carece de las herramientas comunes para implementar la lógica de un programa.

VXML fue desarrollado por la W3C para la creación de diálogos vocales que hacen uso de recursos de reconocimiento de voz, síntesis de voz, reconocimiento DTMF (Dual Tone Multi-Frequency), audio digital y grabación de audio en el contexto de aplicaciones telefónicas o interacción a través de la voz en general.

La gran ventaja en el uso de VXML es que se ha convertido en el estándar más utilizado para el diseño de aplicaciones vocales, proporcionando una arquitectura abierta para la programación de este tipo de aplicaciones. Se basa en el XML beneficiándose así, de todas las ventajas que ofrece este lenguaje. Como también se beneficia de los avances continuos de Internet, en términos de eficiencia, anchos de banda y calidad de servicio.

Internet concentra gran cantidad de información que hasta ahora sólo era accesible a través de un navegador. Abrir paso para que todo este contenido pueda volcarse sobre otros dispositivos es un reto que los portales de voz han asumido. La posibilidad de “hablar” con la Red y que “responda”, es decir, navegar a través de la gran tela de araña gracias a la voz es viable gracias al lenguaje VoiceXML. Las posibilidades que se abren son infinitas a nivel de servicios y especialmente, en el caso de determinados colectivos sociales.

### **1.3. JUSTIFICACION**

Con los avances tecnológicos se ha logrado contar con sistemas de comunicación más potentes, prácticos y eficaces. Por ello es necesario realizar investigaciones sobre nuevas tecnologías y formas de comunicación que ayuden a la sociedad en sus diversas áreas y utilizando soportes tecnológicos actuales. VoiceXML es el equivalente al lenguaje de programación HTML y es el estándar más popular para el desarrollo de

aplicaciones de voz. Funciona mediante un navegador de voz cuya salida es audio, y cuya entrada es audio y teclado.

La entrada de audio está controlada por un reconocedor de voz integrado con el navegador de VoiceXML. La salida de audio consiste en audio pre-grabado y/o en voz sintetizada por un sistema de Text-To-Speech. Un navegador de voz normalmente funciona en base a una pasarela de voz que es un nodo conectado tanto a Internet como a la RTC (Red Telefónica Conmutada). La pasarela de voz puede soportar cientos o miles de llamadas simultáneas y permite que accedan a ella cualquiera de los 2, 500, 000,000 millones de teléfonos que se estima que hay en el mundo, entre los móviles y los teléfonos fijos de toda la vida.

Una de las grandes ventajas de este tipo de aplicaciones es que el usuario con facilidad escucha la información a través de un teléfono y no necesitan ni tiempo de arranque, ni sistemas operativos, ni conocimientos previos sobre computación. Además la voz, siempre ha sido la forma natural de comunicarse a través del teléfono y el acceso más común para toda la población.

El lenguaje VoiceXML se beneficia además de:

- El crecimiento del WWW – World Wide Web y de las posibilidades que este ofrece.
- Las mejoras en el campo del reconocimiento de voz y la síntesis de texto al habla.
- La expansión del WWW a niveles más allá del ordenador de sobremesa.

Actualmente a nivel mundial existen aplicaciones que se han desarrollado de manera particular y con fines específicos en el estándar VoiceXML, los cuales no han tenido un impacto general tanto para la comunidad científica como para la sociedad que lo necesita. Cabe recalcar que cada uno de estos proyectos ha sido diseñado a partir de requerimientos específicos de cada empresa y con el financiamiento económico de la misma. Podríamos nombrar algunos por ejemplo: proyecto GEMINI, Proyecto eEuropa, Sistema de Información basado en VXML para la universidad Francisco Gavidia, Aplicación de Soporte VXML con interfaz web y base de datos para las plataformas

telefónicas de Sixbell Nekotec Solutions. Pero existe la deficiencia de acceso a estos proyectos debido que su desarrollo se ha realizado bajo la orientación de código propietario.

En lo que respecta a nuestro país y específicamente en nuestra institución se desconoce la realización de algún proyecto y/o estudio realizado, según la fuente de datos de las investigaciones que se han desarrollado en la misma. El resultado de esta investigación reflejará un prototipo que permitirá demostrar el uso del estándar VoiceXML aplicado a la realización de consultas de información mediante el uso de un teléfono. VoiceXML permite crear un script de la conversación que el usuario tiene con una aplicación y es el Servidor quien la ejecuta. Según las respuestas del usuario el diálogo se complementa con la respuesta de unos archivos de audio.

El sistema básicamente se compone de un Servidor, Voice Browser y el Lenguaje VoiceXML para el desarrollo de la aplicación de voz.

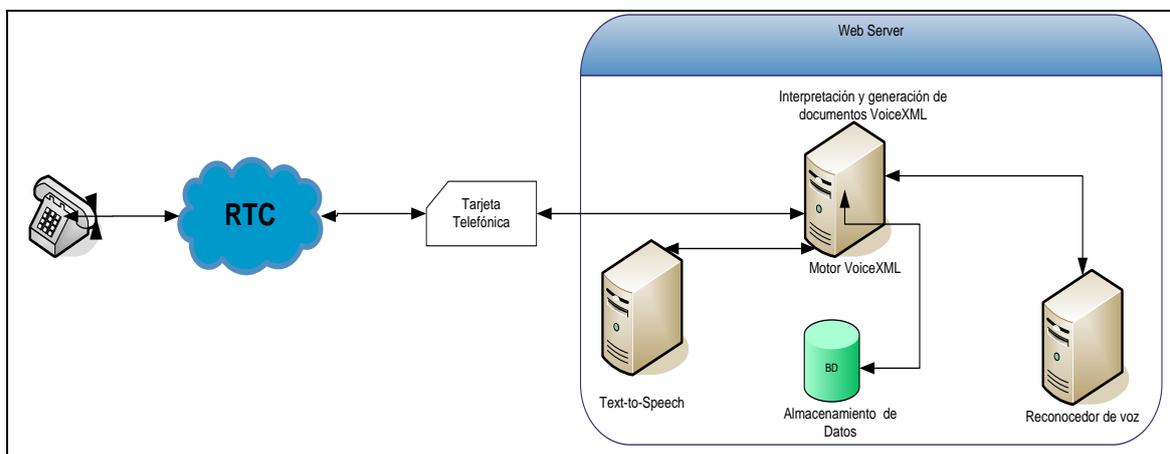
El prototipo funcionará de una forma similar a la que se plantea a continuación:

- Recoger una solicitud de información (planilla de teléfono) hecha por un usuario a través de una línea telefónica.
- La RTC servirá para enlazar dos equipos terminales (teléfono y pc) mediante un circuito físico que se establece específicamente para la comunicación y que desaparece una vez que se ha completado la misma. Se trata por tanto, de una red de telecomunicaciones conmutada.
- La aplicación solicitará al cliente el número telefónico a consultar el saldo.
- El usuario expresará el número telefónico a consultar.
- La aplicación recogerá la información del número telefónico ingresado por el usuario.
- El prototipo capturará el número telefónico a ser consultado.
- Se desarrollará la verificación de la información en una base de datos de prueba.
- El prototipo se encargará de realizar la consulta telefónica del saldo del número requerido.
- La información de respuesta desde el computador será envía en modo voz al usuario que se encuentra en la línea telefónica.

- Y por último el usuario escuchará respuesta a la petición requerida.

## ARQUITECTURA DEL PROTOTIPO

Ver Gráfico I.1.



**Gráfico I.1: Arquitectura de Prototipo**

### 1.4. OBJETIVOS

#### 1.4.1. OBJETIVO GENERAL

- Realizar el estudio del estándar VoiceXML aplicado al desarrollo de un prototipo para consultas de planillas telefónicas.

#### 1.4.2. OBJETIVOS ESPECIFICOS

- Investigar la arquitectura y el funcionamiento del estándar VoiceXML.
- Estudiar la estructura de programación sobre el estándar VoiceXML.
- Diseñar e implementar el prototipo para consultas de planillas en base a la voz usando VXML.
- Integrar el prototipo a la RTC mediante el uso de una tarjeta telefónica digital.
- Evaluar y discutir los resultados del prototipo.

## 1.5. HIPOTESIS

- El estudio del estándar VoiceXml permitirá incrementar las alternativas de consultas de datos con la disminución del empleo de recursos.

## 1.6. METODOS Y TECNICAS

### 1.6.1. Tipo de Investigación

La presente investigación es de tipo exploratoria, en vista que el estándar VoiceXML no ha sido investigado en nuestro medio. Con el conocimiento adquirido en este proyecto crearemos un prototipo que servirá para futuras investigaciones y de esta manera aportando nuevos conocimientos a la comunidad científica.

### 1.6.2. Métodos

**Método científico:** Este proyecto hará uso del método científico porque nos da un conjunto de reglas y lineamientos que regirá el procedimiento para ejecutar esta investigación.

### 1.6.3. Técnicas

- **Observación:** Utilizaremos herramientas hardware y software para observar la navegación de voz.
- **Lluvia de ideas:** La lluvia de ideas será una de las principales técnicas para recolectar la información y para procesarla, por el mismo hecho que esta investigación se realizará en equipo.
- **Pruebas finales de prototipo:** Las pruebas que se desarrollaran en nuestro prototipo bajo diferentes parámetros permitirán demostrar la hipótesis que se ha planteado.

### 1.6.4. Fuentes para recopilar información

- **Fuentes primarias**

Para solucionar posibles dudas que se presenten en el desarrollo del proyecto se acudirá a los docentes de la Facultad los cuales puedan orientar.

- **Fuentes secundarias**

Textos electrónicos

Documentos electrónicos

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1 INTRODUCCIÓN**

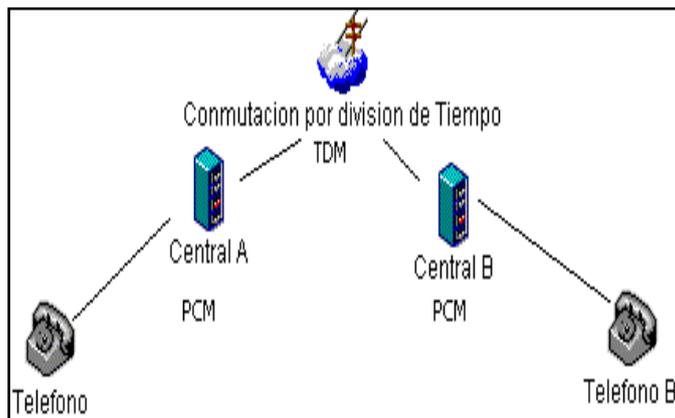
En el presente capítulo se citan los diferentes conceptos más importantes que están relacionados con el desarrollo de la investigación sobre el estándar VoiceXML.

El desarrollo de las diferentes tecnologías actuales es el resultado de nuevas investigaciones en diferentes campos, es así que el estándar VoiceXML para su correcto funcionamiento se encuentra ligado al desarrollo de la telefonía tradicional (PSTN), el crecimiento de los servicios de VoIP (Voice Over Internet Protocol) y la más importantes en nuestros días, la telefonía celular.

## 2.2 EVOLUCIÓN DE LA TELEFONÍA

### 2.2.1 TELEFONÍA FIJA

La telefonía fija representa todavía la parte principal del mercado de telecomunicaciones en el mundo. Es una de las redes de mayor cobertura a nivel mundial.



**Gráfico II.2: Public Switched Telephone Network**

#### 2.2.1.1 FUNCIONAMIENTO

Un teléfono está formado por dos circuitos que funcionan juntos: el circuito de conversación, que es la parte analógica, y el circuito de marcación, que se encarga de la marcación y llamada. Tanto las señales de voz, como las de marcación y llamada (señalización), así como la alimentación, comparten el mismo par de hilos, a esto a veces se le llama "señalización dentro de la banda (de voz)".

La impedancia característica de la línea es  $600\Omega$ . Lo más llamativo es que las señales procedentes del teléfono hacia la central y las que se dirigen a él desde ella viajan por esa misma línea de sólo 2 hilos. Para poder combinar en una misma línea dos señales (ondas electromagnéticas) que viajen en sentidos opuestos y para luego poder separarlas se utiliza un dispositivo llamado transformador híbrido o bobina híbrida, que no es más que un acoplador de potencia (duplexor).<sup>1</sup>

---

<sup>1</sup> El teléfono  
<http://es.wikipedia.org/wiki/Tel%C3%A9fono>

### **2.2.1.2 CARACTERÍSTICAS**

- Ofrece a cada usuario un circuito para señales analógicas con una banda base de 4KHz para cada conversación entre dos domicilios. Esta banda incluye espacios para banda de guarda anti-tralape (anti-aliasing) y para eliminación de interferencias provenientes de las líneas de «Distribución domiciliar de potencia eléctrica»
- Única red con cobertura y capilaridad nacional, donde por capilaridad se entiende la capacidad que tiene la red para ramificarse progresivamente en conductores que llevan cada vez menor tráfico.
- Capacidad de interconexión con las redes móviles. Es decir, la telefonía básica es entre aparatos fijos y mediante equipos de última tecnología permite interactuar con la red móvil lo que hace más interesante sus servicios tanto para las dos redes, cada una con tecnologías diferentes. El costo para el usuario por la ocupación del circuito depende de la distancia entre los extremos y la duración de la conexión
- Consta de Medios de Transmisión y Centrales de conmutación. Los Medios de transmisión entre centrales se conocen como Troncales, y en la actualidad transportan principalmente señales digitales sincronizadas, usando tecnologías modernas, sobre todo ópticas. En cambio, los medios de transmisión entre los equipos domiciliarios y las centrales, es decir, las líneas de acceso a la red, continúan siendo pares de cobre, y se les sigue llamando líneas de abonado (abonado viene del Francés y significa subscriptor). Las demás formas de acceder del domicilio a la central local, tales como enlaces inalámbricos fijos, enlaces por cable coaxial o fibra óptica, u otros tipos de líneas de abonado que transportan señales digitales (como ISDN o xDSL), no se consideran telefonía básica.

### **2.2.1.3 ARQUITECTURA**

Los componentes incluidos en la arquitectura de toda PSTN son:

Terminal de abonado y línea telefónica de abonado (bucle local), centrales de Conmutación de circuitos, sistema de transmisión, sistema de señalización. Esto correspondería exclusivamente al funcionamiento del teléfono.

Para poder transmitir datos por una Red Telefónica Conmutada, se necesita añadir otro elemento a la disponibilidad en tanto sitios de la infraestructura de la PSTN la constituye en la solución más apropiada para introducir rápidamente cualquier servicio de telecomunicación nuevo.



**Gráfico II.3: Arquitectura de la PSTN<sup>2</sup>**

### 2.2.2 TELEFONÍA MÓVIL

Las telecomunicaciones han avanzado tanto que hoy en día es común que un porcentaje alto de personas posean un teléfono móvil. El teléfono móvil es un dispositivo inalámbrico electrónico que permite tener acceso a la red de telefonía celular o móvil. Se denomina celular debido a las antenas repetidoras que conforman la red, cada una de las cuales es una célula, si bien existen redes telefónicas móviles satelitales. Su principal característica es su portabilidad, que permite comunicarse desde casi cualquier lugar. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado otras funciones como son cámara fotográfica, agenda, acceso a Internet, reproducción de vídeo e incluso GPS y reproductor mp3.

El primer antecedente respecto al teléfono móvil es de la compañía Motorola, con su modelo DynaTAC 8000X. El modelo fue diseñado por el ingeniero de Motorola Rudy Krolopp en 1983. El modelo pesaba poco menos que un kilo y un valor de casi 4.000

<sup>2</sup> Arquitectura de la PSTN

[http://hosting.udlap.mx/profesores/luisg.guerrero/Cursos/IE445/Presentaciones/Pres1\\_PSTNISDN\\_Wapeches.ppt](http://hosting.udlap.mx/profesores/luisg.guerrero/Cursos/IE445/Presentaciones/Pres1_PSTNISDN_Wapeches.ppt)

dólares. Krolopp se incorporaría posteriormente al equipo de investigación y desarrollo de Motorola liderado por Martin Cooper. Tanto Cooper como Krolopp aparecen como propietarios de la patente original. A partir del DynaTAC 8000X, Motorola desarrollaría nuevos modelos como el Motorola MicroTAC, lanzado en 1989, y el Motorola StarTAC, lanzado en 1996 al mercado.

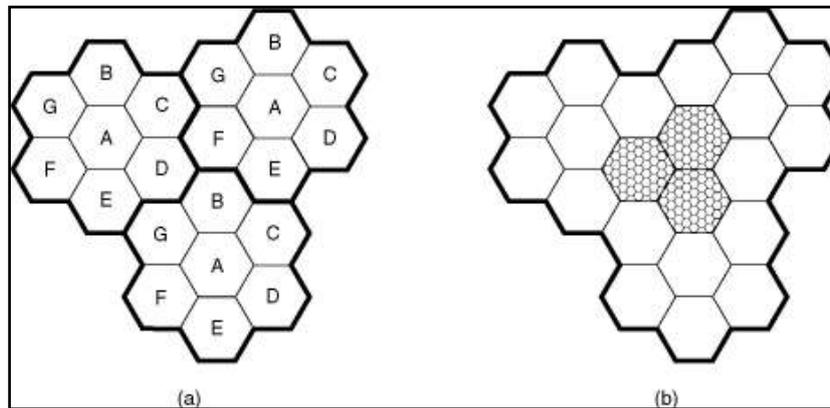


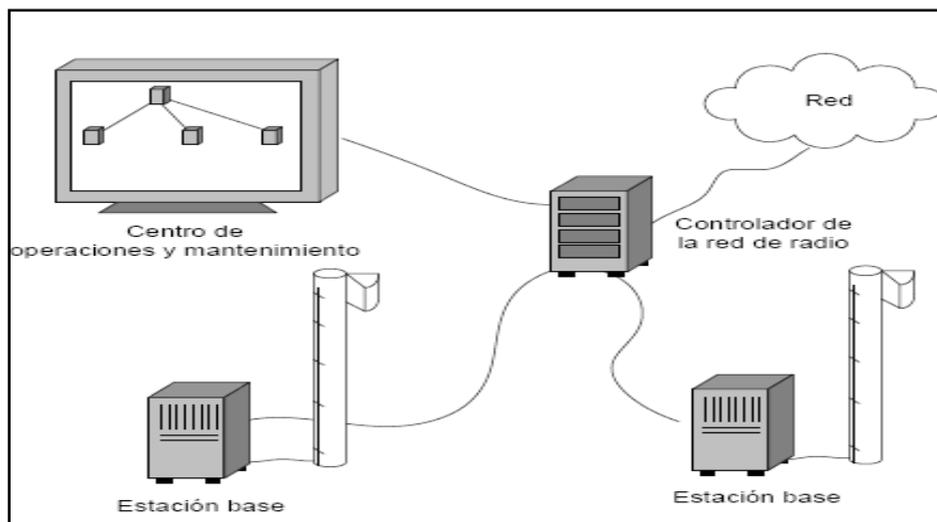
Gráfico II.4: Redes Celulares<sup>3</sup>

### 2.2.2.1 FUNCIONAMIENTO

La comunicación telefónica es posible gracias a la interconexión entre centrales móviles y públicas. Según las bandas o frecuencias en las que opera el móvil, podrá funcionar en una parte u otra del mundo. La telefonía móvil consiste en la combinación de una red de estaciones transmisoras-receptoras de radio (repetidores, estaciones base o BTS) y una serie de centrales telefónicas de conmutación de 1er y 5to nivel (MSC y BSC respectivamente), que posibilita la comunicación entre terminales telefónicos portátiles (teléfonos móviles) o entre terminales portátiles y teléfonos de la red fija tradicional.

---

<sup>3</sup> Libro de Red de Telefonía Móvil  
Michael Kircher & Prashant Jain



**Gráfico II.5: Representación de la red de telefonía celular**

### **2.3 VOZ SOBRE IP**

La tecnología de VoIP consiste básicamente en la transmisión de voz sobre redes IP. Se originó a partir de distintos factores que entre sí se potencian. El crecimiento de Internet y el desarrollo de métodos de compresión de voz, transmisión en tiempo real, y principalmente la necesidad de estar siempre comunicados, son la base y antesala a la VoIP.

La voz sobre IP consiste en convertir las señales de voz en paquetes de datos, los cuales son transportados de forma comprimida en lugar de utilizar las tradicionales líneas telefónicas se utilizan redes de datos.

#### **2.3.1 PROTOCOLO SIP**

SIP (Session Initiation Protocol) fue aceptado y presentado oficialmente como un estándar IETF en 1999. SIP es un protocolo de control de la capa de aplicación en el modelo ISO/OSI.

Fue diseñado para iniciar, modificar y terminar sesiones multimedia con uno o más participantes. Las sesiones pueden ser: llamadas telefónicas sobre Internet, distribución de contenido multimedia y video conferencias. La forma en que interactúan dos dispositivos SIP es través de mensajes de señalización. Estos cumplen los propósitos básicos de:

- Registrar un usuario y sistema.
- Invitar a unirse a una sesión.
- Negociar términos y condiciones de una sesión.
- Establecer un stream entre dos o más puntos finales.
- Finalizar una sesión.

## **2.4 TECNOLOGÍAS ANEXAS AL ESTÁNDAR VOICEXML**

Para el correcto funcionamiento de una aplicación desarrollada sobre el estándar VoiceXML se utilizan tres elementos indispensables y básicos, los mismos que se detallan brevemente a continuación.

### **2.4.1 PBX (Private Branch Exchange)**

#### **2.4.1.1 CONCEPTO**

Una PBX, también conocida como una central telefónica privada, es un servicio de conmutación telefónica para una entidad privada o una empresa y su objetivo es el de proveer comunicaciones de voz (e incluso datos) a los usuarios dentro de la organización. Estos usuarios suelen compartir un conjunto de líneas externas para realizar y recibir llamadas desde la PSTN. Inicialmente, una central telefónica era una oficina en la que convergían todos los cables que conducían a las comunicaciones de voz. En estas centrales existían operadores que recibían las llamadas y las conectaban al teléfono con el cual los abonados deseaban comunicarse.

#### **2.4.1.2 FUNCIONAMIENTO**

Los usuarios de una PBX pueden fácilmente comunicarse entre sí dentro de su organización, marcando simplemente el número asignado de la extensión. A menudo, para comunicarse con una persona que no pertenezca a la red interna, la central privada debe enrutar la llamada hacia la Red Telefónica Pública Conmutada, lo que puede involucrar el marcado de un código de acceso (0 o 9) junto al número telefónico de la persona a la que deseamos llamar. Las labores de enrutamiento y conmutación de circuitos son elementos básicos de una PBX; sin embargo, en la actualidad es innumerable la cantidad de servicios adicionales que puede prestar para las organizaciones y empresas, trayendo beneficios al mundo de la actividad comercial, industrial y financiera.

### **2.4.1.3 FUNCIONES BÁSICAS DE UNA PBX**

Las funciones principales de una PBX son:

- Establecer conexiones entre los aparatos telefónicos de dos usuarios, mediante el mapeo de un número telefónico con un teléfono físico.
- Mantener las conexiones durante el tiempo que los usuarios lo requieran.
- Proveer información de estadísticas de llamada.

### **2.4.1.4 INTERCONEXIÓN DE LA RED TELEFÓNICA A LA PBX**

Para poder conectarse a los sistemas telefónicos llámese fijo, móvil o IP, es necesario que la plataforma implementada en el proyecto tenga una forma para conectarse a estos sistemas. Esta conexión se puede hacer de diferentes maneras bien utilizando interfaces de telefonía analógicas o digitales y en el caso de telefonía IP, se puede hacer utilizando el protocolo SIP.

### **2.4.1.5 ADAPTADORES ANALÓGICOS**

Son dispositivos con una interfaz para conectar un teléfono analógico (slot para conector RJ-11) y otra interfaz para conectar a la red (slot para conector RJ-45). Básicamente su función es la de proveer señalización FXO a los teléfonos, es decir, se comporta como un dispositivo FXS. Se explicará brevemente estos dos términos:

- FXO: Foreign eXchange Office, es la interfaz que se conecta a la red de Telefonía Básica (RTB, PSTN) o a una PBX y normalmente está presente en todos los teléfonos analógicos. Recibe la señalización dada por la FXS.
- FXS: Foreign eXchange Subscriber, es la interfaz que se conecta directamente a un teléfono analógico y le brinda tono de timbrado y voltaje, entre otras cosas. En un escenario convencional (telefonía analógica), el FXS está en la central de conmutación, brindando señalización al dispositivo FXO (teléfono analógico).

Se tienen dos posibilidades para usar teléfonos analógicos en una red VoIP: Una es que el servidor de VoIP tenga una tarjeta con módulos FXS y la otra es tener en la red ciertos gateways que conviertan la señal analógica en datos IP. De esta forma, la PBX IP se comunica con los teléfonos analógicos a través de los gateways usando los protocolos de señalización mencionados anteriormente.

#### 2.4.1.6 TARJETAS ANALÓGICAS DE LA SERIE X400P

Estas tarjetas tienen interfaces FXS, lo que permite conectar:

- Teléfonos (FXS)
- Centralitas hacia sus líneas externas (FXS)

Así como interfaces FXO, típicamente para repartir las llamadas generadas en la centralita o teléfonos y/o líneas telefónicas de respaldo. (Los FXO También pueden conectarse a extensiones de centralitas).

Cuenta además con dos relés para dar continuidad entre los puertos FXS con los puertos FXO en caso de ausencia de alimentación. Estas tarjetas válidas para equipos Teldat VyDa, así como Teldat Atlas. Realiza las funciones de codificación/decodificación (G.711, G729, G723), así como otras auxiliares (VAD, CNG, ECN) mediante DSPs integrados, sin consumir recursos del equipo base. La tarjeta 2+2(FXS/FXO) se complementa con las funciones realizadas por el equipo base (VyDa): H323/SIP, funciones de centralita, planes de numeración, etc.<sup>4</sup>

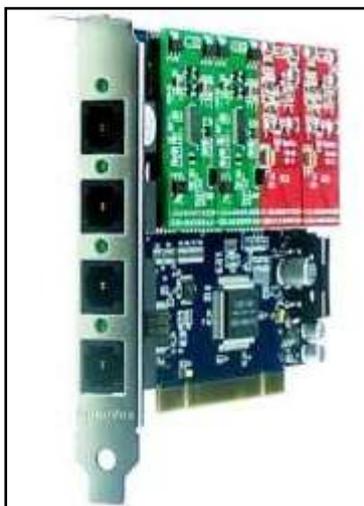


Gráfico II.6: Tarjeta X400P

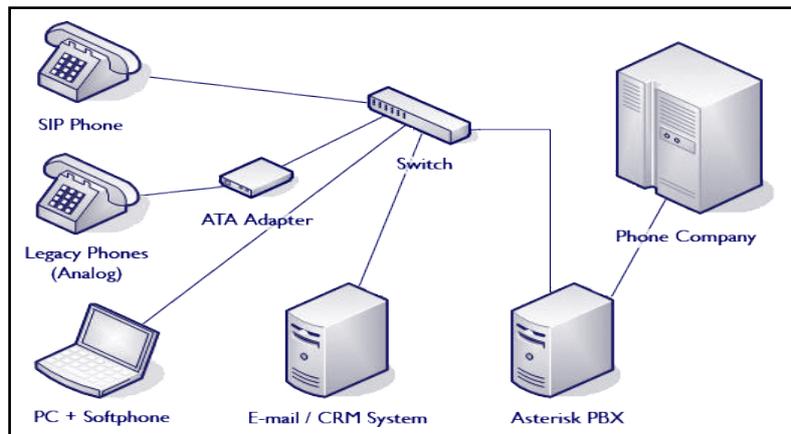
#### 2.4.1.7 TELÉFONOS IP

Existen dos tipos de teléfonos IP: softphones y hardphones. En cuanto a características y funcionalidades son idénticos, por lo que no se hará distinción entre uno u otro. La única diferencia que tienen es que mientras el hardphone es un dispositivo físico

---

<sup>4</sup> Openvox A400P  
[http://www.fonlogic.net/Tarjetas\\_Openvox\\_A400P.html](http://www.fonlogic.net/Tarjetas_Openvox_A400P.html)

(teléfono tangible), el softphone funciona de la misma forma que cualquier programa en la computadora.



**Gráfico II.7: Central de Telefónica IP**

La principal ventaja de un teléfono IP es la movilidad, es decir, se puede mover el equipo en cualquier punto de la red y se mantiene su mismo número de extensión, esto no es posible con los teléfonos analógicos, donde cada ranura identifica a un número. Esto conlleva un reuso de la infraestructura de datos para pasar voz, abaratando costos a largo plazo. Sin embargo también existen desventajas. La principal es el retardo producido por el proceso de codificación – transporte – decodificación, en la comunicación entre la red pública y cualquier extensión IP, el cual produce eco en los teléfonos al oírse los usuarios a sí mismos luego de cierto período de tiempo. Esto se soluciona eficazmente con un cancelador de eco.

#### **2.4.1.7.1 ASPECTOS IMPORTANTES:**

- Reducción costos en telefonía y en ciertos casos sin costo.
- Donde quiera que estés conectado a Internet, se podrá recibir llamadas.
- Números telefónicos gratuitos para usar con VoIP están disponibles en Estados Unidos de América, Reino Unido y otros países de organizaciones como Usuario VoIP
- Los agentes de Call center usando teléfonos VoIP pueden trabajar en cualquier lugar con conexión a Internet lo suficientemente rápida.

Los aspectos mas importantes radican en la versatilidad del producto este te permitirá hacer llamadas internas en tu empresa ya sea a través de teléfonos IP mismos que son semejantes a los teléfonos convencionales o a través de softphone que no es otra cosa que un software simulador de teléfono IP que se utiliza con la ayuda de un headphone. Se pueden realizar llamadas internacionales, nacionales, etc. Por supuesto en este punto deben considerarse ciertos casos ya que pueden ser llamadas vía internet si ambos involucrados en la conversación usan telefonía IP la llamada será gratuita, pero si uno de los dos esta utilizando telefonía tradicional el la llamada tendrá costo<sup>5</sup>.



Gráfico II.8: Sofphone X-lite

#### 2.4.1.7.2 CARACTERÍSTICAS:

- **Usuario:**
  - Fácil uso a través de teléfonos IP o softphone
  - Solo necesita conocer el número de extensión para poder comunicarse
  - No necesita capacitación adicional para su uso.
- **Técnicas:**
  - VOIP y PSTN Asterisk es en esencia una alternativa ideal a los PBX convencionales, no solamente manteniendo las funciones básicas de un PBX, sino muchas otras aplicaciones incluyendo telefonía IP
- **Administrador:**
  - Agregar extensiones

---

<sup>5</sup> Central Telefonía IP  
<http://portal2.conlinux.net/centralIP>

- Conocer estadísticas
- Administrar usuarios
- Conocer Reportes
- Configurar Fax
- Configurar Email

#### **2.4.1.8 TELÉFONOS ANALÓGICOS**

Para poder usar teléfonos analógicos se debe tener hardware especial instalado en el servidor. Se trata de módulos FXS que brindan señalización y energía a los teléfonos, tal como lo hace una central pública a un usuario convencional. Estos módulos FXS pueden estar directamente conectados a la central de conmutación (PBX) o a la red LAN, en cuyo caso el dispositivo se denomina ATA (Analog Telephone Adapter).<sup>6</sup>



**Gráfico II.9: Teléfono Analógico**

#### **2.4.1.9 TELÉFONOS CELULARES**

Los celulares son dispositivos electrónicos con diseños intrincados, con partes encargadas de procesar millones de cálculos por segundo para comprimir y descomprimir el flujo de voz, contiene las siguientes partes<sup>7</sup>:

- Un circuito integrado que contiene el cerebro del teléfono.
- Una antena
- Una pantalla de cristal líquido (LCD)
- Un teclado pequeño

---

<sup>6</sup> Definición de Teléfono Analógico  
[http://www.pcmag.com/encyclopedia\\_term/0,2542,t=analog+phone&i=37753,00.asp](http://www.pcmag.com/encyclopedia_term/0,2542,t=analog+phone&i=37753,00.asp)

<sup>7</sup> Definición de Teléfono Celular  
<http://www.monografias.com/trabajos14/celularhist/celularhist.shtml>

- Un micrófono
- Una bocina
- Una batería



**Gráfico II.10: Teléfono Celular**

#### **2.4.2 SISTEMAS DE RECONOCIMIENTO DE VOZ**

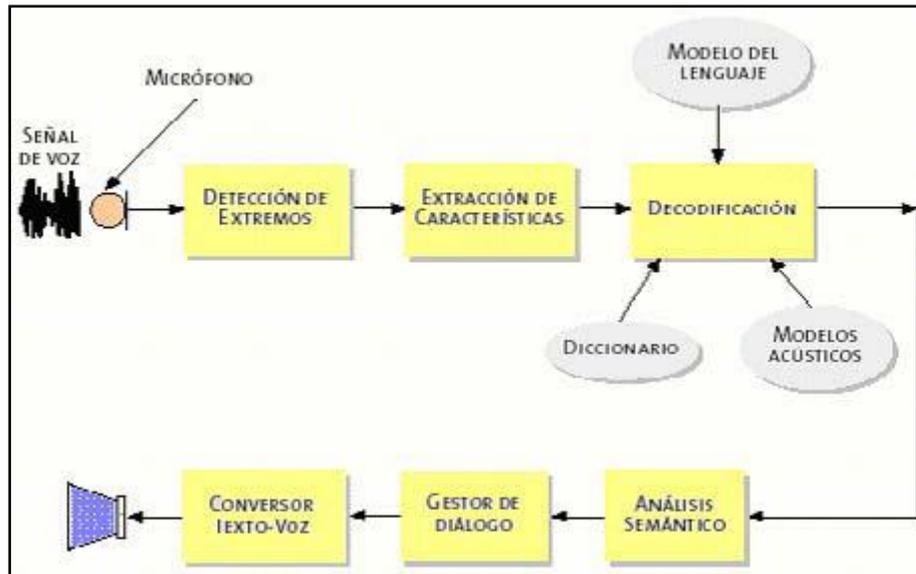
Los avances en tecnologías de reconocimiento del habla son los que han permitido dar un giro radical en la forma de automatizar las llamadas. Los tradicionales diálogos basados en árboles jerárquicos de hasta 10 opciones (las 10 teclas de teléfono), han dado paso a unos diálogos que se quieren aproximar al lenguaje natural<sup>8</sup>.

El reconocimiento de voz consiste en convertir un flujo de palabras del lenguaje a texto. De acuerdo con las características y funciones de los reconocedores, estos pueden clasificarse como:

- Reconocedores de propósito específico: Son aquellos cuyo vocabulario está restringido por un dominio, es decir, un conjunto o subconjunto determinado como letras, números, vocales, entre otros.
- Reconocedores de propósito general: Son aquellos cuyo dominio es general, como un idioma en particular cuyas palabras no caen en un conjunto determinado.

---

<sup>8</sup> Libro Portal de Voz del Ciudadano  
Marçia José Mulas, Jorge Navas Elorza



**Gráfico II.11: Arquitectura de Reconocimiento de Voz**

#### **2.4.2.1 CARACTERÍSTICAS PRINCIPALES**

- Permiten reconocer elementos de grandes listas (del orden de 10 000 entradas). Y por tanto hacer preguntas como ¿Desde qué localidad nos llama? ¿Con qué persona desea hablar? ¿Qué valor de bolsa desea consultar? ¿A qué restaurante desea ir?
- Permiten reconocer uno o varios elementos dentro de una frase, siempre que la respuesta pertenezca a un contexto restringido. Por ejemplo, en una aplicación bancaria, el diálogo podría contener preguntas del tipo “¿Qué operación desea realizar?” Y reconocer en respuestas como “Quiero conocer el saldo de mi cuenta de ahorro” los elemento clave “saldos” y “cuenta de ahorro”.
- Permiten reconocer elementos de listas generadas dinámicamente en función de los datos de la llamada.

#### **2.4.2.2 LIMITANTES**

La limitación básica de estas tecnologías reside en que exigen una inversión de tiempo importante en el diseño y ajuste de los diálogos y de los contextos de reconocimiento. En efecto, el reconocimiento está estrechamente ligado al diálogo, a la forma de preguntar las cosas. Para cada pregunta hay que definir el posible conjunto de respuestas esperado (denominado gramática), y validarlo, una vez desarrollada la aplicación, en base a la escucha de cientos de respuestas reales de los llamantes. Es

una fase en el desarrollo de los portales de voz que se ha denominado fase de afinado y que tiene como objetivo mejorar conjuntamente diálogo y gramáticas de reconocimiento.

El tiempo de afinado depende de los distintos criterios. Si el diálogo es muy dirigido, por ejemplo: “Diga uno a uno los dígitos de su código postal” o “Diga únicamente el nombre de la provincia desde donde está llamando”, el tiempo de afinado es menor que en el caso de un diálogo muy abierto, por ejemplo: “¿Qué desea?”.

Si los usuarios potenciales del sistema son usuarios avanzados, el tiempo de afinado se reduce. Los usuarios avanzados saben cómo hablar con una máquina, articulan bien, responden de forma breve, dejan tiempo para que la máquina responda, escuchan bien las preguntas para saber cómo responder. Si el sistema se dirige a un abanico amplio de usuarios distintos el tiempo de afinado será mayor. En estos casos es mejor optar por diálogos más guiados que delimiten bien los contextos de reconocimiento.

#### 2.4.2.3 ARQUITECTURA DE UN SISTEMA DE RECONOCIMIENTO DE VOZ



**Gráfico II.12: Componentes del Reconocimiento de Voz**

Un sistema de reconocimiento de voz consta de los siguientes componentes:

- **Extractor de características:** Una vez que recibe la señal, el extractor de características divide la señal en una colección de segmentos, aplicándole alguna técnica de procesamiento de señales para obtener la representación de las características acústicas más distintivas de segmento. Con las características

obtenidas, se construye un conjunto de vectores que constituyen la entrada al siguiente módulo.

- Clasificador probabilístico: Se crea un modelo probabilístico basado en redes neuronales, como sería el caso de los Modelos Ocultos de Markov (HMM). Una vez que se obtuvieron las probabilidades, es realizada una búsqueda para obtener la secuencia de segmentos con mayor probabilidad de ser reconocidos.

#### **2.4.2.4 SISTEMAS DEPENDIENTES DEL LOCUTOR**

Un sistema de reconocimiento natural de voz dependiente del locutor requiere de un entrenamiento previo, los usuarios deben almacenar en una base de datos las grabaciones de sus voces pronunciando las palabras que luego el sistema deberá reconocer. Cuando se llama al motor de reconocimiento para que detecte que palabra ha pronunciado un usuario, se compara ésta con las señales acústicas almacenadas en su base de datos correspondientes a las palabras o frases programadas y la respuesta se emite en base al nivel de semejanza. Existen serias limitaciones con este tipo de motores de reconocimiento, el estado de ánimo de las personas puede cambiar en determinadas situaciones y esto se reflejará en el tono de voz o la velocidad con que se pronuncia las palabras, el sistema no detectará que se haya pronunciado una palabra que está en su base de datos si esta tiene distinto tono, velocidad o volumen aunque sea pronunciada correctamente.

#### **2.4.2.5 SISTEMAS INDEPENDIENTES DEL LOCUTOR**

Los sistemas de reconocimiento natural de voz independiente del locutor pueden detectar lo que cualquier usuario pronuncie, no necesitan un entrenamiento previo y están destinados a aplicaciones específicas de sistemas de respuesta de voz interactiva en los que no se hace diferencia entre usuarios de distinta edad procedencia o sexo. Con estos sistemas se pueden crear básicamente dos tipos de aplicaciones, las de lenguaje natural y diálogo guiado. En las primeras se pueden formular preguntas abiertas al usuario como por ejemplo ¿Qué desea hacer?, las posibles respuestas dependen del servicio, en una empresa que comercialice autos nuevos podría obtenerse: “Comprar un auto”, “Conocer el estado del crédito de mi auto”, “Saber de nuevos modelos”, “Hablar con una operadora” e incluso “Necesito ayuda”. En las aplicaciones de diálogo guiado se lleva al usuario por menús de opciones

determinadas, en el ejemplo anterior se podría tener: ¿Por favor diga que necesita hacer, conocer sobre las últimas promociones, revisar el estado de su crédito o hablar con una operadora?

En cualquiera de los dos casos la aplicación debe ser lo suficientemente capaz de diferenciar las respuestas y tomar la acción debida, generalmente se analiza palabras claves y no exactamente las respuestas completas. Un sistema de reconocimiento de voz independiente del locutor conlleva más complejidad que uno que dependa de él, el proceso de reconocimiento ya no se basa simplemente en comparar las señales acústicas de la voz sino que debe establecer modelos acústicos promedio de los usuarios, pues la señal de una misma palabra pronunciada por un niño, un hombre adulto o una mujer seguramente tendrá un patrón distinto.

Existe también una clasificación de los sistemas de reconocimiento natural de voz de acuerdo a la cantidad de palabras que pueden reconocer, se mencionan los de vocabularios pequeños, medianos, y grandes o ilimitados. En la práctica son los proveedores de software quienes ofrecen módulos especializados para determinadas aplicaciones, por ejemplo, un motor que reconozca números pronunciados en un determinado idioma o palabras que se usan en la banca, etc.

### **2.4.3 SISTEMAS TEXT TO SPEECH <sup>9</sup>**

La síntesis de voz es la producción artificial del habla humana. Se han diseñado diferentes sistemas para este propósito llamados sintetizadores de voz y pueden ser implementados tanto en hardware como en software. Un sistema TTS (text to speech) o sintetizador de voz convierte el lenguaje escrito en habla. Otros sistemas convierten el texto en representaciones simbólicas lingüísticas tales como transcripciones fonéticas. El habla sintetizada se genera concatenando segmentos de grabaciones que se encuentran almacenados en una base de datos. Los sistemas text to speech difieren en diversos aspectos, uno de ellos es el tamaño de las unidades de habla almacenados. Los sistemas que almacenen fonemas y difonemas proveen el rango de salida más amplio, sin embargo es posible que su calidad sea baja. Para una salida de alta calidad, se utiliza la técnica de dominios específicos; en esta técnica el almacenamiento de palabras u oraciones pre-grabadas enteras permiten una salida de alta calidad.

---

<sup>9</sup> Panorama de los Sistemas de Texto a Voz

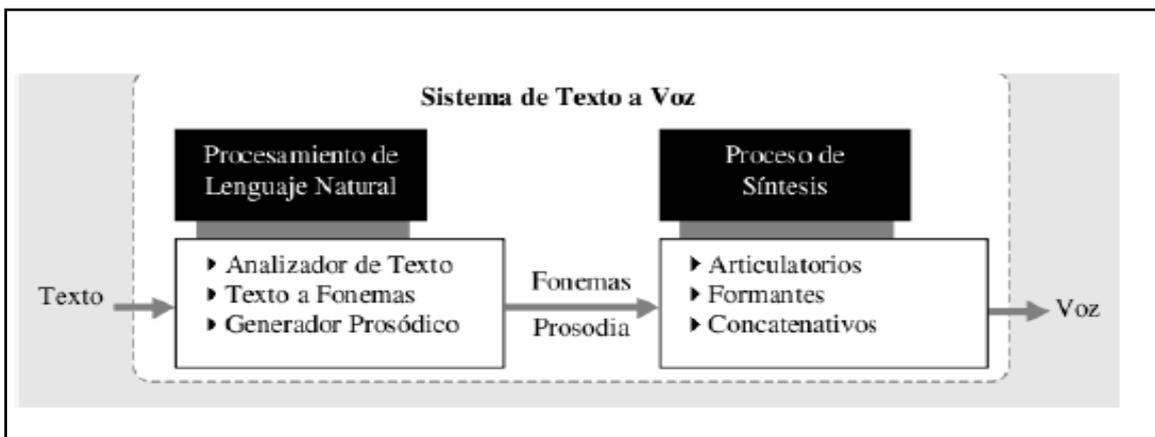
[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/moreno\\_a\\_ga/capitulo1.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/moreno_a_ga/capitulo1.pdf)

Alternativamente, un sintetizador puede incorporar un modelo del tracto vocal y otras características de la voz humana para generar una voz completamente “sintética” o “electrónica”.

Un elemento para juzgar la calidad de la síntesis de voz es su parecido con la voz humana y su potencialidad para ser entendida. Un sistema TTS puede lograr que personas con discapacidad visual, de lectura o en actividades en las cuales ojos y manos están ocupadas puedan escuchar instrucciones, textos leídos, entre otros textos. Muchos sistemas operativos han incluido sintetizadores de voz desde principios de la década de los ochenta y su calidad ha variado mucho de modelo en modelo, incluso actualmente hay TTS de muy variados tipos y calidades.

#### 2.4.3.1 FUNCIONAMIENTO

Existe una gran gama de diferentes sistemas para el procesamiento de texto con complejidad muy variada. Sin embargo, muchos de estos sistemas comparten una arquitectura general que ha demostrado ser muy útil en muchos casos. El gráfico II.13 muestra esta arquitectura común para los sistemas.



**Gráfico II.13: Estructura general de un sistema TTS**

Existen dos bloques principales que forman el sistema: el bloque de Procesamiento de Lenguaje Natural (Natural Language Processing o NLP) y el bloque de Proceso de Síntesis. El bloque de NLP se encarga de producir una transcripción fonética del texto leído, además de la entonación y el ritmo deseados para la voz de salida. Después, el

bloque de Proceso de Síntesis transforma la información simbólica que recibe del bloque anterior, en una voz de salida.

El bloque NLP tiene dos tareas importantes. La primera tarea es convertir el texto en crudo, conteniendo símbolos como números o abreviaciones, en su equivalente en palabras escritas. Esto comúnmente es llamado “normalización de texto”, “pre-procesamiento” o “señalización” (tokenization). Su segunda tarea es asignar transcripciones fonéticas a cada palabra y dividir las en unidades prosódicas tales como frases, cláusulas y oraciones. El proceso de asignar transcripciones fonéticas en palabras es llamado conversión “texto a fonema” o conversión “grafema a fonema”. El bloque NLP tiene como salida una representación lingüística, la cual está formada tanto por las transcripciones fonéticas como por la información prosódica. El bloque de Proceso de Síntesis es el sintetizador en sí, aquí es donde se transforma la representación de la lógica lingüística en sonido.

#### **2.4.3.2 UTILIDAD DE LOS SINTETIZADORES DE VOZ**

Los sintetizadores de voz se utilizan principalmente para dar accesibilidad a personas discapacitadas visualmente, así como para actividades o situaciones en las cuales no se puede usar la vista. Así, una aplicación clásica de esta tecnología son los lectores de pantalla donde personas invidentes o de visión débil pueden escuchar todos los textos presentados en pantalla.

Los sistemas TTS se emplean con frecuencia para apoyar a personas con dislexia y otras dificultades de lectura. También son aplicados para la educación de niños que todavía no saben leer, para ayudar a personas con discapacidad de habla a través de un sistema VOCA (Voice- Output Communication Aid), en sistemas de navegación (GPS), juegos, etc.

Una interfaz auditiva nos puede ayudar a la educación y al aprendizaje. Con dicha interfaz es posible no solo leer la información, sino escucharla. De esta manera tendríamos una interfaz multi modal donde el sentido del oído formaría parte esencial de la manipulación y asimilación de la información.

### 2.4.3.3 TIPOS DE SINTETIZADORES EXISTENTES

Los dos aspectos más importantes de la síntesis de voz son la naturalidad y la inteligibilidad. La naturalidad describe qué tanto el sonido generado se asemeja al habla humana, mientras que la inteligibilidad es la facilidad con la cual se entiende el significado del habla generada. Un sintetizador de voz trata de maximizar ambas características. Las dos tecnologías principales para generar una voz sintética son la síntesis concatenativa y la síntesis formante. La síntesis formante no utiliza fragmentos de habla humana al momento de generar una voz. En vez de esto utiliza un modelo acústico. Parámetros como frecuencia fundamental, fonación y niveles de ruido se analizan para crear una onda acústica de habla artificial.

La síntesis concatenativa se basa en la unión de segmentos de habla grabados. Generalmente la síntesis concatenativa produce el sonido más natural en la síntesis de una voz. Sin embargo, las diferencias entre las variaciones naturales de la voz y la naturaleza de las tecnologías para automatizar la fragmentación de las ondas sonoras resultan en una salida imperfecta. Los tres tipos más comunes de síntesis concatenativa son:

- Selección de Unidades: Fonemas, sílabas, morfemas, palabras, frases y oraciones son las unidades en las que se puede dividir un sintetizador de selección de unidades. Este tipo de síntesis utiliza una gran base de datos de habla grabada. Durante la creación de la base de datos, cada grabación es seccionada en una o más de las unidades antes mencionadas. La selección de unidades provee gran naturalidad debido a que aplica muy poco procesamiento digital de señales (DSP) a las grabaciones. El procesamiento digital de señales generalmente hace que la voz suene menos natural, aunque algunos sintetizadores utilizan el procesamiento de señales marginalmente en el punto donde se concatenan las grabaciones para suavizar la onda acústica.
- Difonemas: La síntesis de difonemas utiliza una base de datos mínima, la cual contiene todas las transiciones de sonido a sonido del lenguaje que se desee sintetizar. El número de difonemas depende de las fonotácticas del lenguaje. Por

ejemplo, el español tiene aproximadamente 800 difonemas mientras que el alemán tiene 2500. En la síntesis por difonemas solamente se almacena en la base de datos un ejemplo de cada difonema. Al momento de ejecución, la información prosódica de una oración es impuesta sobre estas unidades mínimas por medio de técnicas de procesamiento digital de señales tales como LPC (Linear Predictive Coding), PSOLA o MBROLA.

- Dominio Específico: Estos sintetizadores concatenan palabras o frases pre-grabadas para generar nuevas expresiones. Se utiliza en aplicaciones donde la variedad de textos de salida del sintetizador se limita a un dominio en particular, como un reloj o una calculadora. El nivel de 11 naturalidades es muy elevado debido a que el número de oraciones almacenadas es muy reducido y se asemejan mucho a la entonación y pronunciación de las grabaciones originales.

## 2.5 EL ESTÁNDAR VOICEXML

### 2.5.1 EL WORLD WIDE WEB Y VOICEXML

La inmensa cantidad de posibilidades que ofrece el mundo Web ha ido creciendo vertiginosamente desde sus comienzos. Al principio, los servidores Web, solo almacenaban contenidos estáticos. Ahora, son capaces de generarlos dinámicamente mediante scripts, paginas servidor, servlets, acceso a bases de datos y muchas otras tecnologías. VoiceXML aprovecha todas estas tecnologías ya desarrolladas. También la representación de la información Web ha avanzado con el tiempo. VXML es una forma general y flexible de representar la información, que además permite de forma sencilla y mediante tecnologías de transformación, convertir una estructura VXML en otra VXML o en otra que no siga este formato. VoiceXML se basa en el XML beneficiándose así, de todas las ventajas que ofrece este lenguaje.

Como también se beneficia de los avances continuos de Internet, en términos de eficiencia, anchos de banda y calidad de servicio. Mejoras que conducen a nuevas aplicaciones Web y servicios, los cuales generan en si nuevas mejoras. Por ejemplo, la posibilidad de mover información de audio a través de la Web de forma eficiente es un claro beneficio para VoiceXML.

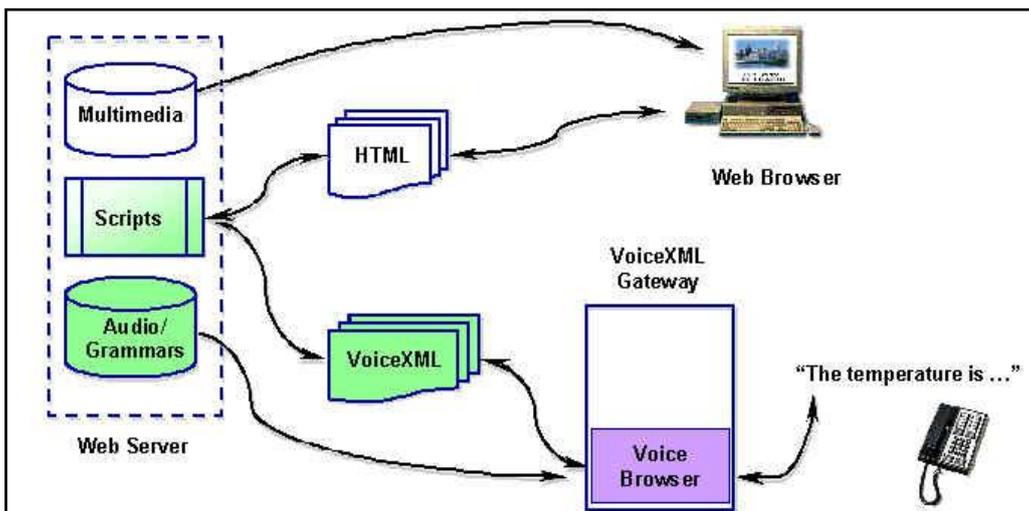


Gráfico II.14: VoiceXML vs Web

Para trabajar en todo éste mundo Web, han ido apareciendo aplicaciones y herramientas, que con el tiempo han ido evolucionando y convirtiéndose en más potentes. Algunas de ellas, como por ejemplo los editores sensitivos a XML, pueden usarse para desarrollar aplicaciones en VoiceXML de forma sencilla y útil.

Internet se extiende a más dispositivos que los ordenadores personales: por ejemplo, los dispositivos PDA con conexión wireless, los teléfonos móviles que soportan tecnologías como WAP (Wireless Application Protocol) o GPRS (General Packet Radio Service) o los teléfonos i-mode con tecnología NTT Docomo's (tecnología adoptada por los japoneses).

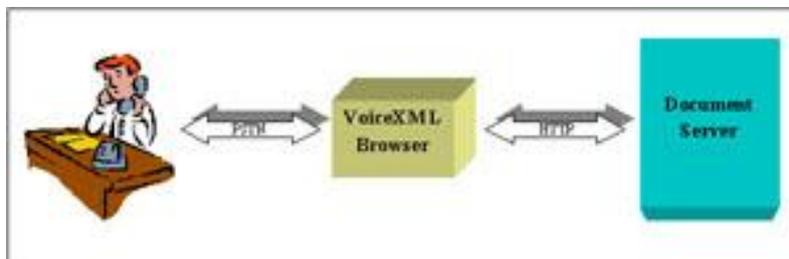
El futuro nos llevará más dispositivos preparados para el Web: MP3 de mano con conexión a Internet, máquinas de venta de productos que ellas mismas hacen encargos cuando se les acaba algún producto, displays de pared que descargan cuadros de Internet, televisores basados en el Web, etc. La tecnología de voz es una interfaz natural y muy potente para los dispositivos Web. Los micrófonos son mucho más pequeños que los teclados y los altavoces que las pantallas. Parece bastante probable que la mayoría de los futuros dispositivos Web incorporaran un sistema de reconocimiento y síntesis de voz.

	<b>WEB</b>	<b>VOICEXML</b>
<b><i>Los documentos definen</i></b>	Interface visual, paginas de texto y graficas.	Conversación que el usuario tiene con la aplicación.
<b><i>Entradas y navegación</i></b>	Teclado y Mouse	Habla
<b><i>Ejecución</i></b>	Web Browser (Navegadores de Red)	Voice Browser. Portales de voz a través de un VoiceXML Interpreter.
<b><i>Acceso</i></b>	Computador	Teléfono, móvil, VozIP

**Tabla II.1: Voicexml vs Web**

### 2.5.2 CARACTERÍSTICAS GENERALES<sup>10</sup>

Antes de entrar en discutir VoiceXML tenemos que entender como funciona VoiceXML, para ello se debe revisar el gráfico II.15.



**Gráfico II.15: Características de VoiceXML**

Del gráfico II.15 podemos obtener los siguientes enunciados.

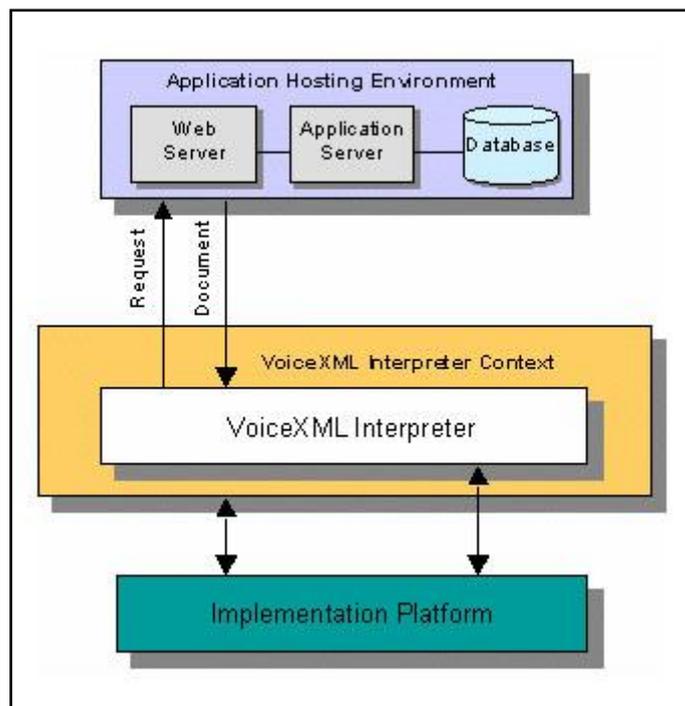
- Mientras que HTML permite crear interfaces "gráficas" para que el usuario ingrese y reciba información, VoiceXML genera interfaces orales. Es decir, el usuario no "ve" la información, la escucha.
- Como VoiceXML no genera interfaces gráficas, el usuario no necesita una computadora (ni una PDA) solo le basta con un teléfono.
- El usuario se conecta al Browser a través la Red Publica de Telefonía (PSTN)
- VoiceXML, al igual que HTML, necesita de un browser para procesar la información, cada browser debe ser capaz de reconocer y procesar las etiquetas de cada lenguaje.
- VoiceXML, al igual que HTML, necesita de un browser para procesar la información, cada browser debe ser capaz de reconocer y procesar las etiquetas de cada lenguaje.

### 2.5.3 MODELO ARQUITECTÓNICO

VoiceXML se basa en una arquitectura que integra los siguientes componentes que se presentan, los cuales se detallan a continuación. Ver Gráfico II.16.

<sup>10</sup> LA NUEVA ERA DE LA VOZ (Parte 2)

[http://www.informatizate.net/articulos/la\\_nueva\\_era\\_de\\_la\\_voz\\_parte\\_02\\_12052004.html](http://www.informatizate.net/articulos/la_nueva_era_de_la_voz_parte_02_12052004.html)



**Gráfico II.16: Arquitectura VoiceXML**

### 2.5.3.1 APPLICATION HOSTING ENVIROMENT

Llamado también "Document Server". Es un ambiente que genera dinámicamente documentos VoiceXML. Básicamente esta compuesto por 3 componentes:

- **Web Server:** Servidor Web que recibe HTTP Request y envía HTTP Response con un documento VoiceXML.
- **Application Server:** Servidor de aplicaciones que mantiene una lógica de negocio que sobre la base de los parámetros enviados por el Web Server genera documentos VoiceXML.
- **Database:** Base de Datos de la cual se obtiene información para generar los documentos VoiceXML

### 2.5.3.2 VOICEXML INTERPRETER

Aplicación que recibe un documento VoiceXML y lo interpreta, es decir procesa las etiquetas que dicho documento contiene.

### **2.5.3.3 VOICEXML INTERPRETER CONTEXT**

Modulo del VoiceXML Interpreter que monitorea las posibles actividades que los usuarios realizan mientras se está interpretando un documento VoiceXML, por ejemplo el usuario podría presionar desconectarse (colgar el teléfono), lo generaría que cancelación de la interpretación del documento.

### **2.5.3.4 IMPLEMENTATION PLATAFORM**

Este componente viene a ser el Browser en sí, pues cada empresa puede desarrollar su propio VoiceXML Browser el cual aparte de interpretar un documento VoiceXML puede implementar mecanismos de cache, procesamiento de llamadas telefónicas, etc.

### **2.5.4 METAS DE VOICEXML**

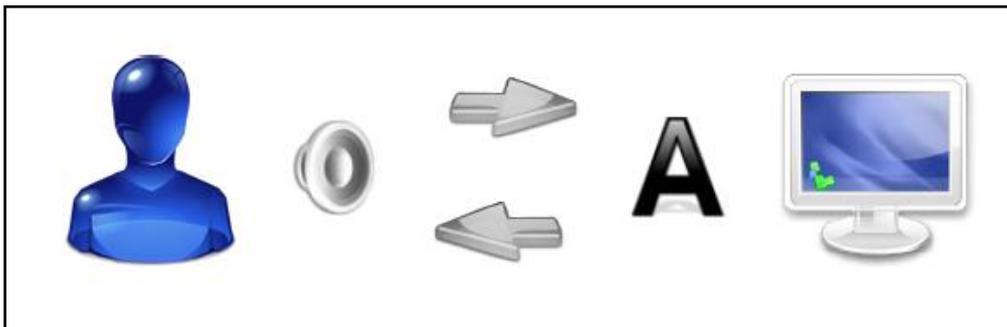
La meta principal de VoiceXML consiste en traer de la Web y entregar su contenido a aplicaciones de respuesta por voz, y liberar a los autores de tales aplicaciones de la administración de los recursos de la programación de bajo nivel. Además, permite la integración de los servicios de voz con servicios de datos usando el paradigma familiar de cliente-servidor. Un servicio de voz se ve como una secuencia de interacción de diálogos entre un usuario y la implementación de una plataforma. VoiceXML es un lenguaje que reduce al mínimo las interacciones cliente-servidor especificando interacciones múltiples por documento. Separa código de la interacción del usuario (en VoiceXML) de la lógica del servicio (ej. escrituras del cgi). Promueve portabilidad del servicio a través de plataformas puestas en prácticas. Es fácil de utilizar para las interacciones simples, pero proporciona características de lenguaje para apoyar diálogos complejos. Mientras que VoiceXML se esfuerza en acomodar los requisitos de una mayoría de servicios de respuesta por voz, los servicios con requisitos rigurosos también se pueden implementar realizando un nivel de control más fino.

### **2.5.5 ALCANCE DE VOICEXML**

El lenguaje describe la interacción humano-maquina proporcionada por los sistemas de respuesta por voz, que incluye

- Salida de voz sintetizado (text-to-speech).

- Salida de archivos audio.
- Reconocimiento de voz.
- Reconocimiento de entrada DTMF.
- Grabación de entrada de voz.
- Control del flujo de diálogo.
- Características de la telefonía tales como transferencia y desconexión de la llamada.



**Gráfico II.17: Interacción Hombre-Maquina**

El lenguaje provee los medios para recoger tonos DTMF y/o entrada de voz, asignando los resultados de la entrada a las variables del documento, definidas en la solicitud, el cual toma las decisiones que afectan la interpretación de los documentos escritos en el lenguaje. Un documento se puede enlazar a otros documentos a través de los identificadores uniformes de recursos (URIs).

#### **2.5.6 PRINCIPIOS DE DISEÑO DE VOICEXML.**

El lenguaje promueve la portabilidad de servicios con la abstracción de los recursos de la plataforma. El lenguaje acomoda diversidad de plataformas en formatos de audio, formatos de gramática y esquemas de URI. Mientras que los fabricantes de plataformas pueden apoyarse a través de varios formatos gramaticales. El lenguaje requiere un formato común gramatical llamado XML sobre Especificaciones gramaticales de reconocimiento de voz (Speech Recognition Grammar Specification – SRGS) para facilitar la interoperabilidad. Similarmente, varios formatos de audio para lectura y la grabación pueden ser soportados.

El lenguaje tiene una semántica bien definida. No se requiere La heurística del cliente para determinar la interpretación del elemento del documento. El lenguaje reconoce

interpretaciones semánticas de gramáticas y pone esta información a disposición de la aplicación. El lenguaje tiene un mecanismo de control de flujo. No es pensado para uso intensivo de cómputo, operaciones con bases de datos, u operaciones de sistema. Éstos se asumen que serán manejados por los recursos fuera del intérprete del documento, ej. Un servidor del documento. La lógica general del servicio, la administración del estado, la generación de diálogo, y la secuencia del diálogo se asume que residen fuera del intérprete del documento. El lenguaje proporciona maneras de enlazar documentos usando URIs, y también de enviar datos a los scripts del servidor utilizando URIs. VoiceXML proporciona las maneras de identificar exactamente qué datos enviará al servidor, y que método del HTTP (GET o POST) a utilizar en el envío.

### **2.5.7 REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DE LA PLATAFORMA**

A continuación se presentan los requerimientos de hardware/software para la instalación de la plataforma de VoiceXML.

- **Adquisición de documento.** Se espera que el contexto del intérprete recupere los documentos para que el intérprete de VoiceXML actúe sobre él. El esquema de URI del "HTTP" debe ser soportado. En algunos casos, la petición del documento es generada por la interpretación de un documento de VoiceXML, mientras que otras peticiones son generadas por el contexto del intérprete, en respuesta a acontecimientos fuera del alcance del lenguaje, por ejemplo una llamada telefónica entrante. Al publicar peticiones del documento vía HTTP, el contexto del intérprete se identifica usando el "User- Agent" como variable "<name>/<version>", por ejemplo, "acme-browser/1.2".
- **Salida audio.** Una implementación de plataforma debe soportar salida de audio usando archivos audio y text-to-speech (TTS). La plataforma debe poder ordenar libremente TTS y la salida audio. Si un recurso de salida de audio no está disponible, un evento de error.noresource debe ser presentado. Los archivos de audio son referidos por un URI. El lenguaje especifica que un sistema de archivos de formatos audio es requerido.

- **Entrada audio.** Una implementación de la plataforma es requerida para detectar y reportar el carácter y/o la entrada de voz simultáneamente, y para controlar la duración del intervalo de la detección de la entrada con un contador de tiempo cuya longitud es especificada por un documento de VoiceXML. Si un recurso de audio de la entrada no está disponible, un acontecimiento de error.noresource debe ser presentado.
- Debe reflejar los caracteres (por ejemplo, DTMF) incorporados por un usuario. Las plataformas deben soportar la gramática de XML del DTMF descritas en la especificación del reconocimiento de voz de W3C. Deben también apoyar la forma aumentada de BNF (ABNF) de gramáticas de DTMF descritas en la especificación de la gramática del reconocimiento de voz de W3C.
- Debe ser capaz de recibir datos dinámicos de reconocimiento de voz. Debe poder utilizar datos de gramática de reconocimiento de voz en la forma de XML. Debe poder recibir datos de la gramática de reconocimiento de voz en la forma de ABNF, y debe soportar otros formatos tales como el formato de la gramática de JSpeech formatos propietarios. Algunos elementos de VoiceXML contienen datos de la gramática de voz; otros refieren a datos de la gramática de voz con un URI. El reconocedor de voz debe ser capaz de acomodar dinámicamente la entrada hablada la cual está escuchando con cualquier método de especificación de datos de gramática de voz.
- Debe poder grabar el audio recibido del usuario. La implementación de la plataforma debe poder colocar la grabación a disposición de la variable a solicitud de esta. El lenguaje especifica un conjunto de requerimientos de formatos de audio los que deben ser soportados por la plataforma.
- **Transferencia.** La plataforma deberá poder hacer una conexión a terceros a través de una red de comunicaciones, tal como el teléfono.

## 2.5.8 ELEMENTOS DE LA PROGRAMACIÓN VOICEXML<sup>11</sup>

Un formato de documento en VoiceXML (o un sistema de documentos relacionados llamado una aplicación) es una forma conversacional automatizada. Es decir, el usuario está siempre en un estado conversacional, o en diálogo, al mismo tiempo con la computadora. Cada diálogo determina la transición al diálogo siguiente. Se especifican las transiciones usando URIs, que definen el documento y el diálogo siguientes a hacer utilizado. Si un URI no refiere a un documento, se asume que es el documento actual. Si no refiere a un diálogo, el primer diálogo en el documento es considerado como el primero. Se termina la ejecución cuando un diálogo no especifica un sucesor, o si tiene un elemento que salga explícitamente de la conversación.

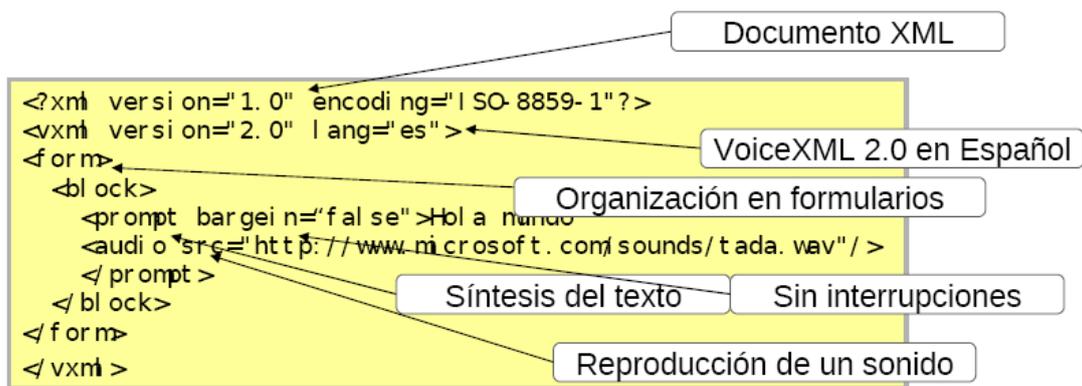


Gráfico II.18: Formato de un documento Voicexml.

### 2.5.8.1 DIALOGOS Y SUBDIALOGOS

Hay dos clases de diálogos: formas (o formularios) y menús. Los formularios definen una interacción que recoge los valores para un sistema de variables. Cada campo puede especificar una sintaxis que define las entradas permisibles para ese campo. Si una sintaxis a nivel de forma está presente, puede ser utilizada para llenar varios campos a partir de una elocución. Un menú se presenta al usuario con una selección de opciones y es entonces cuando las transiciones se dan a otro diálogo basado en dichas opciones.

<sup>11</sup> Sistema de la Información basadas en VoiceXML  
<http://www.wisis.ufg.edu.sv/www.wisis/documentos/TE/005.74-M79d/005.74-M79d-Capitulo%20II.pdf>

Un subdiálogo es como una llamada de función, en que proporciona un mecanismo para invocar una nueva interacción, y luego vuelve al formulario original. Las Variables, gramáticas, y la información del estado de las variables se guardan y están disponibles para el documento que hizo la llamada. Los Subdiálogos se pueden utilizar, por ejemplo, para crear una secuencia de confirmación que pueda requerir una consulta a base de datos; para crear un sistema de componentes que se pueden compartir entre documentos en una sola aplicación; o para crear una biblioteca reutilizable de diálogos compartidos entre muchas aplicaciones.

### 2.5.8.2 SESIONES

Una sesión inicia cuando el usuario comienza a interactuar recíprocamente con un contexto del intérprete de VoiceXML. Continúa hasta que se cargan y se procesan los documentos, y finaliza cuando lo es requerido por el usuario, un documento, o el contexto del intérprete.

### 2.5.8.3 APLICACIONES

Una aplicación es un conjunto de documentos que comparten el mismo documento raíz. Siempre que el usuario interactúe recíprocamente en una aplicación, su documento raíz también se carga. El resto de los documentos es cargado mientras el usuario navega entre documentos de la aplicación y éste se descarga cuando la navegación del usuario se hace a un documento que no esté en uso. Mientras que es cargado, las variables de la aplicación del documento raíz están disponibles para los otros documentos.

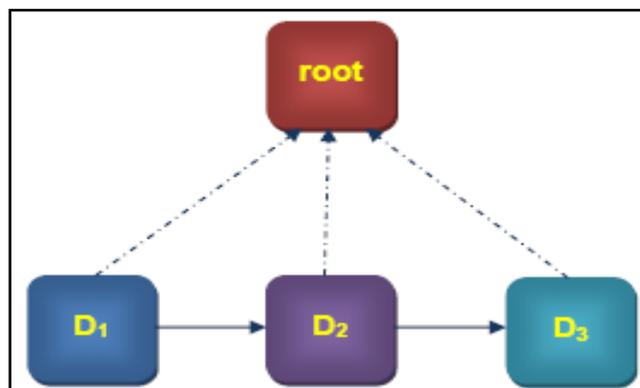


Gráfico II.19: Transición entre documentos VoiceXML

## 2.5.8.4 GRAMÁTICAS<sup>12</sup>

Las gramáticas son los elementos clave de las entradas al sistema por parte del usuario. Cada diálogo en VoiceXML tiene una o más gramáticas de voz y/o DTMF asociadas.

### 2.5.8.4.1 GRAMÁTICAS DE VOZ

Mediante las gramáticas de voz se pueden especificar:

- Un conjunto o *set* de palabras o frases que el usuario puede decir para llevar a cabo una acción o proporcionar una información.
- Además se puede proporcionar un valor de retorno para describir la información o la acción solicitada.

La representación de gramáticas para utilizar en reconocimiento de voz se ha estandarizado con dos sintaxis distintas: la conocida como **ABNF** (Augmented Backus-Naur Format) y la **XML**.

Ambas son reconocidas por el W3C como estándares (<http://www.w3.org/TR/speech-grammar/>) del reconocimiento de voz. El motor de reconocimiento utilizado por Verbio VoiceXML (Verbio ASR), soporta ambas sintaxis.

- Cuando se utilizan gramáticas XML debe especificarse *application/srgs+xml* en el atributo *type* de la etiqueta `<grammar>`.
- Cuando se utilizan gramáticas ABNF no es necesario especificar el atributo *type*, pero en caso de hacerlo este debe ser: *application/srgs*.

```
#ABNF 1.0 ISO8859-1;
public $polite = $startPolite $drink $endPolite;
public $startPolite = ( desearia | podria ) ;
public $drink = ( agua | vino | refresco );
public $endPolite = ( porfavor | gracias ) ;
```

#### Espacio Código II.1: Ejemplo de una gramática ABNF

---

<sup>12</sup>Guía de VoiceXML

[http://www.verbio.com/webverbio3/html/reference/pdf/guide\\_voicexml\\_es.pdf](http://www.verbio.com/webverbio3/html/reference/pdf/guide_voicexml_es.pdf)

Equivalente en el formato XML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<grammar xmlns="http://www.w3.org/2001/06/grammar"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/06/grammar
http://www.w3.org/TR/speech-grammar/grammar.xsd" xml:lang="es"
version="1.0" root="polite" mode="voice">
<rule id="polite" scope="public">
<ruleref uri="#startPolite"/>
<ruleref uri="#drink"/>
<ruleref uri="#endPolite"/>
</rule>
<rule id="startPolite" scope="public">
<one-of>
<item tag="desearia">desearia</item>
<item tag="podria">podria</item>
</one-of>
</rule>
<rule id="drink" scope="public">
<one-of>
<item tag="agua">agua</item>
<item tag="vino">vino</item>
<item tag="refresco">refresco</item>
</one-of>
</rule>
<rule id="endPolite" scope="public">
<one-of>
<item tag="porfavor">por favor</item>
<item tag="gracias">gracias</item>
</one-of>
</rule>
</gramar>
```

### Espacio Código II.2: Ejemplo gramatica.xml

Existen plantillas de transformación u hojas de estilo (XSLT) para convertir gramáticas del formato XML al formato ABNF (no a la inversa).

#### 2.5.8.4.2 GRAMÁTICAS DTMF

Las gramáticas DTMF (Dual Tone Multi-Frequency) permiten definir:

- Un conjunto de números DTMF que el usuario puede pulsar para llevar a cabo una acción o proporcionar una información.
- También permiten proporcionar un valor de retorno para describir la información o la acción solicitada.

El estándar 2.0 de VoiceXML distingue las gramáticas DTMF de las gramáticas de voz mediante el atributo “*mode*” de la etiqueta <grammar>.

El atributo “*xml:lang*” no posee ningún efecto sobre las gramáticas DTMF. Por lo demás, las gramáticas de voz y las DTMF son tratadas por igual incluyendo la opción de definir gramáticas “*inline*” (incrustadas en el documento) o referenciarlas a un documento externo.

```
<grammar mode="dtmf"> 1234{Marc} | "1 2 3"{Javier} | #{Pedro} </grammar>
```

#### Espacio Código II.3: Gramática Inline

Como se muestra en el ejemplo de gramática “*inline*” DTMF dónde se aceptan como valores posibles:

- 1234 -> la gramática devuelve el nombre Marc
- 123->la gramática devuelve el nombre Javier
- #->la gramática devuelve el nombre Pedro

#### 2.5.8.4.3 GRAMÁTICAS BUILTIN

Existen gramáticas especialmente diseñadas para las tareas más comunes (y a menudo difíciles) que han sido integradas en el reconocedor como un recurso incorporado. Las gramáticas básicas incorporadas no son solamente la definición de las reglas sino el procesado interno de los resultados considerando medidas de confianza y opcionalmente múltiples hipótesis. Son las denominadas gramáticas builtin.

La versión 2.0 del estándar de VoiceXML especifica que deben proporcionarse las siguientes builtin, todas ellas soportadas por la plataforma Verbio ASR:

- **boolean:** Verbio ASR incorpora en esta builtin respuestas a preguntas tipo sí/no, cubriendo las palabras “sí”, “no” (con modelado acústico específico), y las palabras y expresiones más comunes para expresar confirmación y negación (ej: “exacto”, “correcto”, “no gracias”, etc.).
- **date:** Verbio ASR incorpora en esta builtin fechas en lenguaje natural, cubriendo fechas fijas en cualquier formato (sólo el día, día y mes o día, mes y año).
- **digits:** Verbio ASR incorpora en esta builtin cadenas de dígitos de longitud limitada o no, usando modelos acústicos específicos para los dígitos conectados (ej: “uno, tres, cuatro”, “siete”, etc.).
- **currency:** Verbio ASR incorpora en esta builtin cantidades monetarias (ej: “veintisiete euros con cinco céntimos”).
- **number:** Verbio ASR incorpora en esta builtin números naturales dentro de un rango limitado (ej: “dos mil trescientos cuarenta y siete”, “doce”, etc.).
- **phone:** Verbio ASR incorpora en esta builtin números de teléfono en lenguaje natural, cubriendo las más frecuentes formas de agrupar los dígitos en números naturales (“noventa y tres, cuatro cero nueve, setenta y uno, veinte”).
- **time:** Verbio ASR incorpora en esta builtin expresiones de tiempo en lenguaje natural que especifican horas y minutos.

Todas estas builtin deben especificarse o bien mediante el atributo type de la etiqueta <field> o bien mediante la sintaxis:

```
<grammar src="builtin:grammar/boolean"/>
```

#### Espacio Código II.4: Sintaxis Gramática Builtin

Además Verbio VoiceXML proporciona mediante Verbio ASR otras gramáticas builtin de gran utilidad para muchas aplicaciones de voz:

- **creditcard:** Verbio ASR incorpora en esta builtin números de tarjetas de crédito (dictados dígito a dígito).
- **code:** Verbio ASR incorpora en esta builtin: códigos numéricos hablados como una secuencia de números naturales.

- **nif**: Verbio ASR incorpora en esta builtin número de identificación fiscal personal (NIF) de España.
- **spell**: Verbio ASR incorpora en esta builtin deletreos, tanto alfabéticos como numéricos.

Siguiendo la recomendación 2.0 del estándar, estas builtin pueden usarse utilizando también el atributo *type* de la etiqueta <field> o bien mediante la etiqueta <grammar>, pero siempre precediendo el nombre de la builtin con una “x-“ para indicar que es una gramática builtin específica de plataforma:

```
<field name="asknif " type="x -nif"/>  
o  
<grammar src="builtin:grammar/x-nif"/>
```

**Espacio Código II.5: Ejemplos de Gramática Builtin**

#### 2.5.8.4.4 ESTRUCTURA DE ARCHIVOS DE GRAMÁTICA

Un archivo de gramática escrito con la recomendación SRGS debe tener un encabezado bien definido, por lo menos con los elementos que se indican en el espacio de código siguiente. Ver Espacio de Código II.6.

```
<?xml versión="1.0"?>  
<grammar versión ="1.0"  
  xml:lang="es"  
  mode="voice"  
  root="regla_raiz"  
  tag-format="semantics/1.0"
```

**Espacio Código II.6: Estructura de un Archivo de Gramática**

- La primera línea es la declaración del archivo XML, su atributo “version” determina la versión del lenguaje XML que se utiliza en el documento, es obligatorio empezar el archivo con esta línea. Lo que viene luego es el prólogo, se define por el

elemento <grammar> y su primer atributo obligatorio es también “version”, que corresponde con la versión de la recomendación SRGS con la que se está trabajando.

- El atributo <xml:lang> define el lenguaje utilizado en el reconocimiento de todo el archivo, este puede ser redefinido en cada elemento <item>, <token>, o <ruleref>.
- Un archivo de gramática puede estar orientado a reconocer voz o tonos DTMF (Dual Tone Multi Frequency) especificando “voice” o “dtmf” respectivamente en el atributo “mode”.
- Debe existir una regla principal (raíz) en la cual inicia el reconocimiento mediante la gramática especificada en el archivo, esto se lo hace mediante el atributo “root”.

### 2.5.9 ESTRUCTURA DE UN DOCUMENTO VOICEXML

Un documento VoiceXML debe tener una estructura bien definida, un encabezado, los diálogos y menús de la aplicación y de ser el caso, las referencias o transiciones a otros documentos. La primera línea necesaria en todo documento XML define la versión del lenguaje que estamos utilizando, además de la codificación de caracteres.

Los atributos del <vxml> incluidos:

<b>version</b>	La versión de VoiceXML de este documento (requerido).
<b>Xmlns</b>	El namespace designado para VoiceXML (requerido). El namespace para VoiceXML se define por <a href="http://www.w3.org/2001/vxml">http://www.w3.org/2001/vxml</a> .
<b>xml:base</b>	La base URI para este documento esta definido según en [ <a href="#">Xml-base</a> ]. Como en [ <a href="#">HTML</a> ]
<b>xml:lang</b>	La base URI para este documento esta definido según en [ <a href="#">Xml-base</a> ]. Como en [ <a href="#">HTML</a> ]
<b>Aplicación</b>	El URI de la aplicación raíz del documento, si lo hay.

**Tabla II.2: atributos del < vxml >**

El elemento <vxml> define los parámetros necesarios del documento VoiceXML, su primer atributo “xmlns” es obligatorio y corresponde al espacio de nombres designado para este tipo de documentos. Generalmente un intérprete VoiceXML necesita de un servidor Web de documentos, con el atributo “xml:base” se especifica este servidor, para que todas las referencias a archivos se basen en esa dirección. El atributo “xml:lang” es opcional y define el lenguaje que se utilizará para la aplicación, por ejemplo se capturará el evento <help> cuando el usuario pronuncie “ayuda” si el lenguaje está definido como “es”. La versión del lenguaje VoiceXML que se está utilizando en el documento se especifica con el atributo “version”. Los elementos <meta> son opcionales y sirven para definir ciertos parámetros del documento como el autor, los derechos de autor, una dirección de correo electrónico de soporte, etc.

```
<?xml version="1.0" encoding="iso-8859-1"?>  
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
```

**Gráfico II.20: Ejemplo de <vxml>**

### 2.5.9.1 ELEMENTOS DEL VOICEXML

En la siguiente tabla se describen los principales elementos que se utilizan en el desarrollo de aplicaciones VoiceXML.

ELEMENTO	PROPÓSITO
<assign>	Asigna un valor a una variable
<audio>	Ejecuta un clip de audio
<block>	Un contenedor de código ejecutable (no interactivo).
<catch>	Captura un evento.
<choice>	Define un elemento de menú.
<clear>	Limpia una o más variables en el formulario.
<disconnect>	Desconecta una sesión.
<else>	Usado dentro de una sentencia <if>.
<elseif>	Usado dentro de la sentencia <if>.
<enumerate>	Término para enumerar la selección de opciones en un menú.
<error>	Captura un error de evento
<exit>	Salir de una sesión.
<field>	Declara una entrada de campo en un formulario.
<filled>	Una acción es ejecutada cuando los campos son llenados.

<form>	Un dialogo para presentar información y recoger datos.
<goto>	Ir a otro diálogo en el mismo o en otro documento.
<grammar>	Especifica un reconocimiento de voz o gramática DTMF.
<help>	Captura un evento de ayuda.
<if>	Condicional simple.
<initial>	Declara inicializaciones lógicas sobre entradas en un formulario.
<link>	Especifica transiciones comunes para todos los diálogos en el alcance del link.
<log>	Genera información para depuración de errores.
<menu>	Un diálogo para elegir entre destinos alternativos.
<meta>	Define elementos del metadata como el par nombre/valor.
<metadata>	Define información del metadata utilizando un esquema de metadatos.
<noinput>	Captura un evento de no entrada.
<nomatch>	Captura un evento de no enlace.
<object>	Interactúa con una extensión personalizada.
<option>	Especifica una opción en un campo <field>
<param>	Parámetro en <object> o <subdialog>
<prompt>	Cola de síntesis de voz y salida de audio para el usuario.
<property>	Configuración para el control de la implementación de la plataforma.
<record>	Graba un audio básico.
<reprompt>	Realiza la ejecución de un prompt, cuando un campo se visita nuevamente después de un acontecimiento.
<return>	Retorna de un subdiálogo.
<script>	Especifica un bloque de ECMAScript del lado del cliente.
<subdialog>	Invoca otro diálogo como también un subdiálogo en la sesión activa.
<submit>	Envía valores al servidor.
<throw>	Lanza un evento.
<transfer>	Trasfiere las llamadas a otro destino.
<value>	Inserta los valores de una expresión en un prompt.
<var>	Declara una variable.
<vxml>	Elemento ubicado en el nivel superior en cada documento de tipo VoiceXML.

**Tabla II.3: Etiquetas de VoiceXML**

A continuación se detallan los principales elementos de la programación en VoiceXML.

- **PROMPT**

El elemento **prompt** (en lenguaje VoiceXML) controla la salida de la voz sintetizada y de los audio pregrabados. Los prompts básicos están formados por texto comprendido entre etiquetas <prompt>:

```
<prompt>Por favor, dime tu nombre.</prompt>
```

**Espacio Código II.7: Sintaxis de la Etiqueta <prompt>**

Puede omitirse la sintaxis <prompt> ... </prompt> siempre que no sea necesario especificar un atributo del prompt (como por ejemplo el *bargein* <prompt bargein="true">) y que el prompt consista en PCDATA (no contenga etiquetas SSML) o en etiquetas <audio> o <value>.

- **SSML**

VoiceXML permite utilizar las etiquetas definidas por el lenguaje estándar del W3C SSML (Speech Synthesis Markup Language), en la elaboración de prompts.

Dicho lenguaje debe ser soportado por el sistema de TTS que se integre con el motor de VoiceXML. El motor de síntesis de Verbio (Verbio TTS), soporta en su plenitud las especificaciones del estándar SSML en su versión 1.0, permitiendo así generar prompts que por ejemplo controlen el énfasis o las pausas de las frases:

```
<prompt>Esto es un texto generado por <emphasis> computador  
</emphasis>. <break size="medium"/> ¿Qué te parece? </prompt>
```

**Espacio Código II.8: Uso de la etiqueta <prompt>**

- **AUDIO**

Para reproducir un archivo de audio se define el elemento <audio> dentro del <prompt> con el atributo "src" indicando el nombre del archivo.

```
<prompt>  
    <audio src="audi.wav"/>  
</prompt>
```

**Espacio Código II.9: Reproducción de archivos de audio**

- **DIÁLOGOS**

Existen dos tipos de diálogos, formularios y menús. El primero pretende recolectar información del usuario y el segundo presentarle un conjunto de alternativas.

- **FORMAS O FORMULARIOS**

Un formulario se define por el elemento `<form>` y puede tener dos atributos:

<b>ID</b>	El nombre del formulario. Si está especificada, la forma se puede referir dentro del documento o de otro documento. Para la instancia:  <code>&lt;form id="weather"&gt;, &lt;goto next="#weather"&gt;</code>
<b>Scope</b>	El alcance por defecto del formulario. Si es diálogo entonces las gramáticas del formulario son activas solamente en el formulario. Si el alcance es el documento, entonces las gramáticas del formulario son activas durante cualquier diálogo en el mismo documento. Si el alcance es el documento y el documento es un documento raíz de la aplicación, entonces las gramáticas del formulario son activas durante cualquier diálogo en cualquier documento de esta aplicación. Observe que el alcance de las gramáticas individuales del formulario toma precedencia sobre el alcance por defecto; por ejemplo, en documentos no-raíz en un formulario con el alcance "dialog" por defecto, y una gramática del formulario con el alcance "document", entonces esa gramática es activa en cualquier diálogo en el documento.

**Tabla II.4: Atributos de los formularios.**

## ELEMENTOS DE LOS FORMULARIOS

Son los elementos que se pueden visitar en el lazo principal del algoritmo de la interpretación del formulario. Cuando el FIA (form interpretation algorithm) selecciona un artículo de control, el artículo de control puede contener un bloque del código procedimental para ejecutarse, o puede decir al FIA configurar el prompt y recogerla de un formulario mezclado. Dentro de un formulario existen elementos de entrada, el más común <field> y generalmente pregunta al usuario que acción desea tomar para luego invocar al motor de reconocimiento de voz o tonos DTMF mediante archivos de gramática, para así detectar la respuesta y continuar con la aplicación. Otro elemento es <record>, el cual permite grabar lo que el usuario dice, es muy utilizado para aplicaciones de correos de voz. El elemento <subdialog> permite hacer las llamadas a subdiálogos, que se las considera como un tipo de funciones para realizar alguna acción específica y luego retornar al diálogo desde donde fueron invocados.

## ELEMENTOS DE ENTRADA

Un elemento de entrada especifica una variable de entrada para recolectar la información del usuario. Los artículos de entrada tienen avisos para decirle al usuario qué tiene que decir, las gramáticas que definen las entradas permitidas, y los manejadores de eventos que procesan cualquier acontecimiento que resulte. Un artículo de entrada puede también tener un elemento <filled> que defina una acción a tomar. Los artículos de la entrada consisten en:

<b>&lt;FIELD&gt;</b>	Un elemento de entrada cuyo valor se obtiene vía gramáticas de asr o de dtmf.
<b>&lt;record&gt;</b>	Un elemento de entrada cuyo valor es un clip de audio grabado por el usuario. Un elemento <record> puede recoger un mensaje del correo de voz, por ejemplo.
<b>&lt;transfer&gt;</b>	Un elemento de entrada que transfiere a un usuario a otro número de teléfono. Si la transferencia devuelve el control, la variable de campo será fijada al estado del resultado.
<b>&lt;object&gt;</b>	Este elemento de entrada invoca un "objeto" con varios parámetros.

	<p>El resultado del objeto de la plataforma es un objeto ECMAScript. Un objeto de la plataforma podría ser un diálogo que recopila la información de la tarjeta de crédito.</p> <p>Otros podrían recolectar un mensaje de texto usando un cierto método propietario de entrada de texto (DTMF). No hay requisito para las puestas en práctica de objetos para plataforma específicas, aunque las implementaciones deben manejar el elemento &lt;object&gt; con un error "error.unsupported.objectname", si el objeto en particular no es soportado.</p>
<subdialog>	<p>Un elemento de entrada &lt;subdialog&gt; es como una llamada de función. Invoca otro diálogo en la página actual, o invoca otro documento de VoiceXML.</p>

**Tabla II.5: Elementos de Entrada de un Formulario**

## ELEMENTOS DE CONTROL

Hay dos tipos de elementos de control:

<BLOCK>	<p>Una secuencia de las declaraciones procedimentales usadas para llevar avisos (prompting) y cálculos, pero no para recolectar información de entrada.</p>
<initial>	<p>Este elemento controla la interacción inicial en un formulario. Sus prompts se deben escribir para animar al usuario a que diga algo de acuerdo a la gramática del formulario. Cuando por lo menos una variable entrada se llena como resultado del reconocimiento durante un elemento &lt;initial&gt;, la variable del formulario &lt;initial&gt; llega a ser verdad, eso es quitándolo como alternativa para el FIA.</p>

**Tabla II.6: Elementos de Control del Formulario**

## Ejemplo de Formulario

```
<form>
  <block>
    <prompt>Digitally Imported : Vocal Trance.</prompt>
    <prompt>
      <audio src="uri:http://64.236.98.50:80/stream/1065">
        Transfer Test.
      </audio>
    </prompt>
  </block>
</form>
```

**Espacio Código II.10: Ejemplo de un formulario**

### o MENÚ

Un menú puede ser considerado como un formulario que presenta varias alternativas al usuario y toma una acción determinada cuando se escoge una de ellas. Para simplificar este tipo de formularios que suelen ser muy comunes se ha creado el elemento <menú>. Cada elemento <choice> es una alternativa del menú, su atributo "next" especifica que acción tomar, que generalmente es un diálogo que está en otro archivo VoiceXML.

### Elementos de Menú

Esto identifica el menú, y determina el alcance de sus gramáticas. Los atributos de los elementos del menú son:

<b>id</b>	El identificador del menú. Permite que el menú sea el blanco de un < goto > o un < submit >.
<b>scope</b>	El alcance de la gramática de menú. Si es diálogo (por defecto), las gramáticas del menú son solamente activas cuando el usuario navega en él. Si el alcance es un documento, sus gramáticas son activas sobre el documento entero.
<b>accept</b>	Cuando se configura como "exact" (por defecto), el texto de los elementos escogidos en el menú define la frase exacta que se

	reconocerá. Cuando se configura como "approximate", el texto de los elementos escogidos define una frase aproximada del reconocimiento. Cada < opción > puede eliminar este ajuste.
<b>dtmf</b>	Para aceptar como respuestas pulsaciones de las teclas del teléfono, se le debe asignar el valor "true" para que cada opción del menú corresponda con los números 1, 2, 3, etc. según su orden.
<b>choice</b>	Puede especificar su propio valor DTMF y no ser tomado en cuenta para la asignación automática de números, cabe resaltar que solamente se podrán tener hasta nueve opciones y no se podrán asignar las teclas *, #, ni el número 0.

**Tabla II.7: Atributos de < menú >**

#### Elemento <choice>

El elemento < choice> responde a varios propósitos:

- Puede especificar una gramática de speech,
- Puede especificar una gramática de DTMF.
- El contenido se puede utilizar para enumerar <enumerate> las opciones disponibles para el usuario.
- Y especifica el lanzamiento de un evento o el URI al cual se redirecciona cuando se selecciona la opción.

Los atributos del elemento <choice> son:

<b>Dtmf</b>	La secuencia de DTMF para esta opción. Es equivalente a un simple DTMF <grammar> y las características de DTMF se aplican al reconocimiento de la secuencia. A diferencia de gramáticas de DTMF, los espacios en blanco son opcionales: dtmf="123#" es equivalente a dtmf="1 2 3 #".
<b>accept</b>	Cuando se configura como "exact" (por defecto), el texto de los elementos escogidos en el menú define la frase exacta que se reconocerá. Cuando se configura como "approximate", el texto de

	los elementos escogidos definen una frase aproximada del reconocimiento
<b>next</b>	El URI del diálogo o del documento siguiente.
<b>expr</b>	Especifica una expresión para evaluar como se envía a un URI.
<b>event</b>	Especifica el lanzamiento de un acontecimiento en vez de especificar el siguiente.
<b>eventexpr</b>	Una expresión de ECMAScript que evalúa al nombre del acontecimiento que se lanzará.
<b>message</b>	Una secuencia del mensaje que proporciona contexto adicional sobre el acontecimiento que es lanzado. El mensaje está disponible como el valor de una variable dentro del alcance del elemento.
<b>messageexpr</b>	Una expresión de ECMAScript que evalúa la secuencia del mensaje.
<b>fetchaudio</b>	Esto omite la característica del fetchaudio.
<b>fetchhint</b>	Esto omite la característica del documentfetchhint.
<b>fetchtimeout</b>	Esto omite la característica del fetchtimeout.
<b>maxage</b>	Esto omite la característica del documentmaxage.
<b>maxstale</b>	Esto omite la característica del documentmaxstale.

**Tabla II.8: Atributos <option>**

Exactamente uno de "next", "expr", "event" o "eventexpr" debe ser especificado; si no, un acontecimiento de error.badfetch es lanzado. Exactamente uno de "message" o de "messageexpr" puede ser especificado; si no, se lanza un acontecimiento de error.badfetch. Si un elemento gramatical < grammar > es especificado en <choice>, entonces la gramática externa se utiliza en vez de una gramática generada automáticamente. Esto permite que el desarrollador controle precisamente la gramática de <choice>

## Ejemplo de Menú

```
<menu id="maimmenu" dtmf="true">
  <prompt bargein="true" xml:lang="text">
    MENU
    1) Temperatura
    2) Tiendas
  </prompt>
  <prompt bargein="true">
    Elige tu opción.
    Para saber la temperatura de su oficina, el tipo 1.
    Para ordenar las tiendas del edificio, el tipo 2.
  </prompt>
  <choice next = "prueba.php" dtmf = "1"/>
  <choice next = "stores.vxml" dtmf = "2"/>
</menu>
```

### Espacio Código II.11: Ejemplo de un Menú

- **VARIABLES**

Dentro de un archivo VoiceXML se puede tener fragmentos de código ECMAScript, para lo cual se deben definir variables, el elemento `<var>` se encarga de esto. El nombre de la variable se lo especifica en el atributo "namelist" y si son dos o más variables que se desea dejar sin valor se las escribe seguidamente separadas por un espacio en blanco.

El ámbito de una variable depende del lugar en el documento VoiceXML en donde fue declarada, puede ser de sesión, aplicación, documento, diálogo y de contexto local. Las variables de sesión no son declaradas en los archivos VoiceXML sino que las provee la plataforma que los interpreta y generalmente sirven para indicar ciertos parámetros propios de dicha plataforma, estas variables son de solo lectura y pueden ser referenciadas desde cualquier aplicación. Las variables de aplicación y documento hacen referencia a las declaradas dentro del elemento `<vxml>` en un documento raíz de una aplicación o en documentos secundarios respectivamente, y como su nombre lo indica, se puede acceder a ellas desde la aplicación o solamente desde el documento en cuestión. El ámbito de diálogo corresponde a las variables declaradas dentro de un

formulario o menú respectivamente y el contexto local a las que se declara dentro de algún elemento adicional a los mencionados, como puede ser dentro de una excepción

```
<form>
  <block>
    <var name="caller" expr="session.connection.remote.uri"/>
    <submit next="temperature.php" method="post" namelist="caller"/>
  </block>
</form>
```

### Espacio Código II.12: Uso de Variables

- **EXCEPCIONES**

Cuando un usuario no responde, cuando el motor de reconocimiento no puede entender su respuesta, cuando solicita ayuda, o cuando no se puede enrutar una llamada, entre otras, la plataforma que soporta la aplicación lanzará excepciones. Para capturarlas existe el elemento `<catch>` que tiene por atributos “event”, “count” y “cond” y puede ser incluido dentro de un documento o un diálogo.

El atributo “event” especifica el evento o eventos que serán capturados en caso de que ocurran, por ejemplo:

- **“error”** cuando ha ocurrido un error general, se puede especificar el tipo de error mediante sus propiedades, así “error.noresource” es lanzado cuando en la aplicación se referencia a un recurso que no está disponible o “error.semantic” cuando un error de ejecución se ha encontrado en el documento VoiceXML.
- **“noinput”** cuando el usuario no ha pronunciado ninguna respuesta.
- **“nomatch”** cuando el motor de reconocimiento no entiende la respuesta del usuario.
- **“help”** cuando el usuario ha pronunciado “ayuda”.

Para capturar dos o más eventos en un mismo `<catch>` se los debe especificar en el atributo “event” separados por un espacio en blanco.

Además se puede abreviar el uso de estas etiquetas para capturar eventos, `<error>` es un equivalente a `<catch event="error">`, `<nomatch>` es un equivalente a `<catch event="nomatch">`, etc.

Un evento puede ser lanzado varias veces en un diálogo, el usuario puede pronunciar más de una vez su respuesta sin que el motor de reconocimiento tenga éxito en comprenderla; para manejar la captura de eventos de acuerdo al número de veces que hayan ocurrido se utiliza el atributo “count”, por ejemplo, si se desea que al tercer intento de detección de una respuesta no válida se acabe la aplicación se debería tener una estructura similar a la del espacio de código \*\*. Al atributo “cond” se le debe asignar el valor “true” si se desea que se capture la excepción, pero por defecto lo tiene asignado, por esta razón suele ser omitido.

Además de los eventos predefinidos por la recomendación se puede lanzar eventos creados el usuario, para esto existe el elemento <throw> que tiene por atributos “event” y “message” entre otros, el primero identifica el tipo de evento que es lanzado y el segundo una descripción del mismo.

#### 2.5.9.2 EJEMPLOS DE VOICEXML

Los siguientes ejemplos constituyen una guía para un mejor entendimiento de los conceptos antes mencionados. Dichos ejemplos funcionan correctamente en el motor VXAsterisk de la empresa i6net.

- **Código de “hola mundo”**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <form id="saludo">
    <block>Hola mundo</block>
  </form>
</vxml>
```

#### Espacio Código II.13: Hola Mundo

- **Código de “Colores”**

El siguiente ejemplo utiliza gramáticas en línea para el reconocimiento de colores, el programa devuelve el código en hexadecimal del valor reconocido.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
<!--root document  example-starttointerpret this file-->
<form id="foo">
  <field name="color">
    <grammar mode="voice" type="application/verbio-text">
      3400(negro)          negro
      3500( blanco)       blanco
      3600(verde)         verde
      3700( azul)         azul
      3800(purpura)       purpura
      3900( amarillo)     amarillo
      4000(rojo)          rojo
    </grammar>
    <prompt>
      Si dices tu color favorito, yo te dire su codigo en hexadecimal.
      Que <emphasis>color? </emphasis>
    </prompt>
    <filled> <var name="color_code"/>
    <if cond="color == 'negro'"> <assign name="color_code" expr="000000"/>
    <elseif cond="color == 'blanco'"><assign name="color_code" expr="FFFFFF"/>
    <elseif cond="color == 'verde'"><assign name="color_code" expr="00FF00"/>
    <elseif cond="color == 'azul'"><assign name="color_code" expr="0000FF"/>
    <elseif cond="color == 'purpura'"><assign name="color_code" expr="7D26CD"/>
    <elseif cond="color == 'amarillo'">
    <assign name="color_code" expr="8B8B00"/>
    <elseif cond="color == 'rojo'">
    <assign name="color_code" expr="CD0000"/>
    <else/>
    <assign name="color_code" expr="?" />
    </if>
    <prompt>
    el codigo para <value expr="color"/> es <value expr="color_code"/>
    </prompt>
    <clear namelist="color color_code"/>
  </filled>
</field>
</form>
</vxml>
```

**Espacio Código II.14: Ejemplo de gramáticas locales e if-else**

- **Código de “Menú”**

El siguiente ejemplo presenta un menú de 2 opciones que utilizan DTMF para la navegación, para la realización de los cálculos en una de sus opciones utiliza código en lenguaje PHP combinado con etiquetas VoiceXML.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-
ES">

<property name="prompturl2"
value="http://titus.i6net.org/tts/verbio/tts.php"/>
<property name="promptposition" value="00" />
<property name="termtimeout" value="5s"/>

<menu id="maimmenu" dtmf="true">
<prompt bargein="true" xml:lang="text">
MENU
1) Temperatura
2) Tiendas
</prompt>
<prompt bargein="true">
Elige tu opción.
Para saber la temperatura de su oficina, el tipo 1.
Para ordenar las tiendas del edificio, el tipo 2.
</prompt>
<choice next = "temperature.php" dtmf = "1"/>
<choice next = "stores.vxml" dtmf = "2"/>
</menu>
</vxml>
```

**Espacio Código II.15: Script menú**

```
<?php echo '<?xml version="1.0" encoding="iso-8859-1"?>?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-
ES">

  <property name="prompturl2"
value="http://titus.i6net.org/tts/verbio/tts.php"/>
  <property name="promptposition" value="00" />

  <form>
    <block>
      <prompt bargein="false" xml:lang="text">
        TEMPERATURA

      <?php
$caller = $_REQUEST["caller"];
$temp = rand(180, 280)/10;
echo $temp."°C";
?>
        </prompt>
        <prompt bargein="false">
Actualmente se encuentra en su oficina:
        </prompt>
        <prompt bargein="false">
<?php echo number_format($temp, 1, ',', ''); ?>
        </prompt>
        <prompt bargein="false">
grados Celsius.
        </prompt>
        <goto next="menu.vxml"/>
    </block>
  </form>
</vxml>
```

**Espacio Código II.16: Script temperature.php**

```
<?php echo '<?xml version="1.0" encoding="iso-8859-1"?>'; ?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="fr-
FR">
  <property name="prompturl2"
value="http://titus.i6net.org/tts/verbio/tts.php"/>
  <property name="promptposition" value="60" />

  <form>
    <block>
      <prompt bargein="false" xml:lang="text">
<?php
$caller = $_REQUEST["caller"];
$store = $_REQUEST["store"];
$action = $_REQUEST["action"];
if ($action == "ouvrir")
echo " Apertura $store
Actual...";
else
echo " Cierre $store
Actual...";
?>
      </prompt>
      <prompt bargein="false">
<?php
$caller = $_REQUEST["caller"];
$store = $_REQUEST["store"];
$action = $_REQUEST["action"];
if ($action == "fermer")
echo "Cierre $store actual.";
else
echo "Apertura $store actual.";

?>
      </prompt>
      <goto next="menu.vxml"/>
    </block>
  </form>
</vxml>
```

**Espacio Código II.17: Script store.php**

- **Código de “gramáticas por referencia”**

El código descrito a continuación muestra un ejemplo de reconocimiento de voz por medio de gramáticas de referencia.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-laura">
  <form>
    <property name="inputmodes" value="voice"/>
    <property name="timeout" value="30s"/>
    <property name="promptmaxage2" value="0s"/>
    <property name="promptformat2" value="alaw" />
    <property name="prompturl2"
value="http://titus.i6net.org/tts/verbio/tts.php"/>
    <field name="text">
      <grammar mode="voice" src="vdemo.txt"/>
      <catch event="noinput nomatch">
        <reprompt/>
      </catch>
      <prompt>
        Seleccionar :
        Voy a hablar durante la seleccion :
        El cuento es una narración breve, escrita en prosa, en la que se relata una historia
        tanto real como ficticia. Además de su brevedad, el cuento tiene otras
        características estructurales que lo diferencian de la novela, la frontera entre un
        cuento largo y una novela corta no es fácil de definir. También se caracteriza por
        tener un reducido número de personajes y escasas descripciones. En lengua
        castellana, la redacción de cuentos es una especialidad de América, en
        contraposición a la especialidad española en filología y realismo.
      </prompt>
    </field>
    <filled>
      <prompt>
        He reconocido :
      <value expr="text" />
    </prompt>
    <clear namelist="text" />
  </filled>
</form>
</vxml>
```

100(quebrantahuesos)	quebrantahuesos
200(Eduardo Campillo)	Eduardo Campillo
300(Asunción China)	Asunción China
400(Marta Cuenca)	Marta Cuenca
500(Emilio Doblado)	Emilio Doblado
600(Guillermo Estrada)	Guillermo Estrada
700(María Frías)	María Frías
800(Sergio Galíndez)	Sergio Galíndez
900(Eloy de la Fuente)	Eloy de la Fuente
1000(Enrique Hurtado)	Enrique Hurtado
1100(Pablo Jiménez)	Pablo Jiménez
1200(Francisco Luna)	Francisco Luna
1300(Fernando Mayoral)	Fernando Mayoral
1400(Susana Martínez)	Susana Martínez
1500(Germán Morales)	Germán Morales
1600(Alfonso Morillo)	Alfonso Morillo
1700(Juan Oliva)	Juan Oliva
1800(Inmaculada Sánche)	Inmaculada Sánchez
1900(Gloria Machado)	Gloria Machado
2000(Pilar Zaragoza)	Pilar Zaragoza

**Espacio Código II.19: Archivo “vdemo.txt”**

## **CAPÍTULO III**

### **IMPLEMENTACIÓN DE LA CENTRALITA**

#### **3.1 INTRODUCCIÓN**

En el presente capítulo se describe la implementación de la centralita, la misma que servirá para demostrar el correcto funcionamiento del prototipo propuesto en esta investigación, pero antes de proceder a la implementación de la misma primero describiremos los fundamentos teóricos de los elementos tanto hardware y software que permitirá realizar el desarrollo e implementación del prototipo de una manera adecuada. Las herramientas software que se utilizará se encuentran entre libres y propietarias, en el caso de las propietarias se utilizarán versiones de evaluación o demos por tratarse de un prototipo de demostración únicamente. El principal elemento hardware a utilizar es una tarjeta telefónica que permitirá integrar el prototipo con la PSNT, concretamente la tarjeta de marca OpenVox A400P.

### 3.2 ELEMENTOS HARDWARE Y SOFTWARE

En las siguientes tablas se describe los elementos hardware y software que serán utilizados para el desarrollo total del prototipo.

CANTIDAD	DESCRIPCIÓN	CARACTERÍSTICAS
1	Servidor	512 RAM 40 GB en Disco Duro Tarjeta de Red
1	Tarjeta OpenVox A400P	1 modulo FXO 1 Modulo FXs
1	Computador de Escritorio	512 RAM 40 GB en Disco Duro Tarjeta de Red Entrada y salida de audio
1	Micrófono	Básico
1	Audífono	Básico

**Tabla III.9: Elementos Hardware**

DESCRIPCIÓN	LICENCIA	FUNCIÓN
Centos 5.4	GPL	Sistema operativo del servidor
Windows XP	Propietario	Sistema operativo del cliente
Asterisk	GPL	Software para la PBX
Dahdi Linux	GPL	Controladores y librerías de la tarjeta Openvox.
Dahdi Tools	GPL	Programas y herramientas para interactuar con la tarjeta Openvox.
Verbio Engines	Evaluación	Motores de síntesis y reconocimiento
Verbio Client	Evaluación	Librerías de desarrollo

VXIasterisk	Evaluación	Motor de VoiceXML
Verbio Text To Speech	Evaluación	Motor de síntesis de voz
Verbio Automatic Speech Reginocition	Evaluación	Motor de reconocimiento de voz
Apache	GPL	Servidor Web
PHP	GPL	Lenguaje de programación para la web
Mysql	GPL	Servidor de base de datos
Webmin	GPL	Herramienta de configuración de sistemas accesible vía web

**Tabla III.10: Elementos Software**

### **3.3 DESCRIPCIÓN DE LOS ELEMENTOS SOFTWARE**

#### **3.3.1 SISTEMA OPERATIVO**

La tendencia actual en uso de los sistemas operativos son sin duda los de libre distribución, como sistema operativo sobre el cual marchará asentada la plataforma del prototipo se utilizará la distribución Centos 5.4.

##### **3.3.1.1 REQUERIMIENTOS DEL SISTEMA**

Se debe contar con la suficiente cantidad de memoria y un microprocesador en buen estado. Con casi cualquier distribución comercial de Linux, el ambiente gráfico necesitará al menos 192 MB RAM, y 650-800 MB de espacio en disco duro para la instalación mínima. Para contar con la menor cantidad de aplicaciones prácticas, se requieren al menos 800 MB adicionales de espacio en disco duro, repartido en al menos 2 particiones. Se recomienda un microprocesador 80586 (pentium o equivalente) a 200 MHz. Sin ambiente gráfico, como es el caso de un servidor, o bien solamente aplicaciones para modo de texto, 64MB RAM y un microprocesador 80586 a 100 MHz serán suficientes.

Si desea instalar Linux en una computadora personal con las suficientes aplicaciones para ser totalmente funcional, productivo y contar con el espacio necesario para instalar herramientas de oficina (OpenOffice.org), se recomienda contar con al menos

2 GB de espacio, al menos 256 MB RAM y un microprocesador AMD K6, K6-II, K6-III, Athlon, Duron, Pentium, Pentium MMX, Pentium II, Pentium III, Pentium 4, o Cyrix MII a cuando menos 300 Mhz o más. (Alcance Libre)

### **3.3.1.2 CARACTERÍSTICAS PRINCIPALES**

- GNU/Linux® es un poderoso y sumamente versátil sistema operativo con licencia libre y que implementa el estándar POSIX (acrónimo de Portable Operating System Interface, que se traduce como Interfaz de Sistema Operativo Portable).
- El usuario es libre de redistribuir y modificar de acuerdo a necesidades específicas, siempre que se incluya el código fuente, como lo indica la Licencia Pública General GNU (acrónimo de GNU is Not Unix), que es el modo que ha dispuesto la Free Software Foundation (Fundación de Programática Libre). Esto también incluye el derecho a poder instalar el núcleo de GNU/Linux® en cualquier número de ordenadores o equipos de cómputo que el usuario desee.
- La GPL (acrónimo de General Public Licence, que se traduce como Licencia Pública General), a la cual Linus Torvalds incorporó a Linux, está diseñada para asegurar que el usuario tenga siempre la libertad de distribuir copias del sustento lógico libre (y cobrar por el servicio si así lo desea). La GPL tiene como objetivo garantizar al usuario la libertad de compartir y cambiar; es decir, asegurarse de que el sustento lógico siempre permanezca siendo libre para todos los usuarios. La GPL es aplicable a la mayoría del sustento lógico de la Free Software Foundation así como a cualquier otro programa cuyos autores se comprometan a usarlo.
- Es un sistema operativo idóneo para utilizar en Redes, como es el caso de servidores, estaciones de trabajo y también para computadoras personales.
- Permiten desempeñar múltiples tareas en forma simultánea de forma segura y confiable. Los distintos servicios se pueden detener, iniciar o reiniciar independientemente sin afectar al resto del sistema, permitiendo operar las 24 horas del día los 365 días del año.

### **3.3.2 VXIAsterisk**

La plataforma VXIasterisk VoiceXML browser (VXI\*) desarrollada por I6NET permite combinar servicios avanzados de portal de voz y portal de video sobre el estándar W3C VoiceXML. VXI\* es más flexible, fácil de instalar y de configurar que otros sistemas actuales, lo que permite extender estos servicios de voz hacia proveedores de aplicaciones móviles que hasta ahora dependían de entornos más costosos u orientados a operadores.

VXI\* es una plataforma convergente por excelencia que permite procesar contenidos en formatos wav, gsm, mp3, 3gp sobre redes de telefonía convencional, VoIP, GSM, UMTS-3G. Con un solo servidor de hardware estándar que funciona a la vez de gateway IP/3G y de motor IVVR, se ejecutan aplicaciones interactivas alojadas en entorno web.

VXI\* es completamente escalable, lo que permite también a grandes empresas apostar por un modelo de ahorro de costes y desplegar servicios de forma progresiva.

VXI\* incorpora motores de síntesis de voz (TTS) y reconocimiento de voz (ASR) de diferentes proveedores reconocidos en modo plugin, de tal forma que el cliente puede elegir los tipos de componentes adicionales para sus servicios interactivos.

VXI\* es ideal para empresas que desean lanzar servicios convergentes web/móvil con voz o vídeollamada 3G. Está totalmente basado en estándares y soporta VoiceXML, lo que le permite construir aplicaciones abiertas de forma evolutiva. I6NET contribuye al desarrollo de nuevos servicios de telefonía avanzada incorporando su motor VXI\* basado en software, auto-instalable en servidores convencionales, fácilmente escalable y con costes más reducidos.

### **3.3.3 DAHDI**

DAHDI es el acrónimo para Digium Asterisk Hardware Device Interface. En este proyecto se ha separado en 2 paquetes, dahdi-linux y dahd-tools, la primera contiene los controladores y librerías para el hardware digium y la segunda contiene los programas y herramientas para interactuar con este hardware.

### **3.3.4 ASTERISK**

Asterisk es una aplicación software libre de una central telefónica (PBX). Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una RDSI tanto básicos como primarios. Fue creado por Mark Spencer, de Digium, y en la actualidad es su principal desarrollador, junto con otros programadores que han contribuido a corregir errores y añadir novedades y funcionalidades. Originalmente desarrollado para el sistema operativo GNU/Linux, Asterisk actualmente también funciona en BSD, MacOSX, Solaris y Microsoft Windows, aunque la plataforma nativa (GNU/Linux) es la mejor soportada de todas.

Asterisk incluye muchas características anteriormente sólo disponibles en costosos sistemas propietarios PBX: buzón de voz, conferencias, IVR, distribución automática de llamadas y otras muchas más. Los usuarios pueden crear nuevas funcionalidades escribiendo un dialplan en el lenguaje de script de Asterisk o añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación soportado por Linux. Para conectar teléfonos normales analógicos hacen falta unas tarjetas telefónicas con un modulo FXS o FXO fabricadas por Digium o por otros fabricantes<sup>13</sup>.

Asterisk se utilizará como un Gateway que integrará la Red Telefónica Tradicional (PSTN) con VXiAsterik lo cual que permitirá acceder a la aplicación telefónica utilizando un teléfono convencional.

### **3.3.5 VERBIO TTS**

La tecnología TTS permite generar voces-mensajes hablados-a partir de un texto escrito, en forma totalmente automática. Las voces incorporadas dentro del motor TTS (texto a voz) se caracterizan por:

- Voces con gran naturalidad de hombre y mujer.
- Entorno multilingüe.
- Entonación dinámica y adaptativa.

---

<sup>13</sup> Manual de Instalación y Configuración de un Servidor Asterisk

- Algoritmo de ritmos y pausas del habla basado en la propia sintaxis y en métodos estadísticos.
- Locutor corporativo: disponer de una voz sintética a su medida o a la de su cliente.

Es una herramienta imprescindible en aplicaciones cuyo texto, es desconocido o muy cambiante y por lo tanto hay que comunicar información dinámica. La síntesis no pretende sustituir nunca a la calidad de una grabación profesional hecha en un estudio de grabación, pero sí dispone de calidad suficiente como para sustituir temporalmente cuando se necesitan de forma inmediata mensajes o locuciones estáticas o definitivamente cuando lo que se pretende es un ahorro directo de los costos.

Cuando se persigue mayor calidad que lo que representa la simple concatenación directa de ficheros .wav de pequeñas locuciones grabadas en estudio, ya que con ella se perdería la entonación dinámica necesaria en un idioma concreto. Un ejemplo típico de ello es la locución de cantidades numéricas. Con la utilización TTS en un sistema de información se evita la grabación constante en estudio de nuevos mensajes y se consigue por lo tanto un ahorro elevado en los costos y tiempo, disponiendo de una flexibilidad total y disponibilidad inmediata. Este software es el encargado de generar voz natural en tiempo real de cualquier texto e idioma, en nuestro caso se utilizar la voz en castellano de “Carlos”, en una versión de evaluación. La conversión de texto en habla permite que una aplicación tenga la funcionalidad de transformar un texto escrito en habla. Es decir, permite que la máquina pueda “leer en voz alta” cualquier texto.

Para ello utiliza una expansión inteligente de números, fechas, abreviaturas, etc., que puede personalizarse mediante diccionarios propios. Mediante un lenguaje estándar de marcas se puede controlar el comportamiento del motor, para cambiar el idioma o el locutor, variar la velocidad de elocución, introducir pausas, solicitar notificaciones, etc<sup>14</sup>.

---

<sup>14</sup>Guía de Verbio

<http://download-book.net/verbio-manual-pdf.html>

### **3.3.6 VERBIO ASR**

El Reconocimiento Automático del Habla convierte de forma automática una locución hablada en formato de texto en cualquier idioma. Permite una interacción muy natural, puesto que reconoce cualquier palabra o persona sin entrenamiento previo. ASR es una herramienta imprescindible en aplicaciones interactivas entre usuarios y sistemas automáticos o de control por voz. Cuando se requiere una navegación más flexible, con iniciativa y control por parte del usuario.

Para ahorrar tiempos de interacción y para evitar memorización de menús muy largos. En aplicaciones en que el teléfono, o simplemente la voz, siga siendo el único canal o medio de interacción.

Verbio ASR ha sido entrenado con una población de miles de hablantes de cada lengua en una proporción equilibrada en sexo y edad y de todas las partes del territorio de uso de cada lengua, recogiendo sus voces a través de la red telefónica fija y móvil para los modelos a 8KHz, así como micrófono para los modelos a 16KHz.

Por tratarse de un software comercial para la demostración del prototipo se lo utilizará únicamente una versión de prueba, misma que se encuentra activa por un lapso de treinta minutos, después del cual se debe reiniciar el servidor Verbio-ASR.

### **3.3.7 SERVIDOR WEB APACHE**

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache es usado primariamente para enviar páginas web estáticas y dinámicas en la WWW. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby).

Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos Oracle y el IBM WebSphere application server. Mac OS X integra apache como parte de su propio servidor web y como soporte de su servidor de aplicaciones WebObjects. Es soportado de alguna manera por Borland en las herramientas de desarrollo Kylix y Delphi. Apache es incluido con Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux.

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos.

Los programadores de aplicaciones web a veces utilizan una versión local de Apache en orden de previsualizar y probar código mientras éste es desarrollado.

Microsoft Internet Information Services (IIS) es el principal competidor de Apache, así como Sun Java System Web Server de Sun Microsystems y un anfitrión de otras aplicaciones como Zeus Web Server

La mayor parte de la configuración se realiza en el fichero httpd.conf. Cualquier cambio en este archivo requiere reiniciar el servidor.

La licencia de software bajo la cual el software de la fundación Apache es distribuido es una parte distintiva de la historia de Apache HTTP Server y de la comunidad de código abierto. La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original.

El nombre Apache es una marca registrada y puede ser sólo utilizada con el permiso expreso del dueño de la marca.

### **3.3.7.1 CARACTERÍSTICAS**

- Corre en una multitud de Sistemas Operativos.
- Es una tecnología gratuita de código fuente abierta.
- Altamente configurable de diseño modular.
- Trabaja con gran cantidad lenguajes de script.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar.
- Tiene una alta configurabilidad en la creación y gestión de logs.

### **3.3.7.2 MÓDULOS**

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web.

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- mod\_perl: Páginas dinámicas en Perl.
- mod\_php: Páginas dinámicas en PHP.
- mod\_python: Páginas dinámicas en Python.
- mod\_rexx: Páginas dinámicas en REXX y Object REXX.
- mod\_ruby: Páginas dinámicas en Ruby.
- mod\_aspdotnet: Páginas dinámicas en .NET de Microsoft  
(Módulo retirado).
- mod\_mono: Páginas dinámicas en Mono

### 3.3.8 SERVIDOR DE BASE DE DATOS MYSQL

MySQL es un sistema que administra base de datos relacionales, multihilo y multiusuario que actualmente es desarrollada por la empresa MySQL AB y Sun Microsystems. Estas herramientas desarrollan MySQL bajo una licencia dual software libre y comercial, lo que quiere decir que se puede adquirir bajo los términos de la licencia GNU GPL o con condiciones específicas para incluirlo en productos privados.

MySQL es propietario y está patrocinado por una empresa privada la cual posee el copyright de la mayor parte del código, por esta razón su licencia dual.

La compañía ofrece soporte técnico y servicios en todo el mundo.

#### 3.3.8.1 CARACTERÍSTICAS

- MySQL tiene soporte para conectarse con muchos lenguajes de programación como: C, C++, C#, Java, Lisp, Perl, PHP, Python, Ruby, etc. Y para lenguajes que tengan soporte para conexiones ODBC.
- MySQL es muy utilizado en aplicaciones web y de escritorio y por herramientas seguidores de errores como bugzilla.
- MySQL funciona sobre una gran cantidad de plataformas como Windows, Linux, OpenBSD, HP-UX, BSD, Solaris, etc.

### 3.3.9 PHP (Hypertext Preprocessor)

**Hypertext Preprocessor** es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente PHP Tools, o, *Personal Home Page Tools*).

PHP es un lenguaje creado por una gran comunidad de personas. Actualmente PHP se encuentra en su versión 4, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades actuales y solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión son su rapidez, gracias a

que primero se compila y luego se ejecuta, mientras que antes se ejecutaba mientras se interpretaba el código, su mayor independencia del servidor web, creando versiones de PHP nativas para más plataformas- y un API más elaborado y con más funciones.

El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una pagina WML. Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del browser, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

(Desarrollo Web con PHP y Mysql, Ingeniero Joel Gonzales Estrada)

### **3.3.9.1 CARACTERÍSTICAS**

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

### **3.3.9.2 FRAMEWORK**

- Zend Framework
- PHP Prado
- Symfony
- CakePHP
- Qcodo
- Kumbia
- PHP4ECore
- CodeIgniter
- Yii Framework
- MfwLite
- Tomates Framework

### **3.3.9.3 ENTORNOS DE DESARROLLO PARA PHP**

Algunos de los Entornos de Desarrollo Integrados para PHP más conocidos o habituales son:

- Zend Studio: Comercial (Zend).
- PDT, plugin de Eclipse: GPL (Sun).
- Komodo IDE: Libre y gratuito, el IDE es licencia comercial - (Mozilla).
- NuSphere PhpED: Comercial, para Linux y Windows.
- NetBeans: Libre, para Linux y Windows.
- Quanta: GPL y gratuito, para GNU/Linux con QT.
- Bluefish: GPL y gratuito, para GNU/Linux con GTK.
- phpDesigner: Comercial y Freeware, para Linux y Windows.

- Rapid PHP: Comercial para Windows.
- Aptana Studio: GPL, existe una versión comercial.

### **3.4 ELEMENTOS HARDWARE**

La tarjeta telefónica que permite la integración del prototipo con la PSTN, el único elemento hardware a mencionar quedando fuera del alcance de este proyecto la descripción de otros elementos hardware como computador, tarjetas de red, etc.

#### **3.4.1 TARJETA TELEFÓNICA DE COMUNICACIONES**

Existen diferentes fabricantes de tarjetas para sistemas Asterisk que cubren las necesidades de conectividad de cualquier tipo de empresas.

El fabricante oficial de tarjetas de comunicaciones para Asterisk es la empresa Digium. Otras compañías como: OpenVox, Sangoma, Eicon, también están dedicadas a la fabricación de tarjetas clónicas de las Digium mucha de la veces tarjetas con funcionalidades más avanzadas.

En general, las tarjetas empleadas se conectan al equipo servidor mediante conectores PCI, el tipo depende de la tarjeta, y su funcionalidad puede ir desde la conexión básica a la red, a funciones que incluyen incluso la señalización y codificación, aliviando al servidor de la carga de CPU que estas funciones requieren.

##### **3.4.1.1 TARJETA OPENVOX A400P**

Las tarjetas analógicas permiten conectar el servidor Asterisk con la red de telefonía conmutada y/o con teléfonos analógicos. Uno de los principales ingresos de la empresa Digium , dejando al margen el soporte y mantenimiento de servidores Asterisk, es la implementación, distribución y venta de tarjetas PCI compatibles dedicadas a gestionar el tráfico de voz. TDM es el nombre comercial que recibe las tarjetas analógicas desarrollados por Digium y está formada por un conjunto de módulos extraíbles que permiten la conversión analógico-digital y viceversa:

- FXO (módulos rojos): permite conectar tantas líneas externas de la PSTN como módulos.
- FXS (módulos verdes): permite conectar tantos teléfonos analógicos como módulos.

Existen básicamente dos familias de tarjetas analógicas:

A400P: tarjeta mini-PCI 2.2 con capacidad de conectar hasta 4 interfaces FXO (nomenclatura comercial: X100M), 4 módulos FXS (S110M) o cualquier combinación de estos con un máximo de 4 módulos. Digium ha asignado a este grupo de tarjetas la nomenclatura TDM X Y B, siendo X el número de interfaces FXS que incorpora la tarjeta, e Y el número de interfaces FXO. Por tanto, la tarjeta TDM13B es una tarjeta de la familia TDM400P con un módulo FXS y tres interfaces FXO.

#### **3.4.1.2 CARACTERÍSTICAS**

- Diseño Modular: Hasta 4 FXS, FXO o FXS mixta / puertos FXO por tarjeta
- Escalable: Sólo tiene que añadir tarjetas adicionales para ampliar el sistema
- Fácil de usar: software completo y hardware compatibles con Digium™'sTDM400P.
- World Wide se pueden utilizar: interfaz de línea configurable para cumplir con la línea telefónica mundial requisitos de la interfaz
- Alta calidad con precio bajo: El uso de Asterisk® para construir el sistema IP-PBX/Voicemail

### 3.5 IMPLEMENTACIÓN DE LA PLATAFORMA DE PRUEBAS

La implementación de la plataforma de pruebas se encuentra realizada de acuerdo al esquema del Gráfico III.21.



**Gráfico III.21: Escenario de la plataforma a implementar**

En el gráfico III.21 se presenta de una manera gráfica el esquema de la plataforma a implementar, se encuentran integrados tanto los elementos hardware y software a utilizar.

La idea principal de esta parte del estudio es crear un documento que permita plasmar los pasos del proceso a seguir para implementar la plataforma de una forma clara y concisa. Para facilidad de los futuros usuarios el proceso se realizara de una forma "Instalación-Configuración" para un mejor entendimiento, es decir los componentes posibles se precederá a su instalación y seguidamente a su configuración respectiva.

#### 3.5.1 INSTALACIÓN DE CENTOS 5.4

El proceso de instalación del sistema operativo a utilizarse queda fuera del alcance de esta investigación por tratarse de proceso ya conocido por los usuarios que posiblemente utilicen este documento para trabajos futuros. La versión a utilizar de Centos 5.4, es una versión estable la misma que no requiere actualización alguna para esta implementación.

### **3.5.2 INSTALACIÓN Y CONFIGURACIÓN DEL MODULO DAHDI Y TARJETA OPENVOX A400P**

El modulo DAHDI contiene los controladores que permiten integrar la red PSTN con una PBX mediante las interfaces telefónicas que pueden ser analógicas o digitales. Las fuentes dependen del sistema operativo a utilizarse, para el presente estudio las fuentes requeridas obligatoriamente deben ser compatibles con la versión 5.4 de la distribución Red-Hat (Centos). Las fuentes de dahdi y asterisk se las pueden encontrar en la siguiente dirección web: <http://www.i6net.com/downloads/centos-32bit-5/>

Para descargar cada una de los paquetes necesarios debe crearse una carpeta llamada asterisk donde repositará las descargas:

```
# mkdir /usr/src/asterisk
```

```
# cd /usr/src/asterisk
```

#### **3.5.2.1 INSTALACIÓN DE DAHDI-LINUX**

Se requieren los siguientes pasos:

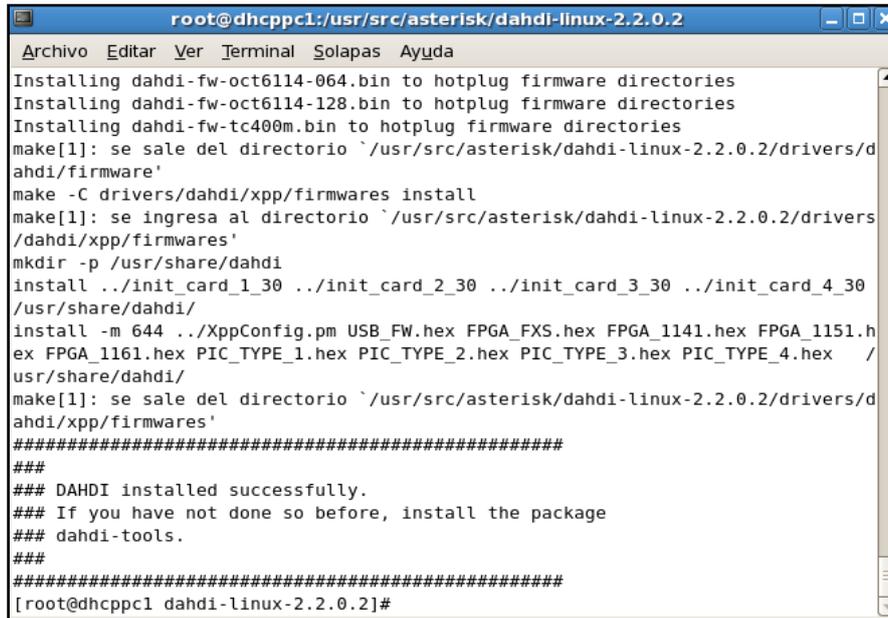
```
# tar -xvzf dahdi-linux-2.2.0.2.tar.gz
```

```
# cd dahdi-linux-2.2.0.2
```

```
# make
```

```
# make install
```

RESULTADO:



```
root@dhcpccl:/usr/src/asterisk/dahdi-linux-2.2.0.2
Archivo Editar Ver Terminal Solapas Ayuda
Installing dahdi-fw-oct6114-064.bin to hotplug firmware directories
Installing dahdi-fw-oct6114-128.bin to hotplug firmware directories
Installing dahdi-fw-tc400m.bin to hotplug firmware directories
make[1]: se sale del directorio `/usr/src/asterisk/dahdi-linux-2.2.0.2/drivers/dahdi/firmware'
make -C drivers/dahdi/xpp/firmwares install
make[1]: se ingresa al directorio `/usr/src/asterisk/dahdi-linux-2.2.0.2/drivers/dahdi/xpp/firmwares'
mkdir -p /usr/share/dahdi
install ../init_card_1_30 ../init_card_2_30 ../init_card_3_30 ../init_card_4_30 /usr/share/dahdi/
install -m 644 ../XppConfig.pm USB_FW.hex FPGA_FXS.hex FPGA_1141.hex FPGA_1151.hex FPGA_1161.hex PIC_TYPE_1.hex PIC_TYPE_2.hex PIC_TYPE_3.hex PIC_TYPE_4.hex /usr/share/dahdi/
make[1]: se sale del directorio `/usr/src/asterisk/dahdi-linux-2.2.0.2/drivers/dahdi/xpp/firmwares'
#####
###
### DAHDI installed successfully.
### If you have not done so before, install the package
### dahdi-tools.
###
#####
[root@dhcpccl dahdi-linux-2.2.0.2]#
```

Gráfico III.22: Instalación Satisfactoria de Dahdi Linux

### 3.5.2.2 INSTALACIÓN DE DAHDI-TOOLS

Se requieren los siguientes pasos:

```
# tar -xvzf dahdi-tools-2.2.0.tar.gz
```

```
# cd dahdi-tools-2.2.0
```

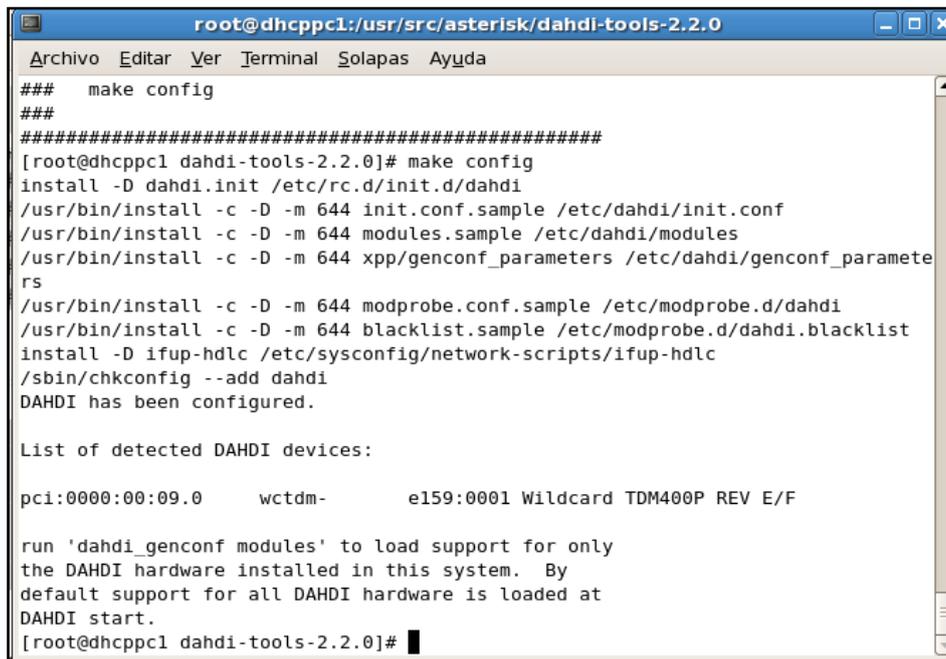
```
# ./configure
```

```
# make
```

```
# make install
```

```
# make config
```

RESULTADO:



```
root@dhcpc1:/usr/src/asterisk/dahdi-tools-2.2.0
Archivo Editar Ver Terminal Solapas Ayuda
### make config
###
#####
[root@dhcpc1 dahdi-tools-2.2.0]# make config
install -D dahdi.init /etc/rc.d/init.d/dahdi
/usr/bin/install -c -D -m 644 init.conf.sample /etc/dahdi/init.conf
/usr/bin/install -c -D -m 644 modules.sample /etc/dahdi/modules
/usr/bin/install -c -D -m 644 xpp/genconf_parameters /etc/dahdi/genconf_paramete
rs
/usr/bin/install -c -D -m 644 modprobe.conf.sample /etc/modprobe.d/dahdi
/usr/bin/install -c -D -m 644 blacklist.sample /etc/modprobe.d/dahdi.blacklist
install -D ifup-hdlc /etc/sysconfig/network-scripts/ifup-hdlc
/sbin/chkconfig --add dahdi
DAHDI has been configured.

List of detected DAHDI devices:

pci:0000:00:09.0      wctdm-      e159:0001 Wildcard TDM400P REV E/F

run 'dahdi_genconf modules' to load support for only
the DAHDI hardware installed in this system. By
default support for all DAHDI hardware is loaded at
DAHDI start.
[root@dhcpc1 dahdi-tools-2.2.0]#
```

Gráfico III.23: Instalación Satisfactoria de Dahdi Tools

### 3.5.2.3 DETECCIÓN DE LA TARJETA

Para detectar el tipo de tarjeta que se usa, al mismo tiempo para ver si la reconoce, se debe realizar lo siguiente: **# dahdi\_hardware**



```
root@dhcpc1:~
Archivo Editar Ver Terminal Solapas Ayuda
Notice: Configuration file is /etc/dahdi/system.conf
line 0: Unable to open master device '/dev/dahdi/ctl'

1 error(s) detected

[root@dhcpc1 ~]# service dahdi start
Loading DAHDI hardware modules:
wct4xsp: [ OK ]
wctel2xp: [ OK ]
wctl1xp: [ OK ]
wctel1xp: [ OK ]
wctdm24xsp: [ OK ]
wcfxo: [ OK ]
wctdm: [ OK ]
wcb4xsp: [ OK ]
wctc4xsp: [ OK ]
xpp_usb: [ OK ]

Running dahdi_cfg: [ OK ]
[root@dhcpc1 ~]# dahdi_genconf
[root@dhcpc1 ~]# dahdi_cfg
[root@dhcpc1 ~]# dahdi_hardware
pci:0000:00:09.0      wctdm+      e159:0001 Wildcard TDM400P REV E/F
[root@dhcpc1 ~]#
```

Gráfico III.24: Reconocimiento de la Tarjeta OpenVox A400P

### 3.5.2.4 DAHDI START/STOP

Las herramientas Dahdi instalan un script de arranque, las cuales permiten observar el buen funcionamiento de las herramientas y de la tarjeta instalada en el equipo, como se muestra en `/etc/init.d/dahdi`. También puede utilizar esta secuencia de comandos para controlar Dahdi desde la línea de comandos de Linux:

- Inicia el servicio  
**# /etc/init.d/dahdi start o # service dahdi start**
- Reinicia el servicio  
**# /etc/init.d/dahdi restart o # service dahdi restart**
- Detiene el servicio  
**# /etc/init.d/dahdi stop o # service dahdi stop**

### 3.5.3 INSTALACIÓN Y CONFIGURACIÓN DE MODULO ASTERISK

Asterisk es el punto de enlace entre las diferentes redes de acceso a la aplicación telefónica, su función principal es la administración de las llamadas entrantes al sistema e interactuar con el modulo VoiceXML.

#### 3.5.3.1 INSTALACIÓN DE ASTERISK

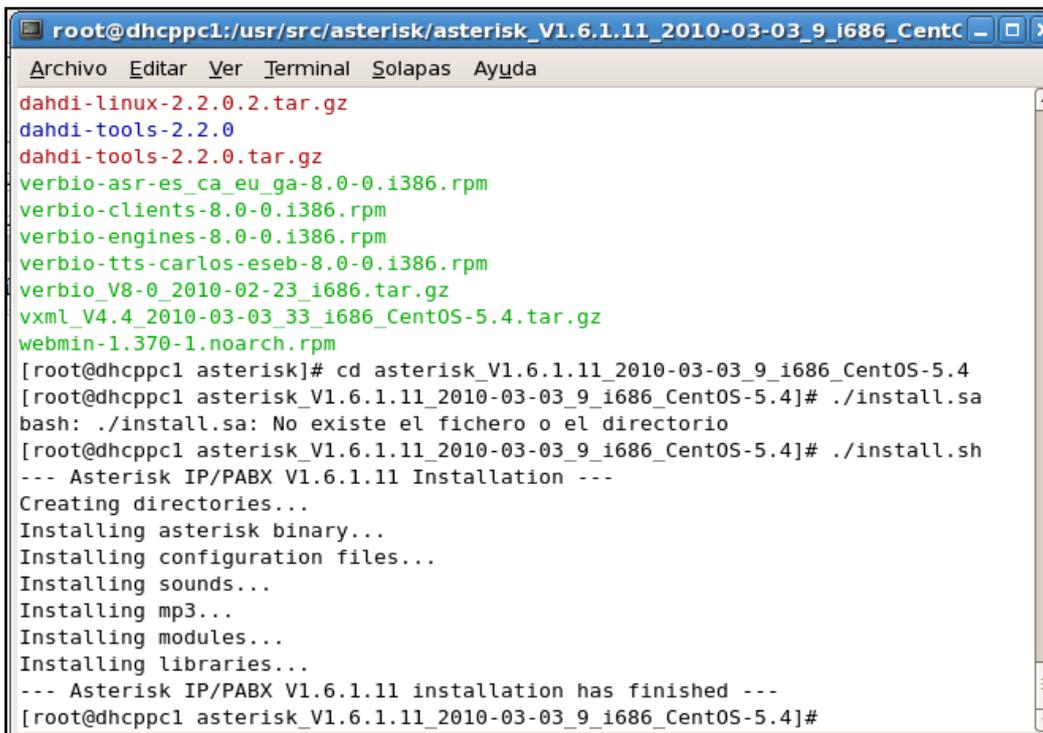
Se requieren los siguientes pasos:

```
# tar -xzf asterisk_V1.6.1.11_2010-03-03_9_i686_CentOS-5.4.tar.gz
```

```
# cd asterisk_V1.6.1.11_2010-03-03_9_i686_CentOS-5.4
```

```
# ./install.sh
```

RESULTADO:



```
root@dhcpc1:/usr/src/asterisk/asterisk_V1.6.1.11_2010-03-03_9_i686_CentC
Archivo Editar Ver Terminal Solapas Ayuda
dahdi-linux-2.2.0.2.tar.gz
dahdi-tools-2.2.0
dahdi-tools-2.2.0.tar.gz
verbio-asr-es_ca_eu_ga-8.0-0.i386.rpm
verbio-clients-8.0-0.i386.rpm
verbio-engines-8.0-0.i386.rpm
verbio-tts-carlos-eseb-8.0-0.i386.rpm
verbio_V8-0_2010-02-23_i686.tar.gz
vxml_V4.4_2010-03-03_33_i686_CentOS-5.4.tar.gz
webmin-1.370-1.noarch.rpm
[root@dhcpc1 asterisk]# cd asterisk_V1.6.1.11_2010-03-03_9_i686_CentOS-5.4
[root@dhcpc1 asterisk_V1.6.1.11_2010-03-03_9_i686_CentOS-5.4]# ./install.sa
bash: ./install.sa: No existe el fichero o el directorio
[root@dhcpc1 asterisk_V1.6.1.11_2010-03-03_9_i686_CentOS-5.4]# ./install.sh
--- Asterisk IP/PABX V1.6.1.11 Installation ---
Creating directories...
Installing asterisk binary...
Installing configuration files...
Installing sounds...
Installing mp3...
Installing modules...
Installing libraries...
--- Asterisk IP/PABX V1.6.1.11 installation has finished ---
[root@dhcpc1 asterisk_V1.6.1.11_2010-03-03_9_i686_CentOS-5.4]#
```

Gráfico III.25: Instalación correcta de Asterisk

### 3.5.3.2 COMPROBACIÓN DEL PROGRAMA DE INSTALACIÓN DE ASTERISK

Asterisk Inicio en el servidor mediante el siguiente comando:

```
#service asterisk start
```

```
#asterisk -r
```

Detener Asterisk con este comando:

```
*CLI> stop now
```

### 3.5.3.3 ASTERISK START/STOP

Se puede usar la secuencia de comandos para control de Asterisk desde la línea de comandos de Linux:

- Inicia el servicio

```
# /etc/init.d/asterisk start    o    # service asterisk start
```

- Reinicia el servicio

```
# /etc/init.d/asterisk restart    o  # service asterisk restart
```

- Detiene el servicio

```
# /etc/init.d/asterisk stop      o  # service asterisk stop
```

### 3.5.4 INSTALACIÓN DE VXI\* VOICEXML BROWSER

Una vez que se haya terminado de instalar Asterisk, ahora se puede instalar VXI \*, el navegador VoiceXML. Se debe descargar el paquete del navegador VoiceXML (vxml\_Vx.x\_date.tar.gz) de sitio web <http://www.i6net.com>.

#### 3.5.4.1 INSTALACIÓN DE PAQUETES VXI \*

```
#tar -xvzf vxml_V4.4_2010-03-03_33_i686_CentOS-5.4.tar.gz
```

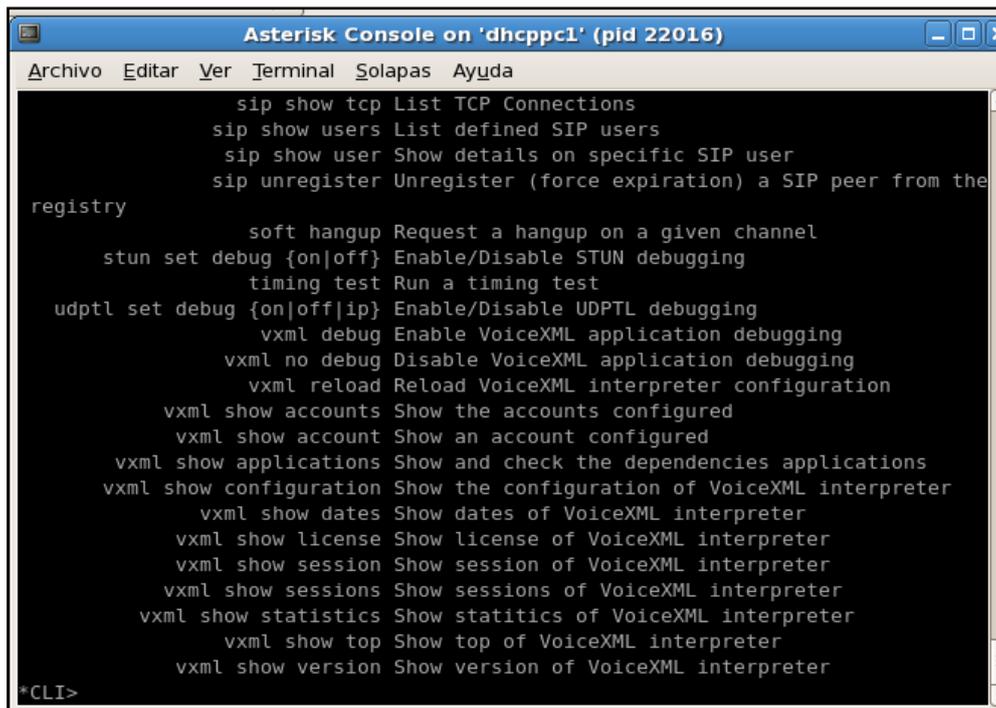
```
#cd vxml_V4.4_2010-03-03_33_i686_CentOS-5.4
```

```
#./install.sh
```

#### 3.5.4.2 COMPROBAR INSTALACIÓN

Por último, si la instalación está completa, se debe tener una aplicación más agregada (VXML). Esto se puede comprobar a partir Asterisk y utilizando el comando **“show application”**. Se puede acceder a la ayuda de línea de comandos introduciendo:

**\*CLI> help**



```
Asterisk Console on 'dhcppc1' (pid 22016)
Archivo  Editar  Ver  Terminal  Solapas  Ayuda

sip show tcp List TCP Connections
sip show users List defined SIP users
sip show user Show details on specific SIP user
sip unregister Unregister (force expiration) a SIP peer from the
registry

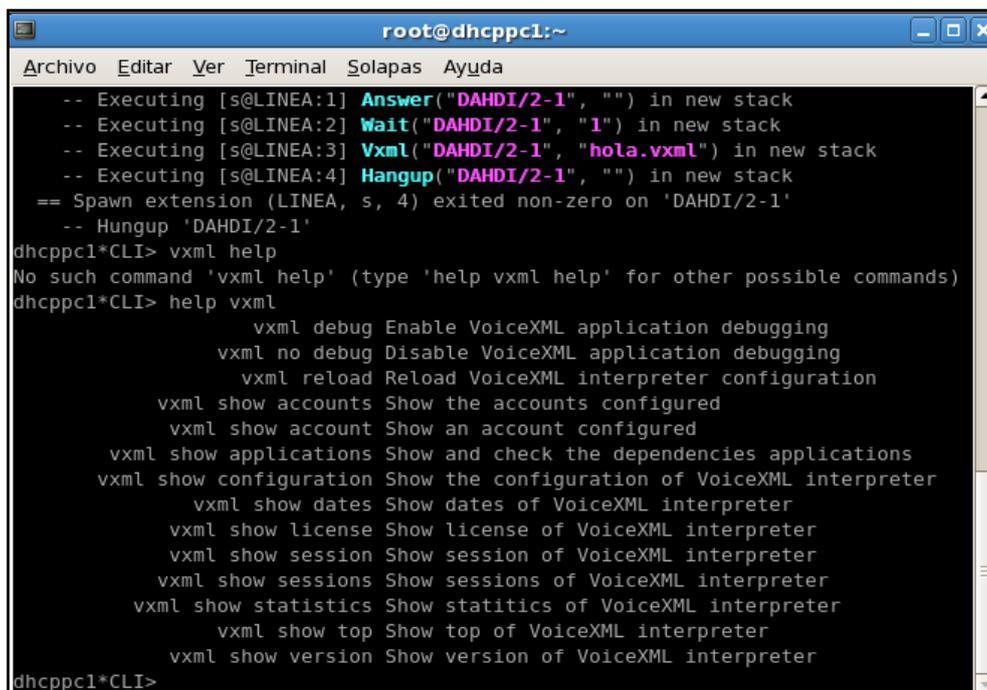
soft hangup Request a hangup on a given channel
stun set debug {on|off} Enable/Disable STUN debugging
timing test Run a timing test
udptl set debug {on|off|ip} Enable/Disable UDPTL debugging
vxml debug Enable VoiceXML application debugging
vxml no debug Disable VoiceXML application debugging
vxml reload Reload VoiceXML interpreter configuration
vxml show accounts Show the accounts configured
vxml show account Show an account configured
vxml show applications Show and check the dependencies applications
vxml show configuration Show the configuration of VoiceXML interpreter
vxml show dates Show dates of VoiceXML interpreter
vxml show license Show license of VoiceXML interpreter
vxml show session Show session of VoiceXML interpreter
vxml show sessions Show sessions of VoiceXML interpreter
vxml show statistics Show statistics of VoiceXML interpreter
vxml show top Show top of VoiceXML interpreter
vxml show version Show version of VoiceXML interpreter

*CLI>
```

Gráfico III.26: Comprobación de Instalación de VXI\*

### 3.5.4.3 AYUDA EN LÍNEA

En el gráfico II.6 se muestran los resultados de la ayuda de VXML, se puede acceder a través de la siguiente línea de comandos: **\*CLI> help vxml**



```
root@dhcppc1:~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda

-- Executing [s@LINEA:1] Answer("DAHDI/2-1", "") in new stack
-- Executing [s@LINEA:2] Wait("DAHDI/2-1", "1") in new stack
-- Executing [s@LINEA:3] Vxml("DAHDI/2-1", "hola.vxml") in new stack
-- Executing [s@LINEA:4] Hangup("DAHDI/2-1", "") in new stack
== Spawn extension (LINEA, s, 4) exited non-zero on 'DAHDI/2-1'
-- Hungup 'DAHDI/2-1'
dhcppc1*CLI> vxml help
No such command 'vxml help' (type 'help vxml help' for other possible commands)
dhcppc1*CLI> help vxml

vxml debug Enable VoiceXML application debugging
vxml no debug Disable VoiceXML application debugging
vxml reload Reload VoiceXML interpreter configuration
vxml show accounts Show the accounts configured
vxml show account Show an account configured
vxml show applications Show and check the dependencies applications
vxml show configuration Show the configuration of VoiceXML interpreter
vxml show dates Show dates of VoiceXML interpreter
vxml show license Show license of VoiceXML interpreter
vxml show session Show session of VoiceXML interpreter
vxml show sessions Show sessions of VoiceXML interpreter
vxml show statistics Show statistics of VoiceXML interpreter
vxml show top Show top of VoiceXML interpreter
vxml show version Show version of VoiceXML interpreter

dhcppc1*CLI>
```

Gráfico III.27: Ayuda en Línea

#### 3.5.4.4 ACTIVACIÓN DE LICENCIA COMERCIAL

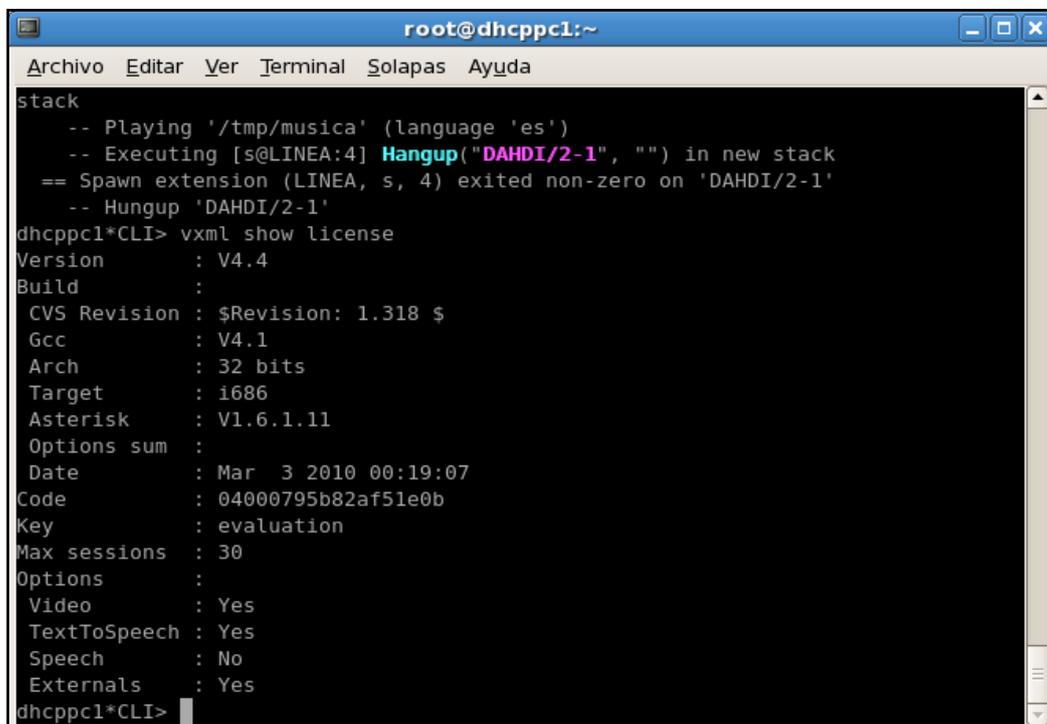
El navegador VoiceXML I6Net está protegido por una clave de licencia. La licencia está basada en host por lo que cada ordenador tiene una clave de licencia diferente. Si la licencia no está activada, puede ejecutar un solo período de sesión en VoiceXML (audio). Este es el modo de uso gratuito.

Para obtener una clave de licencia comercial válida del navegador VoiceXML (para su uso durante más de una período de sesiones), se debe ejecutar Asterisk y obtener su código, escriba los siguientes comandos:

**# asterisk -r**

**\*CLI> vxml show license**

Una vez que el código se muestra, se debe copiar el código y enviarlo por correo electrónico a support@i6net.com. Se debe incluir también información sobre la instalación de referencia, como un número de orden de compra o el contexto de la instalación. En el prototipo se utilizará solo una licencia de evaluación como se muestra a continuación.



```
root@dhcpccl:~
Archivo Editar Ver Terminal Solapas Ayuda
stack
-- Playing '/tmp/musica' (language 'es')
-- Executing [s@LINEA:4] Hangup("DAHDI/2-1", "") in new stack
== Spawn extension (LINEA, s, 4) exited non-zero on 'DAHDI/2-1'
-- Hungup 'DAHDI/2-1'
dhcpccl*CLI> vxml show license
Version      : V4.4
Build       :
CVS Revision : $Revision: 1.318 $
Gcc         : V4.1
Arch        : 32 bits
Target      : i686
Asterisk    : V1.6.1.11
Options sum :
Date        : Mar  3 2010 00:19:07
Code        : 04000795b82af51e0b
Key         : evaluation
Max sessions: 30
Options     :
Video       : Yes
TextToSpeech: Yes
Speech      : No
Externals   : Yes
dhcpccl*CLI>
```

Gráfico III.28: Licencia Comercial

#### **3.5.4.5 DESINTALACIÓN DE VXI\***

En el mismo directorio que el script install.sh, se encuentra la secuencia de comandos para borrar todos los componentes y archivos instalados.

```
# cd vxml_Vx.x_date_build
```

```
# ./uninstall.sh
```

#### **3.5.4.6 ACTUALIZACIÓN DE LA VERSION**

Si existe una nueva versión sobre una versión instalada anteriormente, el script de instalación conserva su archivo de configuración, vxml.conf. Se debe comprobar que el archivo de configuración nuevo, vxml.conf.sample, a fin de integrar algunos parámetros nuevos en vxml.conf. Verificar el archivo de configuración del intérprete VoiceXML, client.cfg también.

#### **3.5.4.7 CONFIGURACIÓN DE VXI \***

##### **Configuración del módulo de Asterisk**

El archivo de configuración es /etc/asterisk/vxml.conf, se lo debe modificar de acuerdo a los requerimientos.

```
# vi /etc/asterisk/vxml.conf
```



```
root@dhcpc1:~
; VoiceXML Configuration
;

[general]
autoanswer=yes
wavcodec=gsm
videosilence=
audiosilence=
dialformat=SIP/%s
speechprovider=unimrcp
speechscore=50

[control]
forward=#
reverse=*
stop=123456789
pause=
restart=0
skipms=5000

[license]
max=1
"/etc/asterisk/vxml.conf" 45L, 585C
[license]
max=1
video=yes
speech=emulation
texttospeech=yes
externals=yes
key=evaluation

[account0]
name=demos
number=*0
url=http://links.i6net.com/index.vxml
max=1
speech=no
dialformat=SIP/%s@myaccount

[account1]
name=audiovideo
number=1234
url=http://localhost/vxml/audio.vxml
urlvideo=http://localhost/vxml/video.vxml
max=1
speech=no
```

**Gráfico III.29: Archivo vxml.conf**

Se debe tener en cuenta que antes de configurar el módulo de Asterisk, se debe hacer una copia de seguridad de vxml.conf. El proceso de actualización de VXI\* debe mantener la configuración actual, pero algunas actualizaciones específicas pueden ser obligatorias para futuras versiones.

## Funciones Generales

La siguiente sección cubre las funciones generales de VXI \*.

- **debug = (0 / 1)**

El valor predeterminado es 0, significa que no hay depuración. Si la función se establece con un valor 1, la aplicación VoiceXML Asterisk generará depuración.

- **autoanswer= {yes/no}**

El valor por defecto es yes. Si se establece, la aplicación VoiceXML responderá el canal del asterisk antes de comenzar la sesión de VoiceXML.

- **autohangup = (yes / no)**

El valor predeterminado es no. Si se establece, la aplicación VoiceXML se cuelgue el canal del asterisk al final de la sesión de VoiceXML.

- **autoexit = (yes / no)**

El valor por defecto es yes. Si se establece, la aplicación VoiceXML forzará la salida (con valor de retorno -1) de Asterisk, si la conexión con OpenVXI se pierde.

- **defaulttimeout = (1 .. 60)**

Usted puede establecer el valor predeterminado de tiempo de espera para permitir el reconocimiento DTMF, en milisegundos (ms). El valor por defecto es 5 segundos.

- **defaultinterdigittimeout = (1)**

El valor predeterminado de tiempo de espera interdigit se puede configurar para permitir el reconocimiento DTMF, si no se establece la sintaxis VoiceXML.

- **wavcode = (pcm/gcm)**

Esta función le permite configurar el codec por defecto para el registro asociado al tipo MIME "Audio / x-wav". Puede seleccionar GSM o PCM. Si bien el codec por defecto es PCM, los archivos de audio grabados podrían ser demasiado grandes para ser fácilmente colocado en un servidor HTTP.

- **audiosilence = (filename)**

El navegador VoiceXML puede circular y generar un sonido durante las fases de interacción. Se debe introducir el archivo de sonido especificado en formato PCM (deberá ser omitido la extensión del fichero). Los archivos de sonido se almacenan en la /Var/lib/asterisk/sounds directorio por defecto, sin embargo, la ruta del directorio se puede cambiar en asterisk.conf. El valor predeterminado es sin nombre de archivo. Esta función está deshabilitada en esta versión.

- **videosilence = (filename)**

El navegador VoiceXML puede circular y generar un clip de vídeo y audio. El valor predeterminado es sin nombre de archivo. Esta función está deshabilitada en esta versión.

- **recordsilence = (yes / no)**

El navegador VoiceXML puede generar un silencio de audio (paquete RTP generados), durante la grabación. Esta característica se ha agregado para evitar pasarela obsesivas, si el servidor no genera paquetes durante un período configurado. Si el modo de vídeo está activado, es preferible que haya decidido eliminar esta opción para generar un eco de vídeo durante la fase de grabación (como un espejo). El valor predeterminado es no.

- **threshold= {1...32767}**

Este valor calculado por el promedio de todas las muestras dentro de un marco, por un marco que está decidido a ser el silencio (por debajo del threshold) o ruido (por encima del threshold). Este parámetro se utiliza para detectar el silencio final de registro. El valor es por defecto es 256.

- **dialformat= { parámetros de la aplicación con %s}**

Esta es una cadena para especificar la interfaz y el para que se ha elegido para la transferencia. El "%s" será sustituido por la cadena situado en el atributo dest<transfer>. Recuerde que el prefijo el valor dest con "tel:" generado para la función transfer. Prefijos se han añadido para que coincida con algunas de las

funciones de Asterisk, como conferencia, llamar a una aplicación, etc. El valor predeterminado es SIP/% s.

- **dialformatvideo = (parámetros de la aplicación con% s)**

Esta función es como dialformat, pero para una transacción de vídeo. El valor predeterminado es SIP/% s.

- **blindapplication = (nombre de la aplicación asterisk o vacío)**

Esta es un string para especificar la aplicación utilizada con el modo blind transfer. Si la aplicación está vacía, la aplicación VoiceXML generará un comando de marcación con los parámetros de transmisión después del período de sesiones de VoiceXML. El valor por defecto es "transfer".

- **video = (yes / no)**

Activar o desactivar la función de vídeo (controlado por la clave de licencia). El valor predeterminado es no.

- **removeprompts = (yes / no)**

El intérprete VoiceXML genera dinámicamente los archivos que se le pregunte en el directorio /tmp. Después que requiera su intervención, usted puede optar por eliminar o mantener los archivos, ya que pueden ser sobrescritos por las nuevas secciones<prompt>. El valor por defecto es sí.

- **autoreloadconfig = (yes / no)**

Utilice esta función para actualizar automáticamente la configuración cuando el archivo se modifica. La actualización se basado en la fecha de archivo de configuración o puede iniciar una actualización del archivo en cualquier momento por su cuenta. Se debe tener cuidado de no guardar los archivos de configuración incompleta. El valor predeterminado es no.

- **speech= {yes/no/emulation/automatic}**

Usted puede activar o desactivar el ASR (reconocimiento de voz) con la aplicación VoiceXML. "Sí" significa que el recurso de ASR se asignará durante la duración del período de sesiones VoiceXML.

- El valor "**emulation**" es un modo permisivo que siempre devuelve bien, incluso si el ASR no está presente.
- El valor "**automatic**" distribuirá los recursos del habla si la creación es una gramática planteada por el intérprete de VoiceXML. El recurso del habla se dará a conocer después de todas las gramáticas son liberados. El valor predeterminado es no.

- **speechprovider = (LumenVox / verbio)**

Puede elegir el proveedor de reconocimiento de voz para asignar al recurso del habla.

- **speechscore = (0 ... 100)**

La función speechscore le permite establecer la puntuación de la confianza de que el motor de reconocimiento de voz (ASR). El número predeterminado es 50. Cuando la aplicación VoiceXML Asterisk consigue un resultado ASR, la puntuación se devuelve con la palabra o frase reconocida. Si la puntuación es baja, este valor limitador se utiliza para devolver un error de no coincidencia evento en lugar un resultado incorrecto.

- **speechforcedscore = (0 .. 100)**

Fuerza de la puntuación valor devuelto a la interprete VoiceXML. El valor predeterminado es 0 (para desactivar el forzamiento).

- **speechdirectory = (directorio de archivos)**

Es preferible no establecer este parámetro y dejar que la aplicación VoiceXML obtenga los valores por defecto.

- Para el Verbio ASR, el valor por defecto es "" (uso interno de órdenes en Verbio).
- Para el ASR LumenVox, el valor por defecto es `"/var/lib/openvxi/grammars/ABNF_%s_%s.gram"`

- **cdrupdate = (yes / no)**

Si el valor se establece en yes, al final de la sesión de VoiceXML, los CDR actualizan la información de la aplicación VoiceXML. La información del CDR generada por cada llamada se da con el fin de obtener estadísticas de los datos. Después de la actualización al final del período de sesiones VoiceXML, el CDR será bloqueado. El valor por defecto es yes.

- **cdrdial = (yes / no)**

Si el valor está establecido en "yes", la aplicación de línea utilizada para la <transfer> generará un nuevo conjunto de CDR. Si se pone un "No", no se generará el CDR y los CDR actual serán guardados. El valor por defecto es "sí"

- **cdrprompt = (yes / no / all)**

Si el valor está establecido en "yes" y la propiedad cdrprompt se establece en "true" en el contexto VoiceXML, el de audio/video pide generar un nuevo CDR (la duración es la duración del sistema, DTMF indican el símbolo del sistema que se ha omitido por un DTMF, y colgar si el mensaje es ignorado).

- Si se define "no", se generará CDR (valor por defecto).
- Si el valor está ajustado "all", todo los clips de audio/video generará CDRs. El valor por defecto es "no". El alias promptcdr es también soportado (en vxml.conf y para VoiceXML).

- **cdrconference= {yes/no/all}**

Si el valor está establecido en "yes" y la propiedad cdrconference se establece en "true" en el contexto VoiceXML, la transferencias de conferencias generará un nuevo CDR (la duración es la duración de conferencia de la sesión de los usuarios,

DTMF indican la conferencia han sido omitidos por un DTMF, y colgar si la conferencia se omite).

- Si se pone a "no", no se generará CDR (valor por defecto).
- Si el valor está ajustado a "all", todas las sesiones de la conferencia se generan CDRs. El valor por defecto es "no". El alias `conferencecdr` es también soportado (en el `vxml.conf` y para VoiceXML).

- **`cdroverwrite= {yes/no}`**

Si el valor está establecido en "yes", el campo de origen y destino del RDC se sobrescribe con el contenido de la variables `VXML_REMOTE` y `VXML_LOCAL`. El valor por defecto es "no".

### 3.5.4.8 FUNCIONES DE CONTROL

La sección de control se describe las funciones de control que le permiten configurar comandos DTMF, escuchar un archivo con el avance rápido, pausa, retroceso, reiniciar, etc. Para habilitar un control durante un sistema, debe introducir el nombre de la propiedad VoiceXML a la igualdad de "name=control y value=yes".

```
<property name="control" value="yes" />
```

Esto sólo funciona con las redes GSM y archivos WAV - en los demás casos la propiedad se ignora.

- **`forward= {DTMF(s)/empty}`**

Fast-forward cuando este dígito DTMF es recibido. El valor por defecto es #.

- **`reverse= {DTMF(s)/empty}`**

Reinicia cuando este dígitos DTMF se recibe. El valor por defecto es \*.

- **`stop= {DTMF(s)/empty}`**

Para cuando el dígito DTMF es recibido. El valor por defecto es 0123456789.

- **`pause= {DTMF(s)/empty}`**

Pausa playback cuando el dígito DTMF es recibido. El valor por defecto es empty.

- **`restart= {DTMF(s)/empty}`**

Reinicia el playback cuando el dígito DTMF es recibido. El valor por defecto es **empty**.

- **skipms= {retardo en milisegundos}**

Este es el número de milisegundos para saltar al reiniciar o avanzar rápidamente. El valor por defecto es 5000.

#### **3.5.4.9 FUNCIONES DE LA CUENTA**

Las propiedades de la cuenta se definen dentro de las sesiones. Cada cuenta tiene una sesión específica. En las sesiones son identificados y numerados de account1 a account99. El siguiente ejemplo muestra las funciones de cada sesión cuenta.

**[account0]**

**name=demos**

**number=\*0**

**url=http://links.i6net.com/index.vxml**

**max=1**

**speech=no**

**dialformat=SIP/%s@myaccount**

- **name = (string)**

Esta función indica el nombre o denominación de la cuenta. Esta referencia puede ser utilizada para identificar la cuenta utilizada por la aplicación VXML. Esto le permite ejecutar un período de sesiones VoiceXML correspondientes a los valores de cuenta.

- **number= {called number}**

Esta función aloja la identificación de una cuenta con el número de llamada, la información de señalización desde la función ANI, CALLERID(). Se puede usar el "\*", el carácter comodín, para especificar un substring tales como \* 03 o \* 014612, que la función número contendrá. Puede iniciar el número con "\_", para usar Asterisk Dialplan patrones también (ver: <http://www.voip-info.org/wiki/view/Asterisk+Dialplan+Patterns>). La función también puede ser utilizada para identificar la cuenta, como el nombre, VXML(número de la cuenta). Se puede empezar con el número "@" para que coincida el número de llamadas con el número de cuenta (utilizar la misma la sintaxis como el número llamado).

- **url= {voicexml URL}**

Esta función indica la dirección de la cuenta de VoiceXML.

- **urlvideo= {voicexml URL}**

Esta función define la URL de vídeo VoiceXML de la cuenta. La llamada se identifica como una llamada de video por los codecs indicados o la función CHANNEL(transfercapability)=VIDEO.

- **Max = (0 ... 120)**

Esto indica el número máximo de sesiones permitidas para esta cuenta. Si no hay suficientes sesiones a continuación la aplicación VoiceXML generará un error.

- **dialformat= {application(]/%s[)}**

Esto es similar a la función general, pero para la cuenta solamente. Si no se establece el valor de uso general.

- **dialformatvideo= {application(]/%s[)}**

Esta función tiene el mismo objetivo que el dialformat, sólo para las sesiones de vídeo.

- **force= {video/audio}**

Si se establece en vídeo, pone el Transfercapability en VIDEO (y permitir el procesamiento h324m). Si se establece en audio, la aplicación VXML ejecuta directamente la cuenta URL y evitar la ejecución redirección (parámetro con '@').

- **speech= {yes/no/emulation/automatic}**

Esta función habla como función general, pero para la cuenta solamente. Si no se establece el uso general valor.

- **speechprovider= {lumenvox/verbio}**

Esta función habla es como la función general, pero para la cuenta solamente. Si no se establece el uso general valor.

- **speechscore = (0 .. 100)**

Esta función habla como una función general, pero para la cuenta solamente. Si no se establece el uso general valor.

- **speechforcedscore = (0 .. 100)**

Esta función habla como una función general, pero para la cuenta solamente. Si no se establece el uso general valor.

- **mark= {string/@local/@remote/@id/@param}**

Establece un marcado de string en las trazes del navegador VoiceXML. El identificador de sesión y una cadena que se deben añadir al canal de la columna número (3) en las trazes (Ejemplo:... |33|...→... |33\_1\_user1 | ...).

Cuatro redirección existe:

**@remote : caller number**

**@local : called number**

**@id : VoieXML id parameter value**

**@param : VoiceXML parameter value**

- **durationlimit= {0...}**

Esto indica la duración máxima en segundos de la sesión con esta cuenta. La llamada se colgará después de esta duración máxima.

#### **3.5.4.10 VXI\* START / STOP**

El software del navegador VoiceXML se instala en /usr/sbin y /usr/lib /openvxi. El navegador VoiceXML script de instalación en Linux es /etc/init.d/openvxi. La secuencia de comandos openvxi llama al directorio /usr/sbin/ safe\_openvxi ejecutable que funciona como un monitor y un cargador de auto para su sistema navegador VoiceXML. Esta safe\_openvxi se inicia el navegador VoiceXML y monitores para asegurarse de que todavía se está ejecutando. Si el proceso de VoiceXML browser muere, el script intentará reiniciar.

Para iniciar o detener el navegador VoiceXML, utiliza los comandos "start openvxi" y "stop openvxi":

**#/etc/init.d/openvxi start o #service openvxi start**

**#/etc/init.d/openvxi stop o #service openvxi stop**

**#/etc/init.d/openvxi restart o #service openvxi restart**

#### **3.5.4.11 USANDO VXI\***

El navegador VoiceXML instala la aplicación Asterisk app\_vxml que utiliza el navegador para ejecutar.

### 3.5.4.12 PAGINAS VOICEXML

Se debe comprobar el módulo de la aplicación de VXML. Para obtener más información sobre aplicaciones, simplemente escriba "**show application**" en el prompt del CLI de Asterisk. Para ver los detalles de cómo se utiliza esa aplicación en particular en este archivo (el plan de Asterisk Dial), se debe escribir:

**\*CLI> show application <command>**

Para la aplicación vxml:

**\*CLI> show application vxml**

### 3.5.4.13 USO DE LA SINTAXIS

**Descripción:** La URL puede ser ingresada de diferentes maneras:

- Se debe pasar la URL como parámetros de aplicación.  
Ejemplo: vxml(file:///tmp/test.vxml)
- Se debe pasar el nombre de cuenta o el número de cuenta.  
Ejemplo: vxml(test)
- Pase el "@" para que coincida con el número de llamadas al número de cuenta.
- Establecer la variable VXML\_URL antes de ejecutar la aplicación VXML.
- No se pasa ningún parámetro, y se debe utilizar la sección de configuración de cuentas. Si el número de cuenta coincide con el número llamado, la URL(s) y los parámetros se deben utilizar.

A continuación se describe cómo ejecutar una sesión de VoiceXML.

#### Sintaxis

**VXML ([URL | Nombre | Número])**

#### Variables

- Si la variable VXML\_URL ha sido establecida el VXML lo ejecuta, el valor de esa variable se utilizará para la URL a menos que el parámetro no está establecido en la solicitud.
- Si la variable VXML\_ID ha sido establecida VXML lo ejecuta, la sesión de VoiceXML ID variable llamada "Telephone.id" se establece con este valor (ejecución del contexto de sesión en VoiceXML).

- Si la variable VXML\_PARAM (VXML\_AAI es un alias) se establece cuando se ejecuta VXML, el valor de esa variable se utiliza como "telephone.param" (ejecución del contexto de sesión en VoiceXML) y session.connection.aai. Después de la ejecución, el resultado de la etiqueta <EXIT> VoiceXML y la propiedad 'expr' son accesibles por la variable VXML\_RESULT.

#### 3.5.4.14 COMANDOS DE ADMINISTRACIÓN

Las siguientes entradas son los comandos del CLI de Asterisk para el navegador VoiceXML.

- **vxml debug**

Habilita la depuración de VoiceXML para la aplicación Asterisk.

- **vxml debug interpreter all**

Habilita la depuración de VoiceXML para la aplicación intérprete ().

- **vxml no debug**

Este comando desactiva la depuración de VoiceXML para la aplicación Asterisk.

- **vxml no interpreter debug**

Este comando desactiva la depuración de VoiceXML para la aplicación intérprete.

- **vxml show license**

Este comando es usado para presentar la información de licencia.

- **vxml show configuration**

Este comando es usado para mostrar el resumen de configuración del intérprete VoiceXML.

#### 3.5.5 CONFIGURACIÓN DE LA PBX

A continuación se muestra la configuración de la PBX.

- Cargar y reconocer el módulo de tarjeta

- #service dahdi start**

- #dahdi\_genconf**

- #dahdi\_cfg**

- dahdi start: Cargar el módulo de tarjeta
  - dahdi\_genconf: Reconoce el tipo de tarjeta instalado y crea/modifica dos archivos: */etc/dahdi/system.conf* y */etc/asterisk/dahdi-channels.conf*

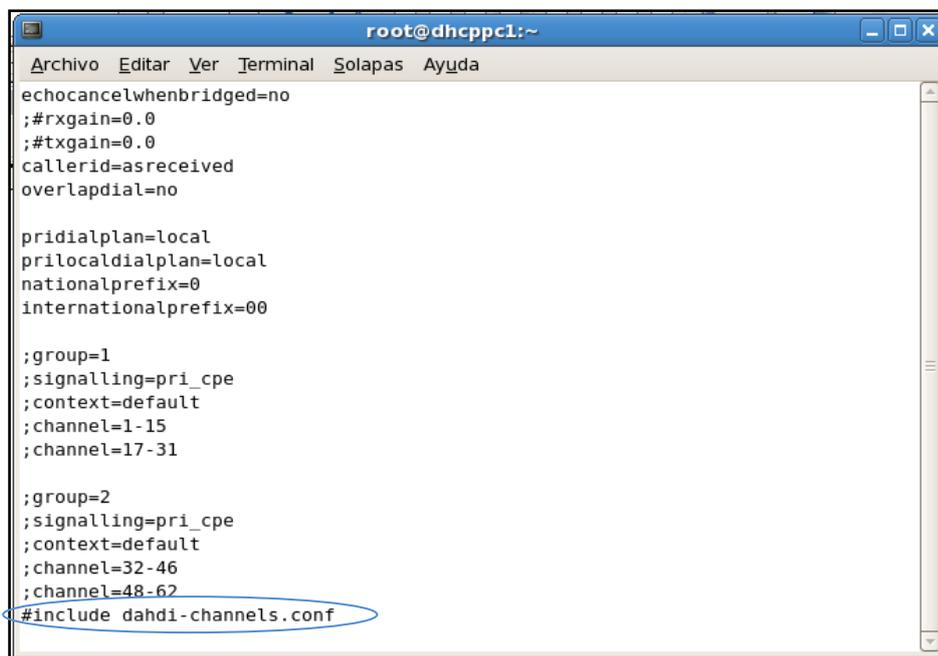
- dahdi\_cfg: verifica que el archivo de configuración está correcto, y por consecuencia muestra los canales bien configurados.
- Configurar en el archivo chan\_dahdi.conf que se encuentra en el path /etc/asterisk/chan\_dahdi.conf, agregar la siguiente línea de código al final del archivo chan\_dahdi.conf:

**#include dahdi-channels.conf**

- Ingresamos al archivo:

**#vi /etc/asterisk/chan\_dahdi.conf**

Como se muestra en el gráfico III.30:



```
root@dhcpcl:~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
echocancelwhenbridged=no
;#rxgain=0.0
;#txgain=0.0
callerid=asreceived
overlapdial=no

pridialplan=local
prilocaldialplan=local
nationalprefix=0
internationalprefix=00

;group=1
;signalling=pri_cpe
;context=default
;channel=1-15
;channel=17-31

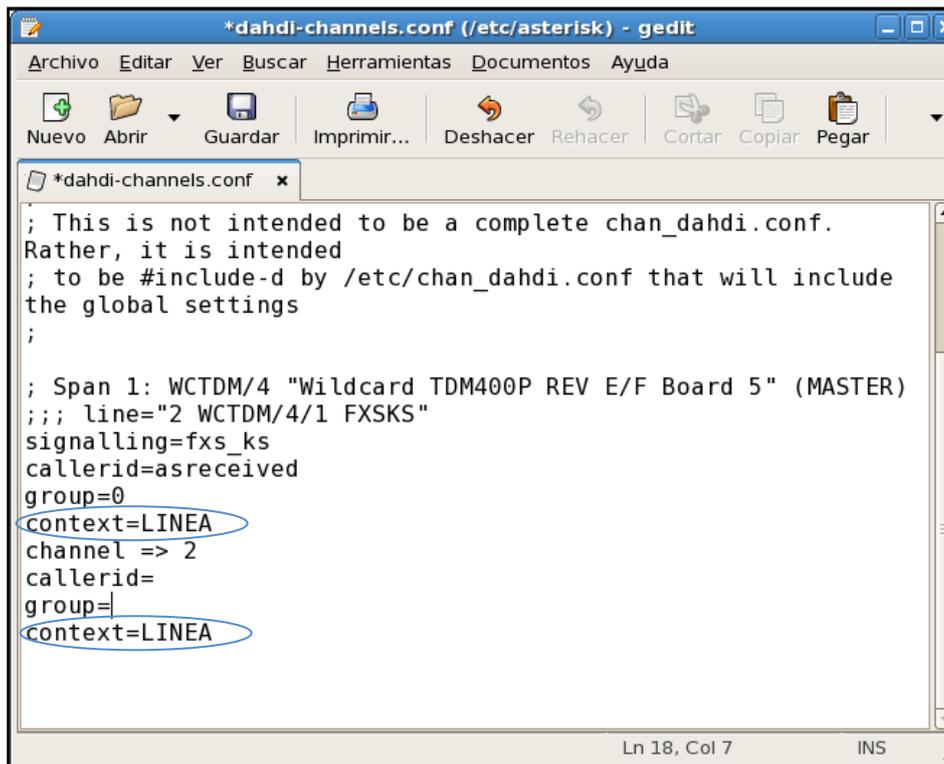
;group=2
;signalling=pri_cpe
;context=default
;channel=32-46
;channel=48-62
#include dahdi-channels.conf
```

**Gráfico III.30: Archivo chan\_dahdi.conf**

- Modificar el archivo dahdi-channels.conf que se encuentra en el path /etc/asterisk/dahdi-channels.conf:

**#vi /etc/asterisk/dahdi-channels.conf**

Donde modificamos el “**context**”, el cual debe llevar un mismo nombre como se muestra a continuación:



```
*dahdi-channels.conf (/etc/asterisk) - gedit
Archivo Editar Ver Buscar Herramientas Documentos Ayuda
Nuevo Abrir Guardar Imprimir... Deshacer Rehacer Cortar Copiar Pegar
*dahdi-channels.conf x
; This is not intended to be a complete chan_dahdi.conf.
Rather, it is intended
; to be #include-d by /etc/chan_dahdi.conf that will include
the global settings
;
; Span 1: WCTDM/4 "Wildcard TDM400P REV E/F Board 5" (MASTER)
;;; line="2 WCTDM/4/1 FXSKS"
signalling=fxs_ks
callerid=asreceived
group=0
context=LINEA
channel => 2
callerid=
group=
context=LINEA
Ln 18, Col 7 INS
```

**Gráfico III.31: Archivo dahdi-channels.conf**

- Configurar el archivo extensions.conf el cual se encuentra en el path /etc/asterisk/extensions.conf. Donde probaremos la funcionalidad del primer archivo vxml desde la PSTN.

```
[LINEA]
;FXO
exten => s,1,Answer()
exten => s,n,Wait(1)
exten => s,n,Vxml(file:///temp/hola.vxml)
exten => s,n,Hangup()
```

**Espacio Código III.20: Archivo extensions.conf**

- El archivo hola.vxml se encuentra en el directorio /temp, como se muestra en el Espacio Código III.21.

```
<?xml version="1.0"?>
<vxml version = "2.0" xmlns="http://www.w3.org/2001/vxml">
<form>
  <block>
    <audio src="musica.wav"/>
  </block>
</form>
</vxml>
```

**Espacio Código III.21: hola.vxml**

### 3.5.6 INSTALACIÓN Y CONFIGURACIÓN DE LA PLATAFORMA DE PRUEBAS

Los módulos que se detallan a continuación son los componentes más importantes de la plataforma de pruebas, a continuación se muestran las herramientas a instalar:

- verbio-clients-8.0-0.i386.rpm
- verbio-engines-8.0-0.i386.rpm
- verbio-tts-carlos-eseb-8.0-0.i386.rpm
- verbio-asr-es\_ca\_eu\_ga-8.0-0.i386.rpm

#### 3.5.6.1 INSTALACIÓN DEL MODULO VERBIO-ENGINES-8.0-0.I386.RPM

Este paquete es de obligada instalación cuando estemos montando el servidor de síntesis/reconocimiento.

El siguiente comando permite instalar el paquete verbio-engines.

```
# rpm -ivh verbio-engines-8.0-0.i386.rpm
```

#### 3.5.6.2 INSTALACIÓN DEL MODULO VERBIO-CLIENTS-8.0-0.I386.RPM

Contiene las librerías, ejemplos y documentación necesarios, para el desarrollo de aplicaciones de voz.

```
# rpm -ivh --force verbio-clients-8.0-0.i386.rpm
```

### 3.5.6.3 INSTALACIÓN DEL MODULO VERBIO-TTS-CARLOS-ESEB-8.0-0.I386.RPM

Se debe instalar los paquetes que correspondan, según los locutores que necesitemos, en este caso utilizaremos la voz en castellano “Carlos”.

```
#rpm -ivh -- verbo-tts-carlos-eseb-8.0-0.i386.rpm
```

El paquete TTS instala el script HTTP en: /var/www/tts/verbio.

### 3.5.6.4 INSTALACIÓN DEL MODULO VERBIO-ASR-ES\_CA\_EU\_GA-8.0-0.I386.RPM

Se debe instalar los paquetes que correspondan, según el lenguaje que necesitemos reconocer en nuestro caso el motor de reconocimiento verbo-asr para reconocimiento en español.

```
#rpm -ivh --force verbo-asr-es_ca_eu_ga--8.0-0.i386.rpm
```

### 3.5.6.5 INSTALACIÓN DEL MODULO VERBIO\_V8-0\_2010-02-23\_I686.tar

El presente modulo contiene los conectores que permiten transformar las ondas que reproducen la voz a partir del modulo verbo-tts. Estos conectores están realizados en lenguaje PHP. En nuestro caso generaremos archivos de audio con extensión wav.

```
#tar -xzvf verbo_V8-0_2010-02-23_i686.tar.gz
```

```
#cd verbo_V8-0_2010-02-23_i686
```

```
#./install.sh
```

### 3.5.6.6 CONFIGURACIÓN

Para verificar el modulo fue correctamente ubicamos se realiza lo siguiente:

```
#verbiod -e
```

```
#ps ef | grep verbo
```

Como resultado se muestra en el espacio de código II.22.

```
root 6530 1 27 17:31 ? 00:00:09 verbiod -e
root 6538 6304 0 17:32 pts/2 00:00:00 grep verbo
```

Espacio Código III.22: Verificación de Instalación de Verbio

### 3.5.6.7 CONFIGURACIÓN DE TEXT TO SPEECH

El navegador VoiceXML integra una interfaz de cliente HTTP para conectarse de un texto HTTP a voz (TTS) de servidor. Esto permite que generen de forma dinámica el contenido de audio con un motor de texto a voz. La ventaja sin embargo, de utilizar el conector HTTP es que el “speech” generado se almacena en caché por el navegador VoiceXML, y volver a utilizar la próxima vez. El texto es publicado a través de una solicitud HTTP, el servidor responde con un archivo wav estándar. Los usuarios pueden utilizar el TTS para generar menús que soliciten, por ejemplo, sin necesidad de comprar una gran cantidad de licencias TTS.

- Para ello ingresamos al archivo client.cfg, que se encuentra ubicado en /etc/openvxi/client.cfg:

**#vi /etc/openvxi/client.cfg**

- Y en el archivo se debe ubicar en la parte de configuración del servidor tts y modificar ciertos parámetros, de acuerdo a los requerimientos. Inicialmente el archivo se presentará sin modificaciones. Ver Gráfico III.32.

```
#####  
# TTS server configuration #  
#####  
  
#client.prompt.resource.0.uri          VXIString  http://localhost/tts/verb/tts.php  
#client.prompt.resource.0.uriVideo    VXIString  http://localhost/tts/video/ttv.php  
client.prompt.resource.0.method       VXIString  POST  
client.prompt.resource.0.cacheDir     VXIString  /tmp/cacheContent  
client.prompt.resource.0.format       VXIString  wav  
client.prompt.resource.0.formatVideo  VXIString  3gp  
client.prompt.resource.0.maxage       VXIInteger -1  
client.prompt.resource.0.checkBreak   VXIInteger 0  
client.prompt.resource.0.cutPrompt    VXIInteger 0
```

**Gráfico III.32: Configuración del Servidor TTS Original**

- Y con los cambios se puede ver la configuración lista para el desarrollo de nuestro tts. En este caso se desarrolla la modificación de 4 parámetros que se describen a continuación:

- Client.prompt.resource.0.uri -> el cual del archivo original se lo debe descomentar para poder concatenar con el archivo tts.php
- Client.prompt.resource.0.format -> se lo cambia debido a que alaw es un archivo de 8kHz 8-bits, en cambio el wav es 8kHz 16-bits.
- Client.prompt.resource.0.checkBreak ->se modifica por 1

```
#####  
# TTS server configuration #  
#####  
client.prompt.resource.0.uri          VXIString    http://localhost/tts/verbio/tts.php  
#client.prompt.resource.0.uriVideo    VXIString    http://localhost/tts/video/ttv.php  
client.prompt.resource.0.method       VXIString    POST  
client.prompt.resource.0.cacheDir     VXIString    /tmp/cacheContent  
client.prompt.resource.0.format       VXIString    alaw  
client.prompt.resource.0.formatVideo  VXIString    3gp  
client.prompt.resource.0.maxage       VXIInteger   -1  
client.prompt.resource.0.checkBreak   VXIInteger   1|  
client.prompt.resource.0.cutPrompt    VXIInteger   0
```

Gráfico III.33: Configuración del Servidor TTS Configurado

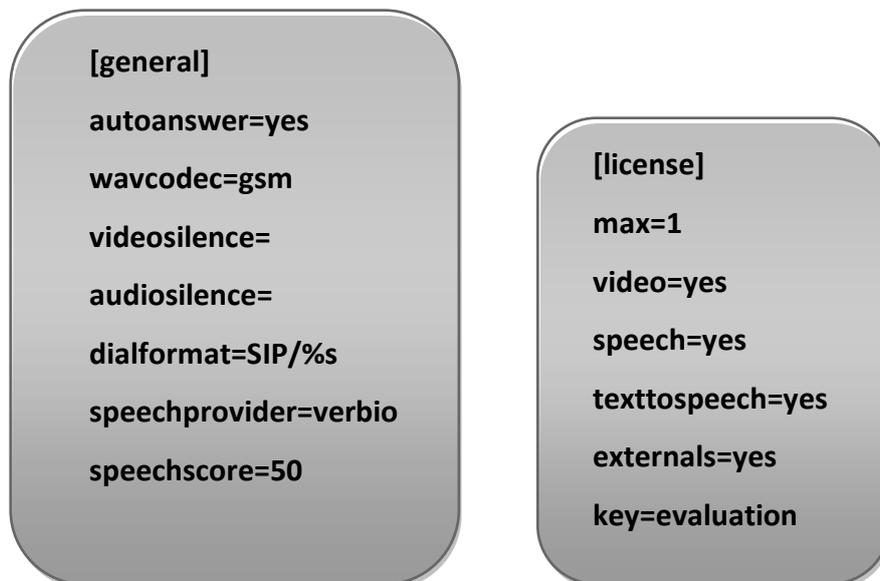
### 3.5.6.8 CONFIGURACIÓN DEL SERVIDOR AUTOMATIC SPEECH RECOGNITION

Para habilitar el reconocimiento de voz, cambiar el parámetro principal de voz. El parámetro "speech" pueden tomar tres valores, "yes", "automatic", "no" o "emulation"(no generan errores si se habilita gramáticas del habla).

- Y se debe configurar de la siguiente forma:

**#vi /etc/asterisk/vxml.conf**

- Donde se debe modificar el **speechprovider=verbio** y **speech=yes**, con el cual se especifica el motor asr que se va utilizar y que se permite el reconocimiento de la voz humana.



Espacio Código III.23: Archivo vxml.conf

### 3.5.6.9 CONFIGURAR MODULES.CONF

En este archivo se habilitan los módulos que se van a utilizar, se encuentra ubicado en **/etc/asterisk/modules.conf**, para el prototipo se modifica de la siguiente manera:

**#vi /etc/asterisk/modules.conf**

Y se debe modificar lo siguiente:

- noload => res\_speech\_lumenvox.so -> el cual debe ir como noload por que no se utilizará ese modulo para el reconocimiento de voz
- load => res\_speech\_verbio.so-> el cual debe añadir como load por que se utilizará ese modulo para el reconocimiento de voz
- load => chan\_dahdi.so-> permitirá que interactuar con la tarjeta y puede navegar por la PSTN.

```
;
; Module Loader Configuration
;
[modules]
autoload=yes
load => res_speech.so
noload => res_speech_lumenvox.so
load => res_speech_verbio.so
load => chan_dahdi.so
noload => chan_gtalk.so
noload => res_jabber.so
noload => chan_ss7.so
```

**Espacio Código III.24: Archivo modules.conf**

### 3.5.6.10 CONFIGURAR VERBIO.CONF

Donde se debe realizar la modificación del tts y asr, las cuales se muestran en el ANEXO 1.

### 3.5.6.11 INSTALACIÓN DEL ADMINISTRADOR WEB (WEBMIN)

- Instalamos un administrador web llamado **WEBMIN**, que ayudará a configurar y administrar el sitio virtual que se necesita. Ver Gráfico III.34. Para ello se utiliza la siguiente línea de comandos:

```
#rpm -vih wembmin -1.370-1.noarch.rpm
```

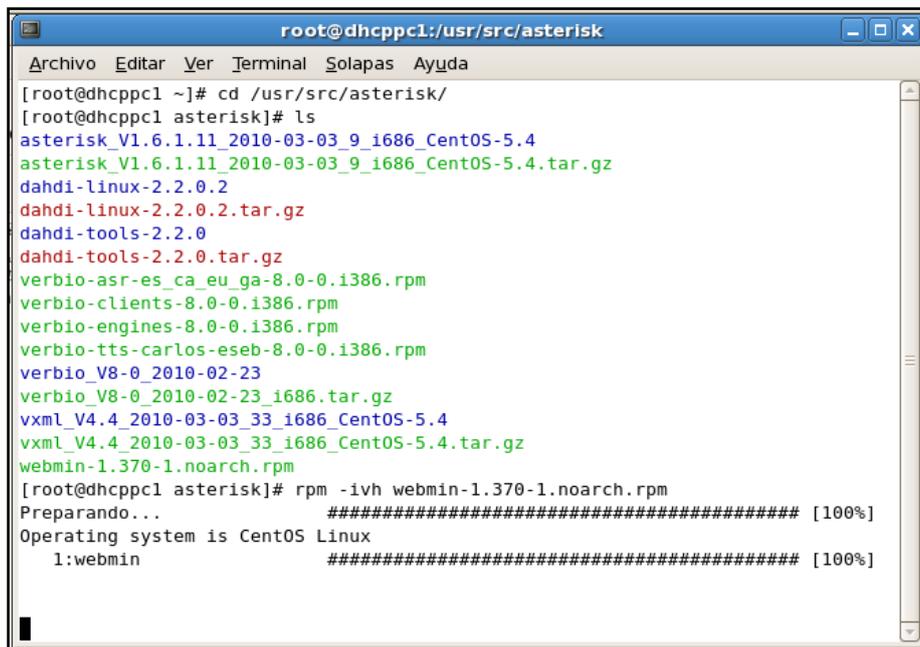
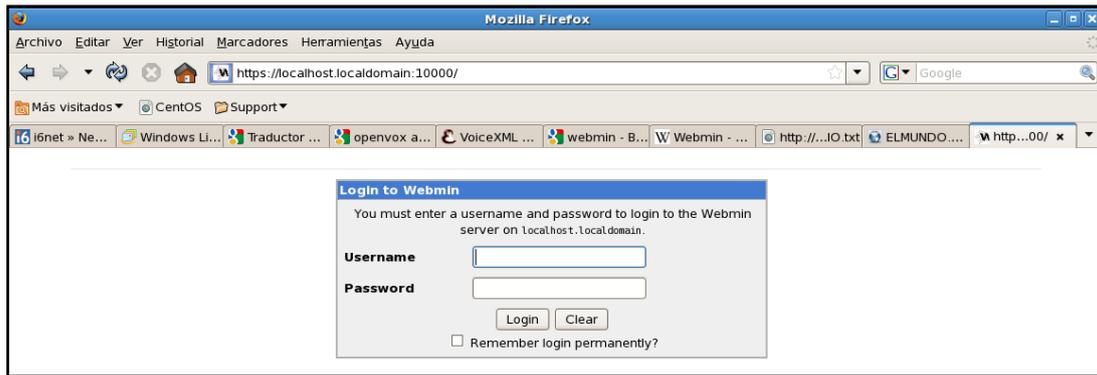


Gráfico III.34: Instalación del Webmin

- Luego ingresamos al administrador Webmin a través de <http://localhost.localdomain:10000> y se muestra la siguiente figura. Ver Gráfico III.35



**Gráfico III.35: Inicio de Sesión Webmin**

Ingresamos al administrador web, utilizando la cuenta root y la contraseña de la misma cuenta. Nos dirigimos al Servidor Web Apache para configurarlo y poder publicar nuestra aplicación web, la cual permita alimentar datos al prototipo.

#### **3.5.6.11.1 CONFIGURACIÓN DEL SERVIDOR WEB APACHE**

El servidor web apache es el encargado de alojar la aplicación web como la aplicación VoiceXML. La aplicación web permitirá realizar la utilización del prototipo telefónico como el ingreso de datos mientras tanto que los archivos VoiceXML serán accedidos desde la PSTN, teléfonos móviles y teléfonos IP por parte del usuario.

- Se debe ubicar a través del webmin, en el servidor web apache y nos aparecerá el Gráfico III.36.

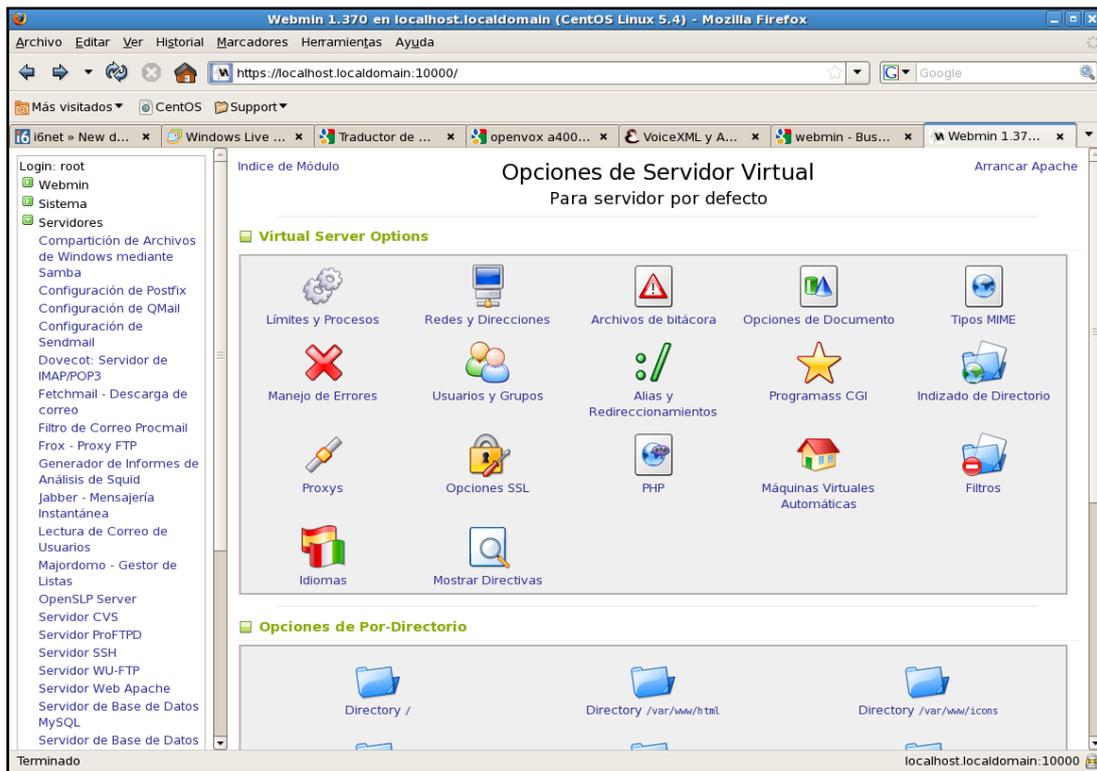
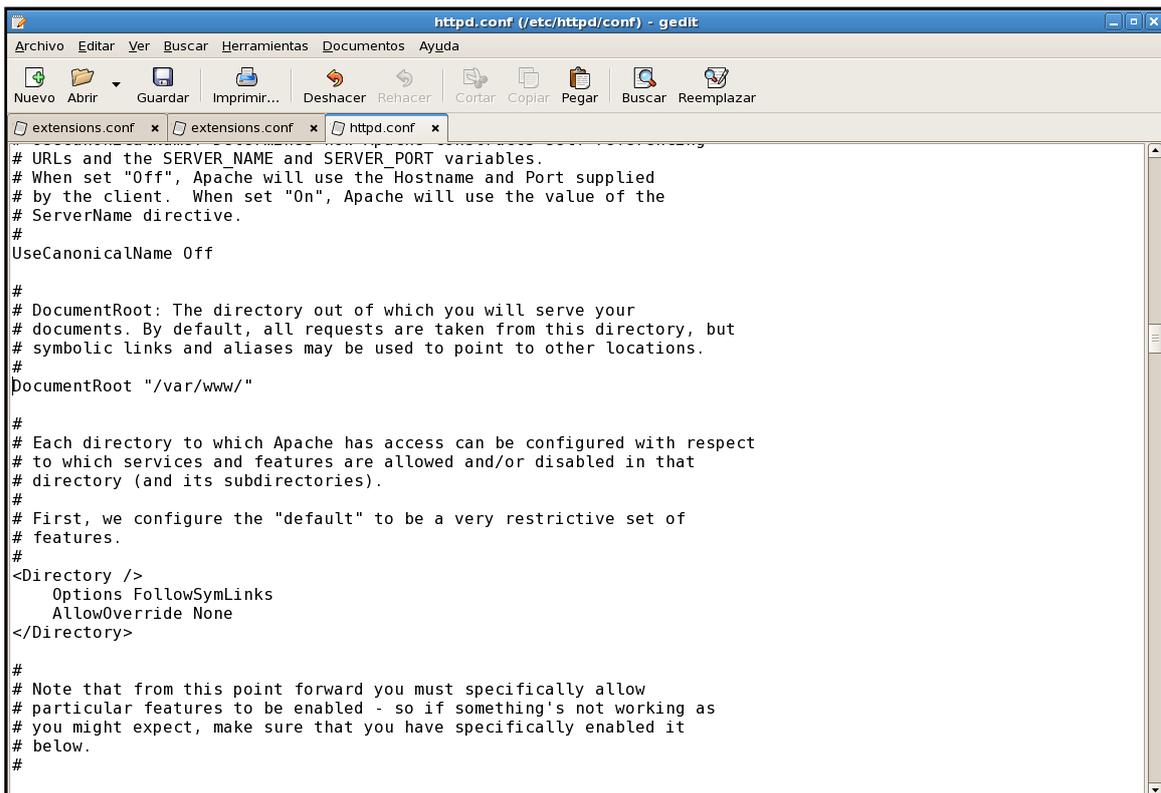


Gráfico III.36: Servidor Web Apache

- Se debe ingresar a la opción, **indexado de directorio** para poder configurar los índices de la aplicación web, en este caso se agrega index.php, index.html y index.vxml.
- Luego se debe ingresar al archivo httpd.conf en el cual se debe configurar el Servidor Virtual para el prototipo, en este caso no se ha desarrollado ningún DNS es por eso que se desarrolla la modificación del **DocumentoRoot "/var/www"**, debido a que los archivos vxml se deben ubicar en la carpeta /www:

**#vi /etc/httpd/conf/httpd.conf**



**Gráfico III.37: Archivo httpd.conf**

- Luego de los cambios, arrancamos el servicio web apache, con la siguiente línea de código:

**#service httpd start**

### **3.5.6.11.2 INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR DE BASE DE DATOS MYSQL**

Este servidor será el encargado de alojar a la base de datos del sistema, a esta base de datos accederán tanto la aplicación web como la aplicación telefónica vía PTSN.

- Se debe crear la base de datos, para el prototipo se creó la base de datos telefonía, esto se desarrolló desde el WEBMIN como se lo muestra a continuación en el Gráfico III.38.

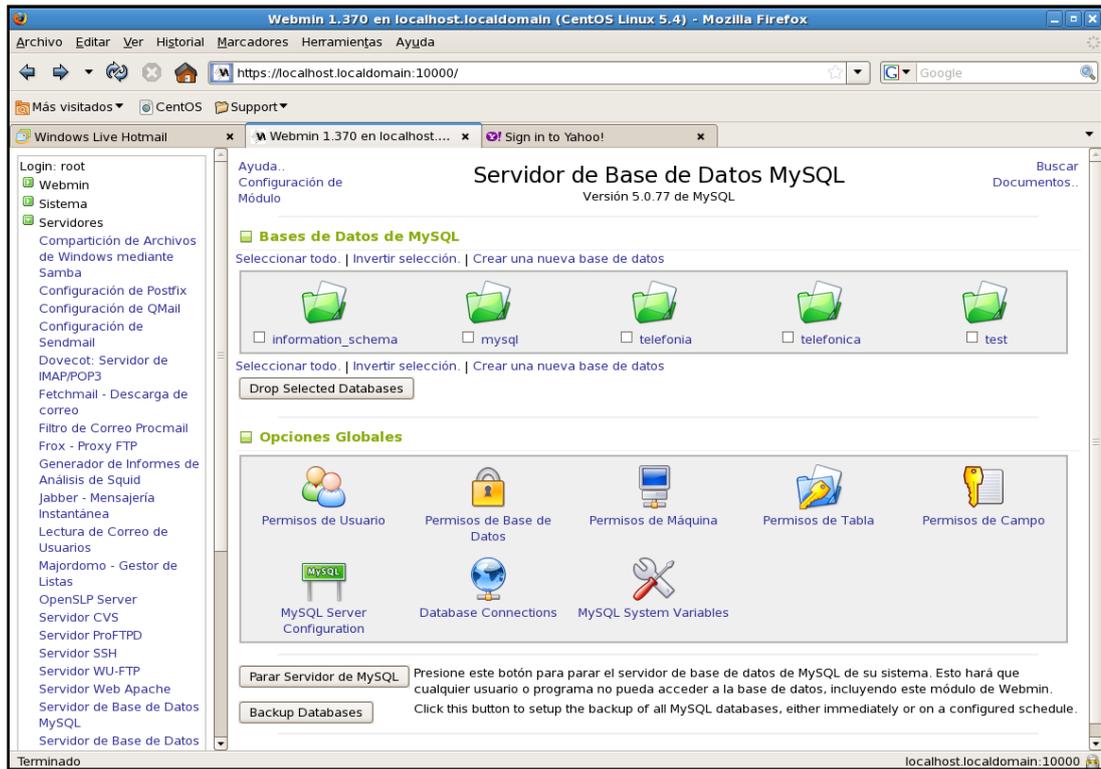


Gráfico III.38: Servidor de Base de Datos MySQL

- Ingresamos a la base de datos y creamos las tablas a necesitar, en este caso son las siguientes tablas:

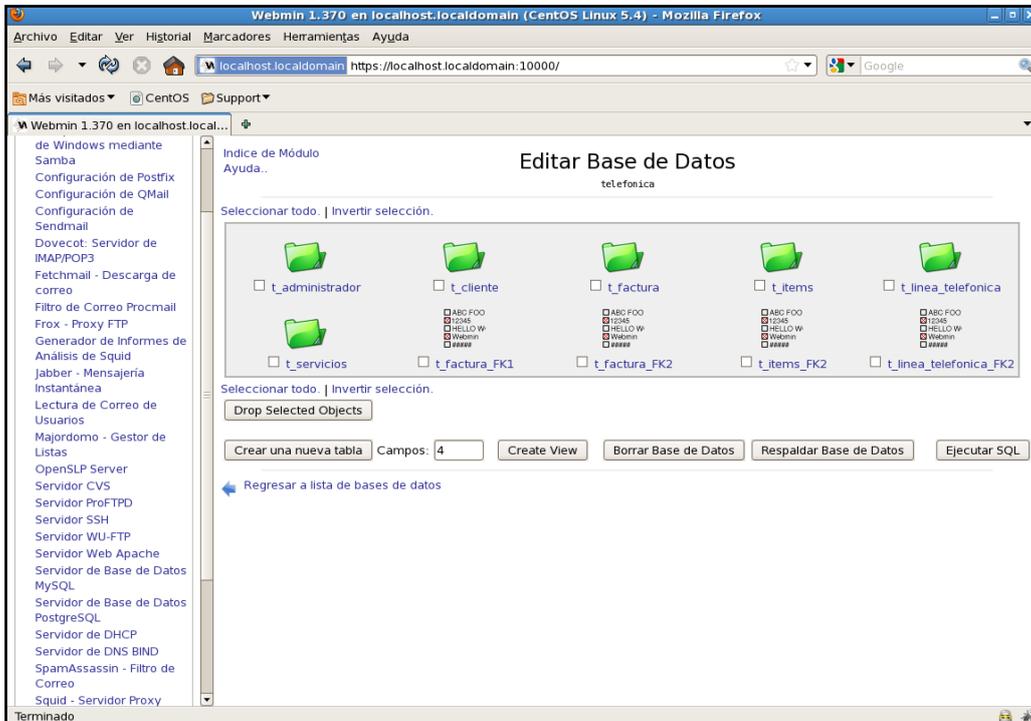


Gráfico III.39: Base de Datos Telefónica

### 3.5.7 PRUEBAS INICIALES DE LA PLATAFORMA

El archivo de registro de sucesos log.txt, en este archivo se guarda todos los sucesos que genera la utilización de la plataforma, basados en este archivos puede indicarse el funcionamiento correcto de nuestra aplicación.

#### 3.5.7.1 FUNCIONAMIENTO DE LA PBX

Al iniciar la llamada a través del archivo log.txt, se puede observar el funcionamiento de la pbx cuando inicia una llamada como se muestra en el Espacio de Código III.25.

```
Apr 26 15:54:20.45|-1718654064|57|7010|DEV|Start call
Apr 26 15:54:20.45|-1718654064|57|7010|DEV|-
URL=http://192.168.1.4/bienvenida.vxml
Apr 26 15:54:20.46|-1718654064|57|8010|DEV|Execute:
http://192.168.1.4/bienvenida.vxml
```

**Espacio Código III.25: Inicio de la PBX**

#### 3.5.7.1.1 PROBAR EL SERVIDOR TEXT TO SPEECH

```
Apr 26 15:54:20.47|-1718654064|57|5010|DEV|Prompt (text/plain):
Buenas tardes          Son las 15horas,          54 minutos
Apr 26 15:54:21.16|-1718654064|57|5010|DEV|Prompt (text/plain):
Bienvenidos al Sistema de Facturacion VoiceXML. a continuacion se
presentan las siguientes opciones.
Apr 26 15:54:21.16|-1718654064|57|8010|DEV|Execute: menu.vxml
Apr 26 15:54:21.17|-1718654064|57|5010|DEV|Wait prompting
Apr 26 15:54:31.78|-1718654064|57|5010|DEV|Prompt (text/plain):
,Menu principal.      , saldos.      , servicios.      , ayuda.      ,
salir.      escoja su opcion por favor
Apr 26 15:54:31.78|-1718654064|57|5010|DEV|Wait prompting
Apr 26 15:54:39.61|-1718654064|57|6010|DEV|Wait input
```

**Espacio Código III.26: TTS**

### 3.5.7.1.2 PROBAR EL SERVIDOR ASR

```
Apr 26 15:54:43.33|-1718654064|57|6010|DEV|Input:servicios
Apr 26 15:54:43.33|-1718654064|57|8010|DEV|Execute:
servicios.vxml
Apr 26 15:54:43.40|-1718654064|57|SEVERE|swi:SBjsi|501|SBjsi:
ECMAScript engine exception|errmsg=ReferenceError: caller is not
defined|line=1|linetxt=|tokentxt=
Apr 26 15:54:43.40|-1718654064|57|8010|DEV|Execute:
servicios.php
Apr 26 15:54:43.94|-1718654064|57|5010|DEV|Prompt(text/plain):
Servicios;      <br></br><b>Notice</b>: Undefined index:
caller in <b>/var/www/servicios.php</b> on line
<b>13</b><br></br>
Apr 26 15:54:43.94|-1718654064|57|5010|DEV|Prompt(text/plain):
La empresa VooiceXML ofrece los siguiente servicios
Apr 26 15:54:44.65|-1718654064|57|5010|DEV|Prompt(text/plain):
Internet el valor del servicio es 100 dolares
Apr 26 15:54:45.20|-1718654064|57|5010|DEV|Prompt(text/plain):
Gracias por visitarnos, proxicamente estaremos lanzando nuevos
servicios. Regresamos al
Apr 26 15:54:46.08|-1718654064|57|8010|DEV|Execute: menu.vxml
Apr 26 15:54:46.09|-1718654064|57|5010|DEV|Wait prompting
Apr 26 15:54:56.38|-1718654064|57|5010|DEV|Prompt(text/plain):
,Menu principal.      , saldos.      , servicios.      , ayuda.
, salir.      escoja su opcion por favor
Apr 26 15:54:56.38|-1718654064|57|5010|DEV|Wait prompting
```

**Espacio Código III.27: ASR**

## **CAPITULO IV**

### **DISEÑO E IMPLEMENTACION DEL PROTOTIPO**

#### **4.1 INTRODUCCIÓN**

En el presente capítulo se describe el diseño e implementación de un pequeño sistema que simule la generación del valor a pagar por consumo de una planilla telefónica, cabe recalcar que el objetivo de este prototipo no corresponde en solucionar el problema antes mencionando sino dar a conocer nuevas formas del acceso a datos como es el caso del estándar VoiceXML.

Actualmente la mayoría de usuarios realizan la consulta de la información a través del internet, como es de dominio público para acceder a esta información debemos estar conectados a un computador necesariamente y muchas de las veces la información que necesitamos obtener no depende de contenidos gráficos, más bien información mínima pero importante a la vez como por ejemplo conocer el valor a pagar por consumo de una línea telefónica.

Para esto, usando el estándar VoiceXML se propone desarrollar una aplicación informática que permita ser accedida desde un aparato telefónico, de fácil de uso, con disponibilidad las veinticuatro horas del día, los trescientos sesenta y cinco días del año.

El sistema consiste en una aplicación VoiceXML, la misma que se accederá marcando un número telefónico, la aplicación será automática y no dependerá de la intervención humana, el usuario deberá proporcionar el número de la línea telefónica para desarrollar la consulta requerida, el sistema realizará la consulta a la base de datos y proporcionará nuevamente al usuario una respuesta.

La aplicación deberá reducir los costos de soluciones propietarias y para esto se utilizará en cuanto sea posible software libre sin descuidar por otro lado la calidad de servicio que se brinde al usuario.

## **4.2 DISEÑO DE LA SOLUCIÓN**

### **4.2.1 DESCRIPCIÓN DEL SERVICIO**

El servicio ofrecido por el prototipo proporcionará a los usuarios un menú de opciones con el cual podrán realizar consultas del valor de las planillas telefónicas y se divide en 3 partes funcionales:

### **4.2.2 CONSULTA DE PLANILLA TELEFÓNICA**

Las personas podrán ser atendidas a sus peticiones, en esta opción se plantearán dos sub-opciones que se presenta a continuación

- Valor total de la planilla telefónica  
El usuario al utilizar esta opción podrá conocer el valor total a pagar por el consumo de la línea telefónica.
- Detalle de la planilla telefónica  
El uso de esta opción presentará al usuario el detalle de la planilla a pagar por el consumo realizado durante el periodo vigente.

### **4.2.3 SERVICIOS**

Como una nueva opción del servicio, la aplicación VoiceXML permitirá al usuario conocer información como:

- Planes de internet
- Consumo de internet
- Tarifas

### **4.2.4 AYUDA**

Con esta opción los usuarios podrán informarse de la forma en que pueden usar el sistema, enfocará aspectos como navegabilidad, formato del ingreso de información, elección de las opciones, etc.

### **4.2.5 DIAGRAMAS DE FLUJO DE LA SOLUCIÓN**

A continuación se presentarán los flujos gramas o diagramas de flujo diseñados para el prototipo VoiceXml:

### 4.2.5.1 APLICACIÓN TELEFÓNICA

El diagrama de flujo de la aplicación telefónica representa la navegabilidad del sistema por parte del usuario.

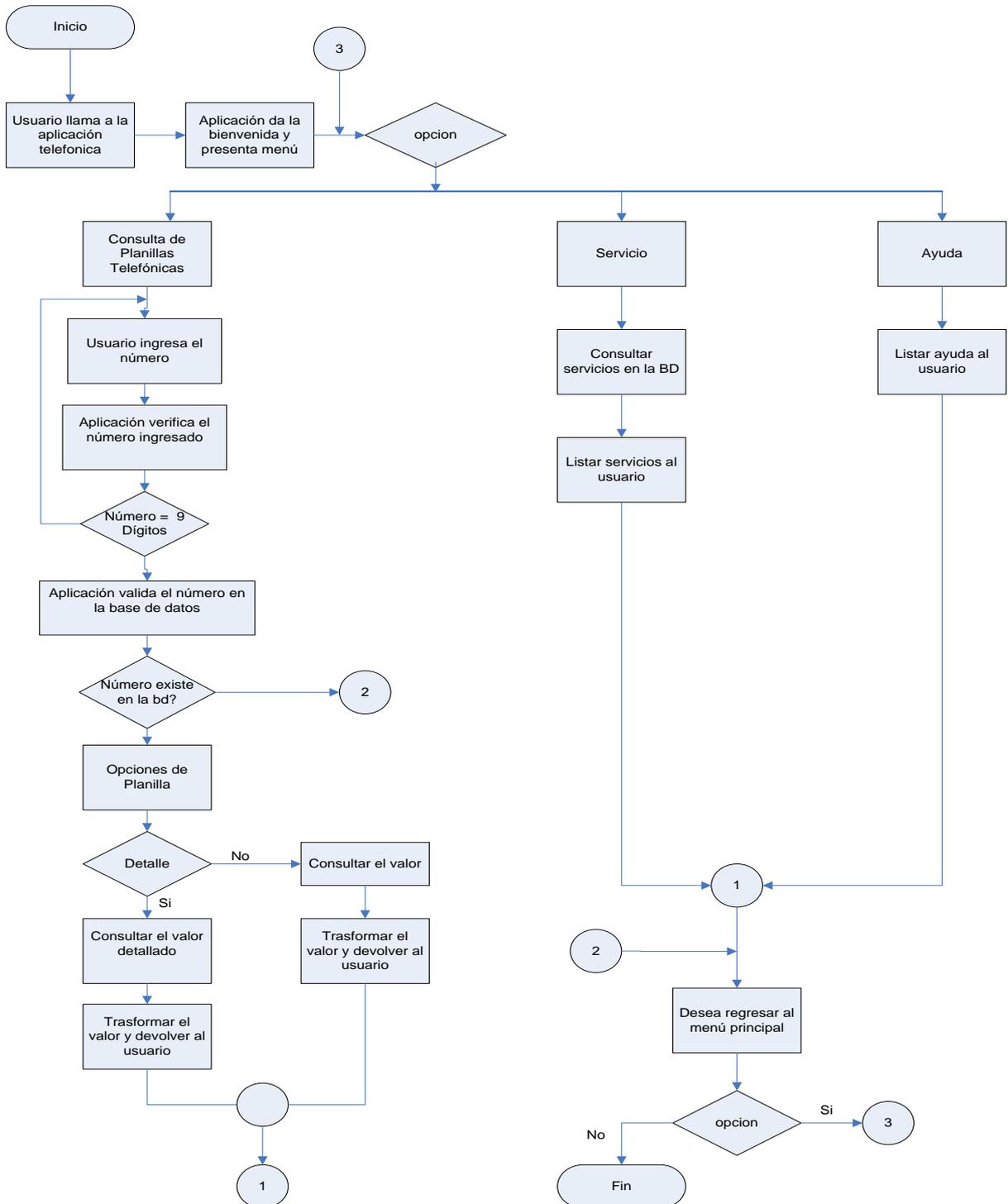
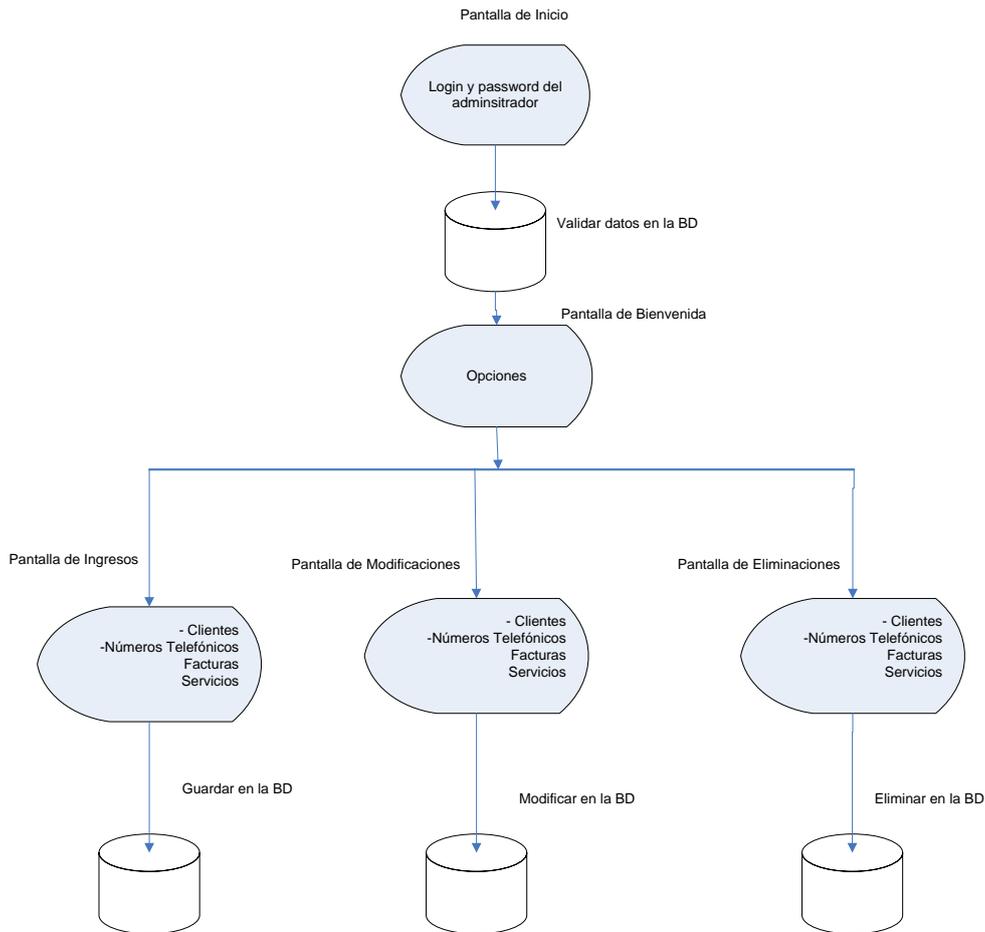


Gráfico IV.40: Diagrama de Flujo de la Aplicación Telefónica

#### 4.2.5.2 APLICACIÓN WEB

La aplicación web permitirá al administrador cargar los datos al sistema, realizará operaciones simples como ingreso, modificación, eliminación de datos.

Para detallar los requerimientos del prototipo a continuación se despliega un diagrama de flujo del comportamiento de la aplicación web.

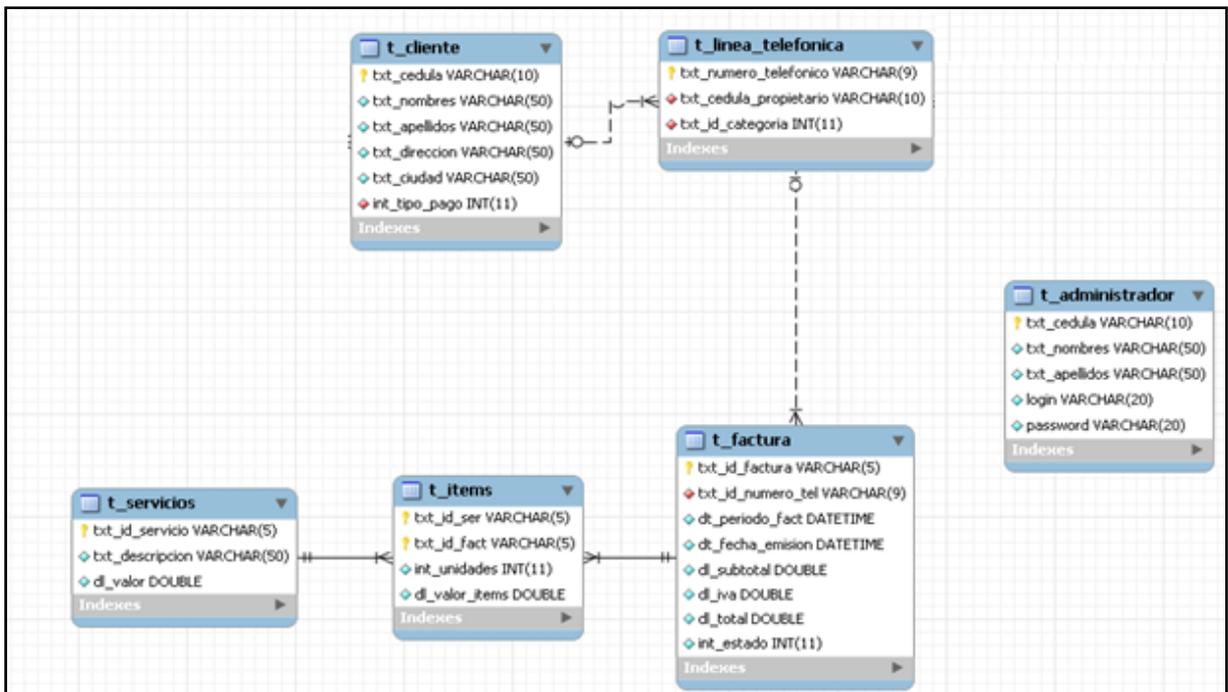


**Gráfico IV.41: Diagrama de Flujo del comportamiento de la aplicación web**

#### 4.2.5.3 MODELO ENTIDAD RELACIÓN

Basados en las consideraciones anteriores tanto del diagrama de flujo de la aplicación telefónica como de la aplicación web el diagrama entidad-relación del prototipo se describe en el Gráfico IV.42.

#### 4.2.5.3.1 DICCIONARIO DE DATOS



**Gráfico IV.42: Diagrama de Flujo del Diccionario de Datos**

A continuación se presentará la descripción de cada uno de los datos de las tablas del modelo antes mostrado:

- **Tabla Administrador (t\_administrador)**

Nombre	Tipo de Dato	Descripción
txt_cedula	VARCHAR	Llave primaria, es el código del administrador para utilizar el sistema
txt_nombres	VARCHAR	Nombre de usuario administrador
txt_apellidos	VARCHAR	Apellidos del usuario administrador
Login	VARCHAR	Es la cuenta para el ingreso del administrador
Password	VARCHAR	Es la clave de ingreso del administrador

**Tabla IV.11: Tabla Administrador**

- **Tabla Cliente (t\_cliente)**

Nombre	Tipo de Dato	Descripción
txt_cedula	VARCHAR	Llave primaria, es el identificador del cliente que puede poseer una o varias líneas telefónicas
txt_nombres	VARCHAR	Nombres del cliente
txt_apellidos	VARCHAR	Apellidos del cliente
txt_direccion	VARCHAR	Dirección del cliente
txt_ciudad	VARCHAR	Ciudad del cliente

**Tabla IV.12: Tabla Cliente**

- **Tabla Factura (t\_factura)**

Nombre	Tipo de Dato	Descripción
txt_id_factura	VARCHAR	Llave primaria, es el identificador de la factura
txt_id_numero_tel	VARCHAR	Es el identificador del número telefónico al cual corresponde esta factura
dt_periodo_fact	DATETIME	Fecha a la cual corresponde el periodo de pago
dt_fecha_emision	DATETIME	Fecha en la que debe ser pagada la factura
dl_subtotal	DOUBLE	Es el sub valor a pagar
dl_iva	DOUBLE	Es el valor correspondiente al iva de esta factura
dl_total	DOUBLE	Corresponde el valor total de la factura
int_estado	INT	Campo mediante el cual identificaremos si una factura aun no ha sido cancelada

**Tabla IV.13: Tabla Factura**

- **Tabla items (t\_items)**

Nombre	Tipo de Dato	Descripción
txt_id_ser	VARCHAR	Clave foránea para identificar el tipo de servicio consumido
txt_id_fact	VARCHAR	Clave foránea para identificar a que factura pertenece el ítem consumido
cantidad	INT	Campo para indicar la cantidad de servicios consumidos
total_item	DOUBLE	Total del ítem consumido

**Tabla IV.14: Tabla Items**

- **Tabla línea telefónica (t\_linea\_telefonica)**

Nombre	Tipo de Dato	Descripción
txt_numero_telefonico	VARCHAR	Llave primaria, es el identificador de la línea telefónica
txt_cedula_propietario	VARCHAR	Clave foránea, permite asociar una línea telefónica con el cliente

**Tabla IV.15: Tabla Línea Telefónica**

- **Tabla servicios (t\_servicios)**

Nombre	Tipo de Dato	Descripción
txt_id_servicio	VARCHAR	Llave primaria, es el identificador servicio
txt_descripción	VARCHAR	Descripción del servicio
dl_valor	DOUBLE	Es el valor del servicio

**Tabla IV.16: Tabla Servicios**

#### **4.2.6 LIMITACIONES DEL SISTEMA**

Por tratarse de un prototipo que a futuro puede ser implementado, el desarrollo del presente sistema estará basado en las siguientes limitaciones:

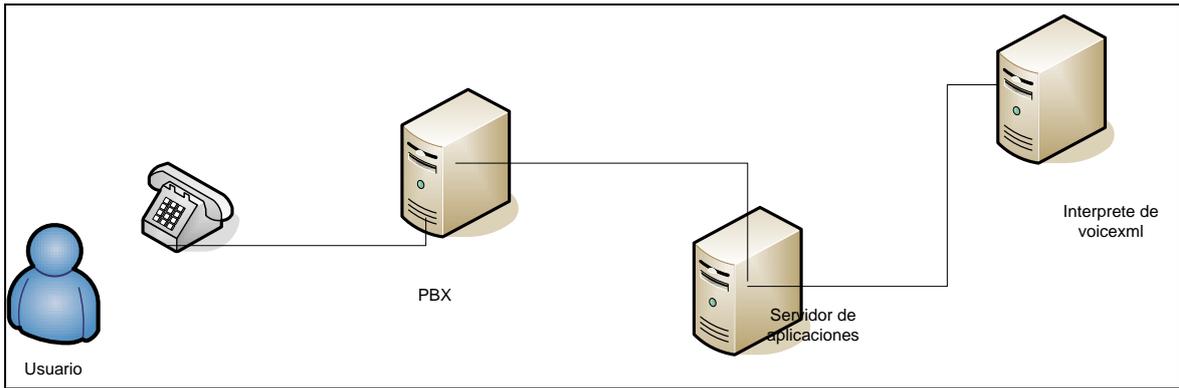
- El administrador será el encargado de ingresar datos al sistema para que puedan ser accedidos por el usuario desde un teléfono convencional.
- El tipo de información que se va a manipular no se considera de carácter confidencial por lo que no es necesario crear un modulo de autenticación de usuarios.
- La interacción del sistema con el usuario es a través de una línea telefónica, la misma que se limitará a una sencilla consulta de base de datos, excluyendo del desarrollo de este sistema operaciones como insertar, modificar o eliminar información por parte del usuario.
- La información se la proporcionará directamente al usuario, el sistema permitirá al usuario volver al menú para utilizar nuevamente el servicio.

#### **4.2.7 DISEÑO DEL SISTEMA**

##### **4.2.7.1 APLICACIÓN TELEFÓNICA**

###### **4.2.7.1.1 INGRESO AL SISTEMA**

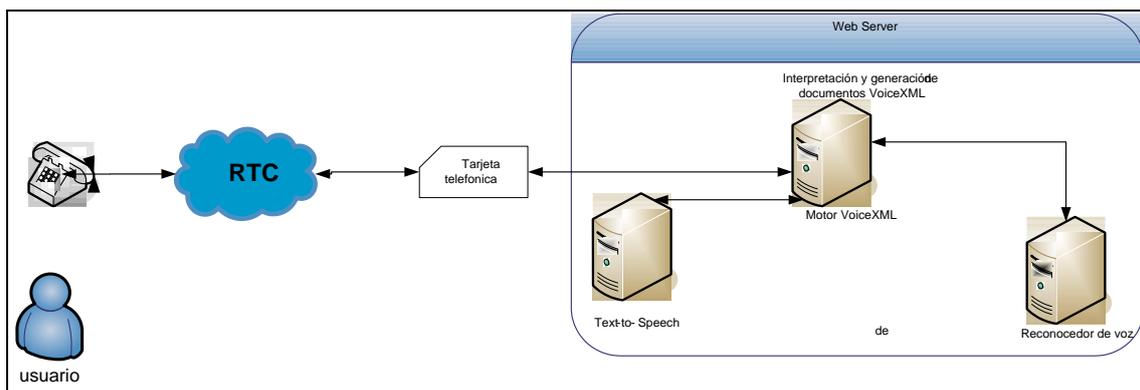
El usuario llama a la aplicación telefónica para esto debe marcar un número el mismo que será receptado por la aplicación e iniciará el proceso de utilización del sistema. Una vez iniciado la interacción del sistema con el usuario, el sistema desplegará el método para el proceso de escoger la opción.



**Gráfico IV.43: Interacción de los módulos en el Ingreso al Sistema**

#### 4.2.7.1.2 ELECCIÓN DEL MENÚ

El intérprete de VoiceXML apoyado en el servidor TTS es el encargado de indicarle al usuario las diferentes opciones del sistema. Una vez que el usuario haya escuchado las diferentes opciones del sistema, éste podrá inclinarse por cualquiera de las mismas, para que el usuario escoja la opción deseada debe pronunciar las palabras que haga referencia a la opción por ejemplo el usuario debe pronunciar ayuda para que el sistema le lleve a la opción de ayuda. En esta parte el sistema hará uso del servidor ASR que es el encargado de realizar el reconocimiento automático de la voz.



**Gráfico IV.44: Interacción de los Módulos para Elección del Menú**

#### **4.2.7.1.3 CONSULTA DE PLANILLAS**

Este modulo de la aplicación consta de dos servicios diferentes los cuales mencionamos a continuación:

- Consulta de planillas por valor total
- Consulta de planillas por detalles

#### **4.2.7.1.4 CONSULTA DE PLANILLAS POR VALOR TOTAL**

El usuario al realizar este tipo de consulta recibirá como respuesta un único valor por concepto del pago del valor de consumo de la factura telefónica.

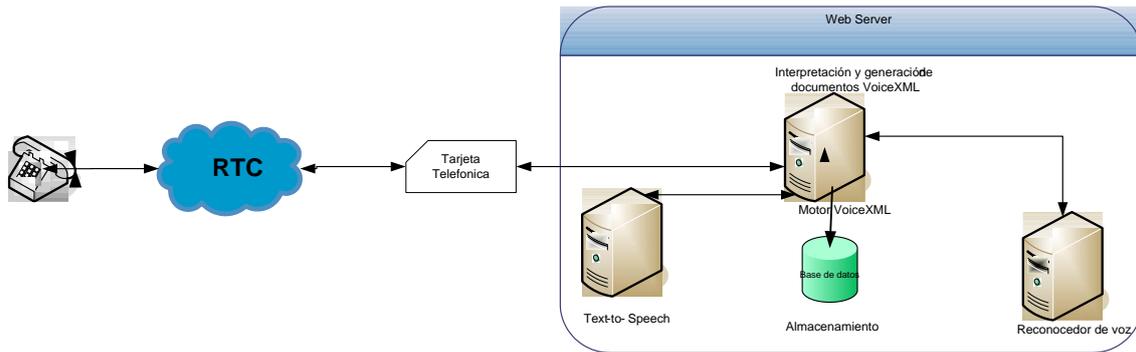
El funcionamiento de esta parte del sistema es la siguiente:

- El sistema pide al usuario que ingrese el número a consultar
- Usuario debe ingresar el número de una forma correcta
- El sistema apoyado en el modulo ASR captura el número ingresado y lo transforma en una cadena de caracteres
- La aplicación ayudado por el servidor de aplicaciones verifica y valida el número ingresado por el usuario
- Si el número es correcto el sistema realiza la consulta en la base de datos y el módulo VoiceXML genera el archivo de respuesta.
- El sistema apoyado en el modulo TTS reproduce la información contenida en el archivo generado anteriormente y lo devuelve al usuario en forma de voz que en este caso será un único valor.

#### **4.2.7.1.5 CONSULTA DE PLANILLAS A DETALLE**

El usuario al realizar este tipo de consulta recibirá como respuesta el detalle de la descripción del valor de la factura.

El funcionamiento es idéntico al proceso anterior con la única diferencia que la consulta que se realiza a la base de datos contiene los valores detallados por consumo los mismos que serán indicados al usuario en forma de voz.



**Gráfico IV.45: Interacción de los Módulos para la Consulta de Planillas Telefónicas**

#### **4.2.7.1.6 CONSULTA DE SERVICIOS**

La consulta de servicios es un agregado de este prototipo, que permitirá al usuario conocer las promociones, precios, servicios de internet, etc.

El funcionamiento de la parte del servicio que brinda la aplicación es muy parecido a los módulos anteriores. El usuario solamente debe escoger la opción servicios en el menú principal y no se debe realizar más ingresos de información.

Para el cumplimiento de esta opción interactúan de forma ordenada todos los módulos de la aplicación.

#### **4.2.7.1.7 AYUDA**

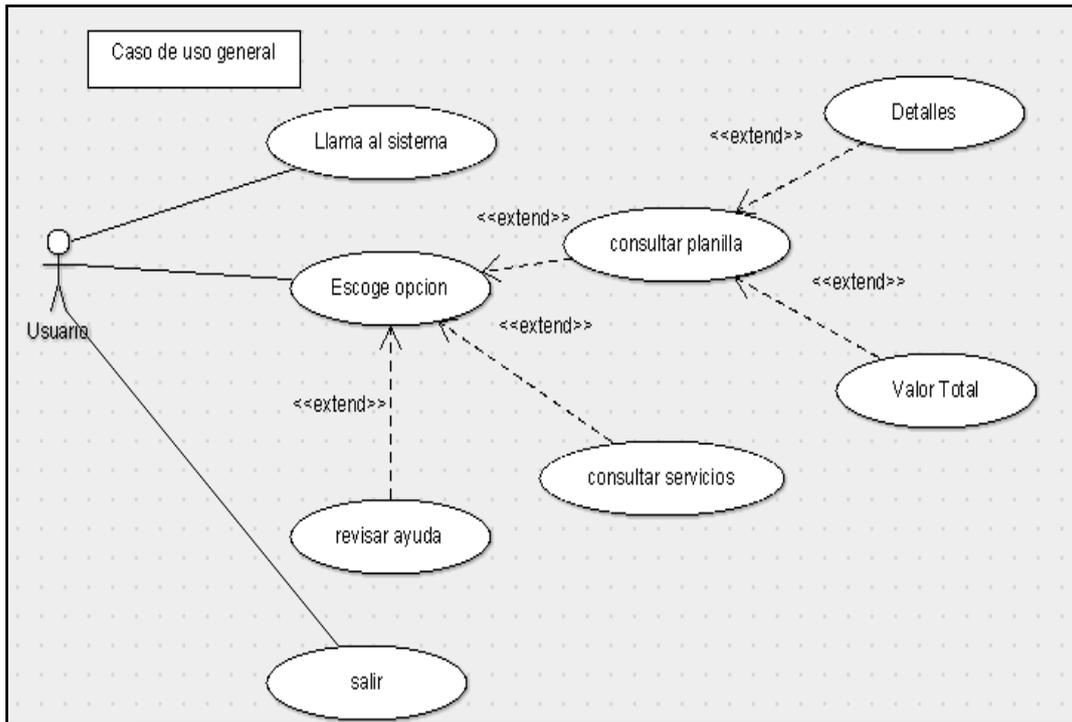
Esta parte del sistema es la más importante y fácil en implementarse en vista que constará únicamente en un archivo de salida, no interactuará con la base de datos y los únicos módulos que funcionarán es el módulo VoiceXML con el módulo TTS.

#### **4.2.7.2 MODELO FUNCIONAL DE LA APLICACIÓN TELEFÓNICA**

Dentro de la descripción del modelo funcional se han incluido los casos de uso que más importantes y relevantes han parecido con respecto a la aplicación telefónica.

##### **4.2.7.2.1 CASO DE USO GENERAL ASOCIADO A LA APLICACIÓN TELEFÓNICA**

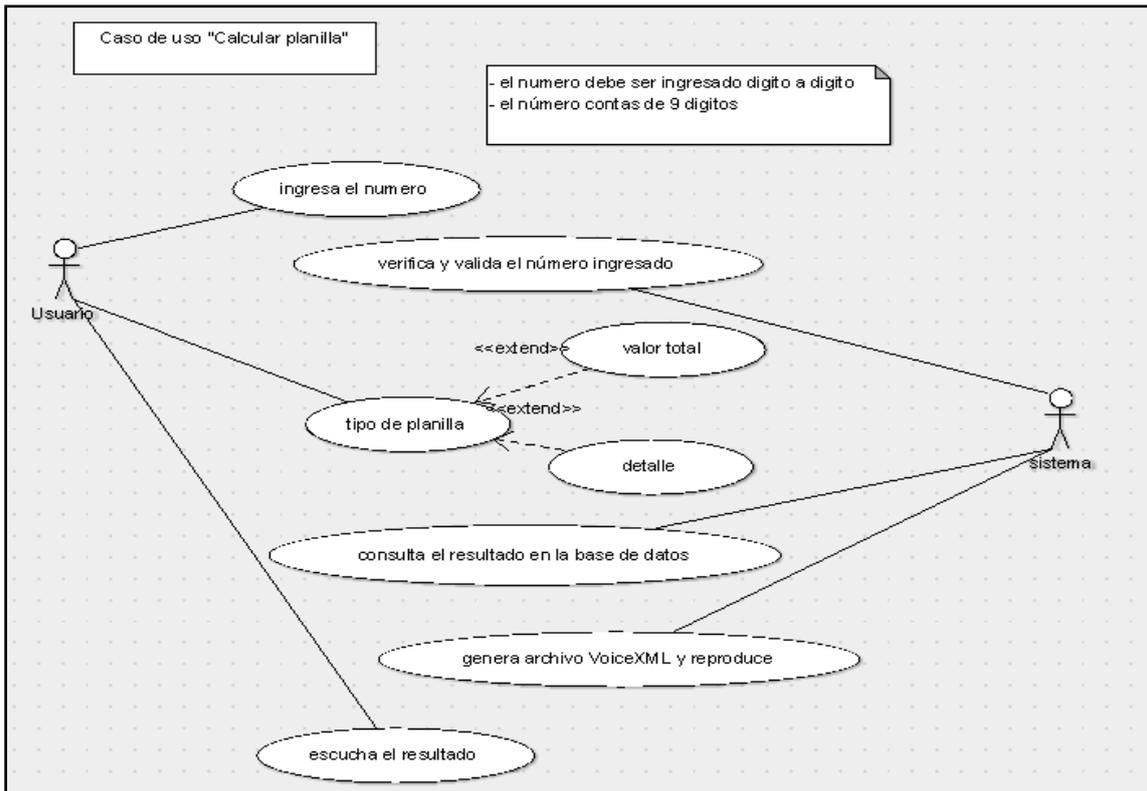
En la siguiente figura se describe el caso de uso general que integra las diferentes operaciones que el usuario puede realizar desde la aplicación telefónica.



**Gráfico IV.46: Caso de uso general de la aplicación telefónica**

#### **4.2.7.2.2 CASO DE USO ASOCIADO A LA CONSULTA DE PLANILLA TELEFÓNICAS**

En el modulo de consulta de planillas el siguiente refleja el caso de uso del comportamiento del sistema



**Gráfico IV.47: Caso de uso para la Consulta de Planillas Telefónicas**

#### 4.2.8 APLICACIÓN WEB

La aplicación web se diseñará de una forma fácil y sencilla tomando en cuenta los requerimientos descritos en el diagrama de flujo de la figura. Para esto se utilizarán conocimientos básicos de código HTML y PHP, tomando en cuenta que el presente estudio no abarca los temas antes mencionados, los objetivos principales de la aplicación web están orientados a realizar una programación rápida y eficaz que permita construir una solución informática que genere los datos necesarios para que sean utilizados por la aplicación telefónica y de esta manera cumplir los objetivos propios del estudio.

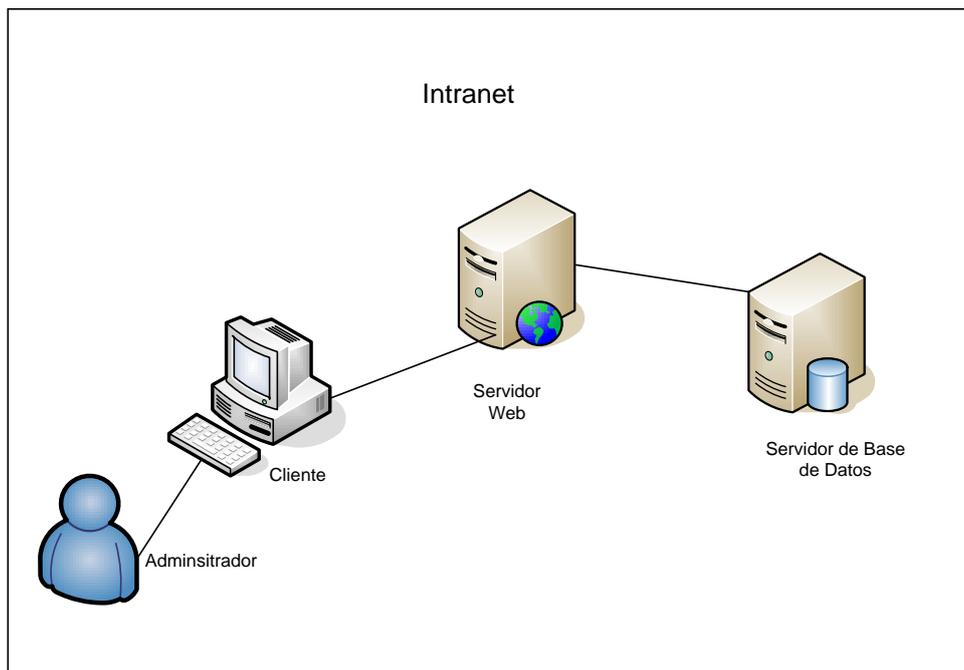
El código HTML se utilizará para la construcción de las interfaces gráficas que permitan capturar los datos que interactuarán con el sistema como ingreso, modificación y eliminación.

El lenguaje PHP permitirá interactuar los formularios creados en HTML con nuestra base de datos, en este caso el motor de base de datos Open Source con grandes prestaciones como es MYSQL, en dicha base de datos se alojarán los datos que

manipule el administrador, a la misma que será accedida por la aplicación telefónica por parte del usuario.

#### 4.2.8.1 ARQUITECTURA DE LA APLICACIÓN WEB

Una aplicación web típica utiliza un servidor web y un servidor de bases de datos. La arquitectura de aplicaciones web es una gran importancia para presentar los datos de las empresas. Permite acceder a la información en el momento adecuado y en el tiempo justo. A continuación se presenta el esquema gráfico de la arquitectura de la aplicación Web.

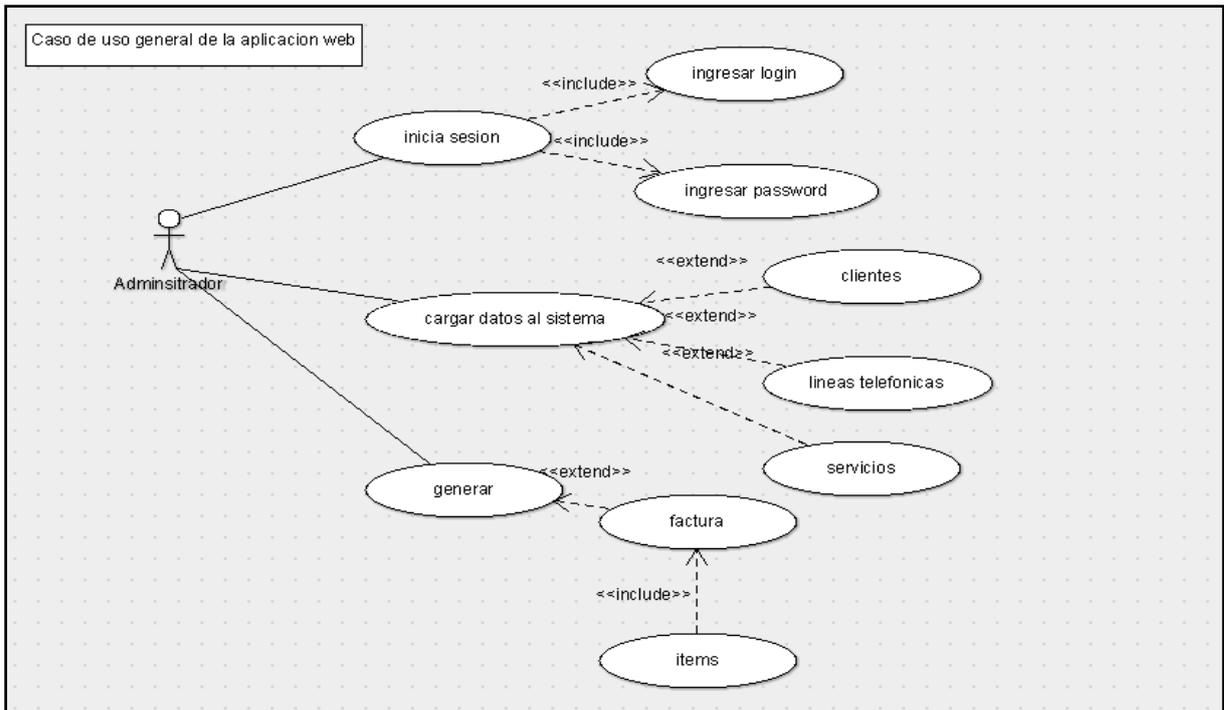


**Gráfico IV.48: Arquitectura de la aplicación web**

#### 4.2.8.2 MODELO FUNCIONAL DE LA APLICACIÓN WEB

A continuación se detalla los casos de uso más representativos que resume el comportamiento de la aplicación web.

##### 4.2.8.2.1 CASO DE USO GENERAL ASOCIADO A LA APLICACIÓN WEB



**Gráfico IV.49: Caso de Uso General correspondiente a la Aplicación Web**

#### 4.3 IMPLEMENTACIÓN DEL PROTOTIPO

El desarrollo del servicio debe estar soportado sobre una arquitectura robusta y escalable, que permita un mantenimiento adecuado de la misma a lo largo del tiempo, por lo que se propone utilizar tecnologías de vanguardia como el modelo LAMP (tecnologías Linux – Apache – MySQL – PHP) perteneciente al movimiento de software libre. El sistema operativo Linux es el encargado de soportar las tres plataformas tecnológicas restantes, mientras que el servidor web Apache aloja las páginas Hyper Text Markup Language (HTML), PHP y VXML, entre otras. El manejador de bases de datos MySQL es el encargado de acceder la información de persistencia propia del negocio. Y, finalmente, el manejador de páginas dinámicas PHP es el encargado de la ejecución del motor de la lógica del negocio.

El sistema por tratarse de un prototipo el servidor se encuentra implementado en una PC de procesador Intel Pentium 4, 512MB RAM corriendo el sistema operativo CentOS 5.4.

La tarjeta OpenVox es la encargada de permitir el acceso vía telefónica y móvil a los usuarios a la aplicación, esta tarjeta se debe configurar en el servidor Asterisk, propiamente en el módulo DAHDI.

### 4.3.1 DESARROLLO DE LA APLICACIÓN WEB

#### 4.3.1.1 PANTALLA DE INICIO

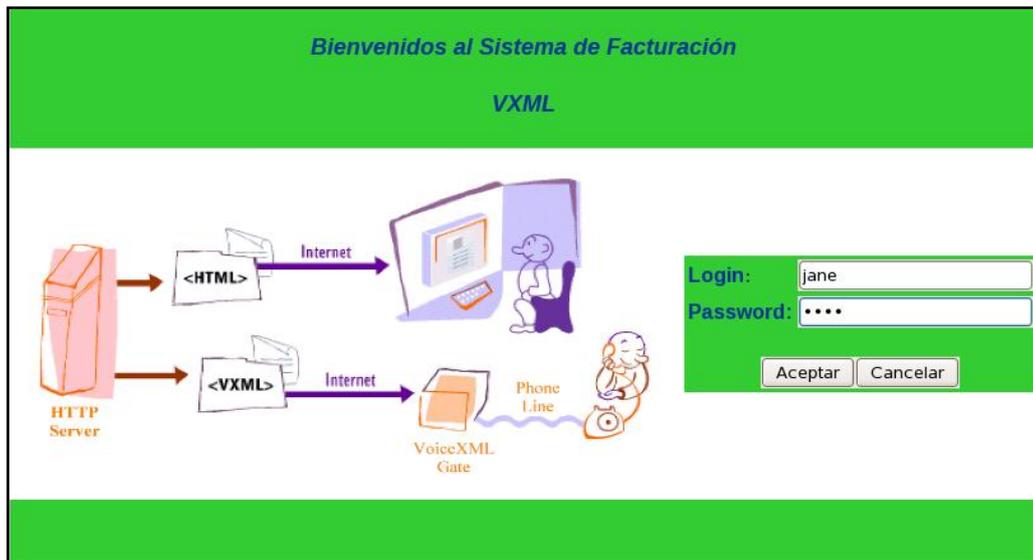


Gráfico IV.50: Pantalla de Inicio

#### 4.3.1.2 PANTALLA DE INGRESOS

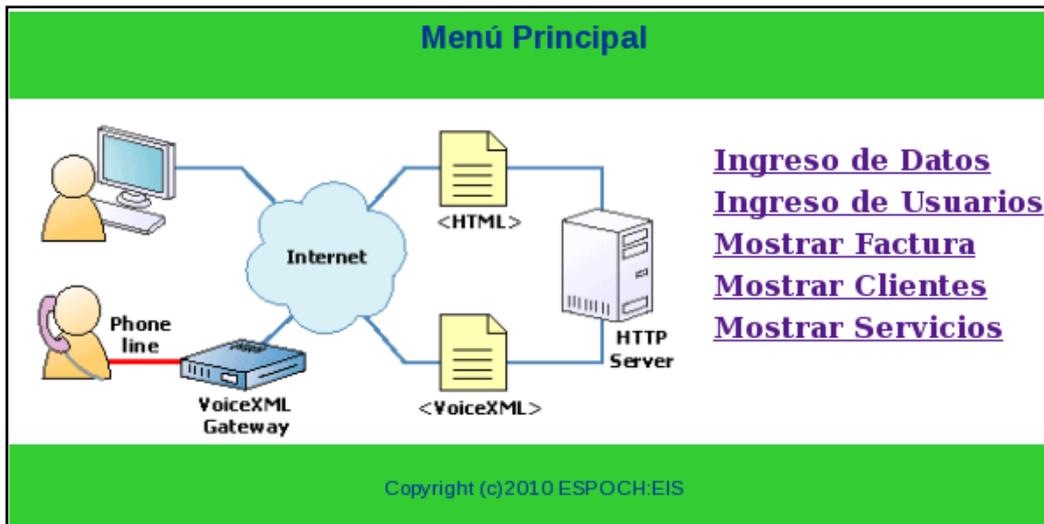


Gráfico IV.51: Pantalla del Menú Principal

#### 4.3.1.3 PANTALLA DE INGRESOS DE OPCIONES



Opciones Insertar

 [Insertar Línea Telefónica](#)  
[Insertar Servicios](#)  
[Insertar Factura](#)  
[Insertar Cliente](#)

[Regresar](#)

Gráfico IV.52: Pantalla del Ingreso de Opciones

#### 4.3.1.3.1 INGRESO DE CLIENTES



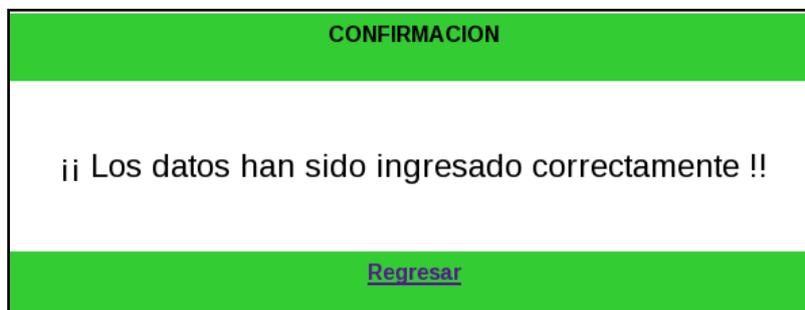
Insertar Cliente

 Cedula:   
Nombres:   
Apellidos:   
Dirección:   
Ciudad:

[Regresar](#)

Gráfico IV.53: Pantalla del Ingreso de Cliente

Luego del ingreso se mostrará la pantalla de confirmación de los datos ingresados.



CONFIRMACION

¡¡ Los datos han sido ingresado correctamente !!

[Regresar](#)

Gráfico IV:54: Pantalla del Confirmación

#### 4.3.1.3.2 INGRESO DE LÍNEAS TELEFÓNICAS

Insertar Línea Telefónica

Número Telefónico 087035576

Cédula del Propetario: 1803753708

Aceptar Cancelar

Regresar

Gráfico IV.55: Pantalla del Ingreso de Línea Telefónica

Luego del ingreso se mostrará la pantalla de confirmación de los datos ingresados.

CONFIRMACION

¡¡ Los datos han sido ingresado correctamente !!

Regresar

Gráfico IV.56: Pantalla del Confirmación

#### 4.3.1.3.3 INGRESO DE SERVICIO

Insertar Servicios

Id. Servicio 023

Descripcion minutos gratis

Valor 12

Estado 1

Aceptar Cancelar

Regresar

Gráfico IV.57: Pantalla del Ingreso de Servicios

Luego del ingreso se mostrará la pantalla de confirmación de los datos ingresados.

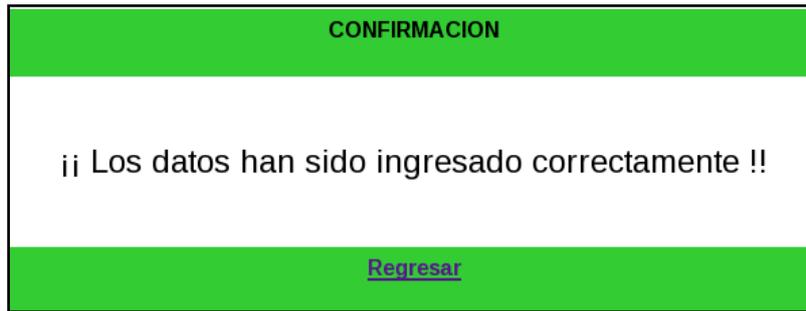


Gráfico IV.58: Pantalla del Confirmación

#### 4.3.1.4 PANTALLA DE GENERACIÓN DE FACTURA



Gráfico IV.59: Pantalla de Generación de Factura

#### 4.3.1.4.1 FACTURA TOTAL

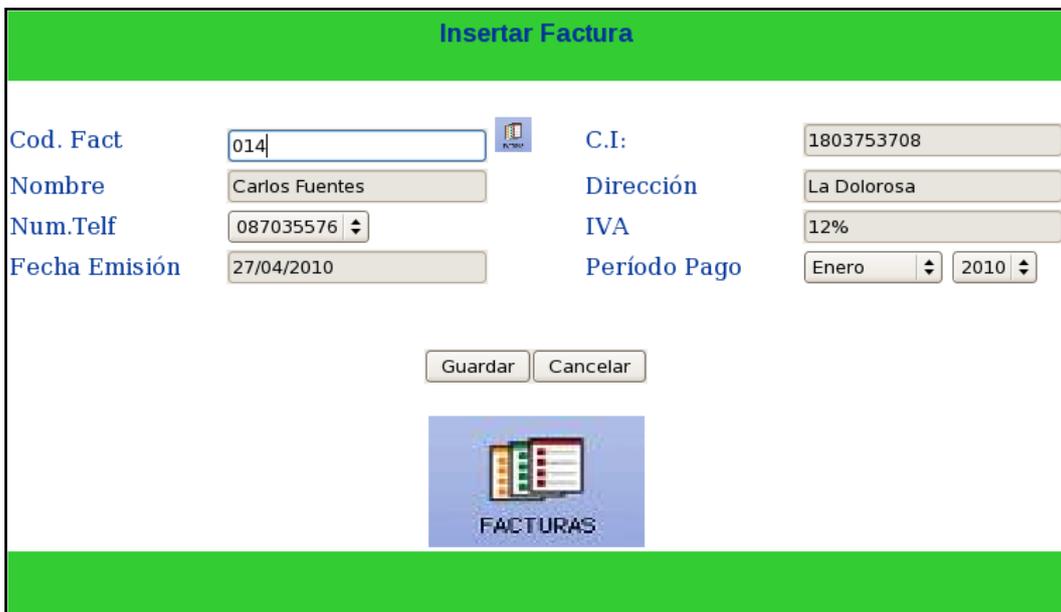


Gráfico IV.60: Pantalla de Ingreso de Datos en la Factura

**Insertar Factura**

Cod. Fact	014	C.I.	1803753708
Nombre	Carlos Fuentes	Direccion	La Dolorosa
Num.Telf	087035576	IVA	12%
Fecha Emision	27/04/2010	Periodo Pago	Enero 2010

Descripcion	Cantidad
minutos gratis	12
Pension Basica	24
Llamada Reg Automatica	125
Llamada a Movistar	145
Llamada a Porta	163
Consumo Local	25
Internet	24



**Gráfico IV.61: Pantalla de Ingreso de Servicios en la Factura**

**Insertar Factura**

Cod. Fact	014	C.I.	1803753708
Nombre	Carlos Fuentes	Dirección	La Dolorosa
Num.Telf	087035576	IVA	12%
Fecha Emisión	27/04/2010	Período Pago	Enero 2010

DESCRIPCION	CANTIDAD	VALOR
Pension Basica	24	148.8
Llamada Reg Automatica	125	2.5
Llamada a Movistar	145	14.5
Llamada a Porta	163	16.3
Consumo Local	25	0.25

Subtotal	182.35
Iva (12%)	21.882
TOTAL	204.232

[Regresar](#)

**Gráfico IV.62: Pantalla de Resultados de la Factura**

#### 4.3.1.5 PANTALLA PARA MOSTRAR DATOS

A continuación se muestra los datos que se encuentran almacenados en la base de datos.

##### 4.3.1.5.1 PANTALLA PARA MOSTRAR DATOS DE FACTURA

Al ingresar a esta opción el usuario escoge el número de la factura de la cual desea obtener toda la información. Como se muestra en el Gráfico IV.63.

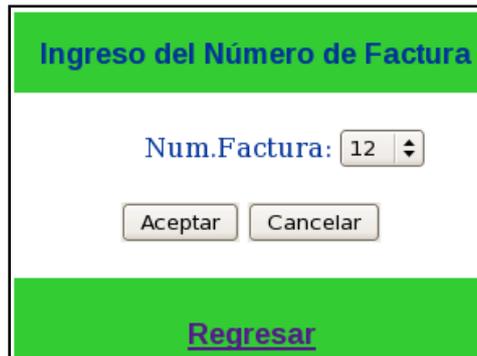
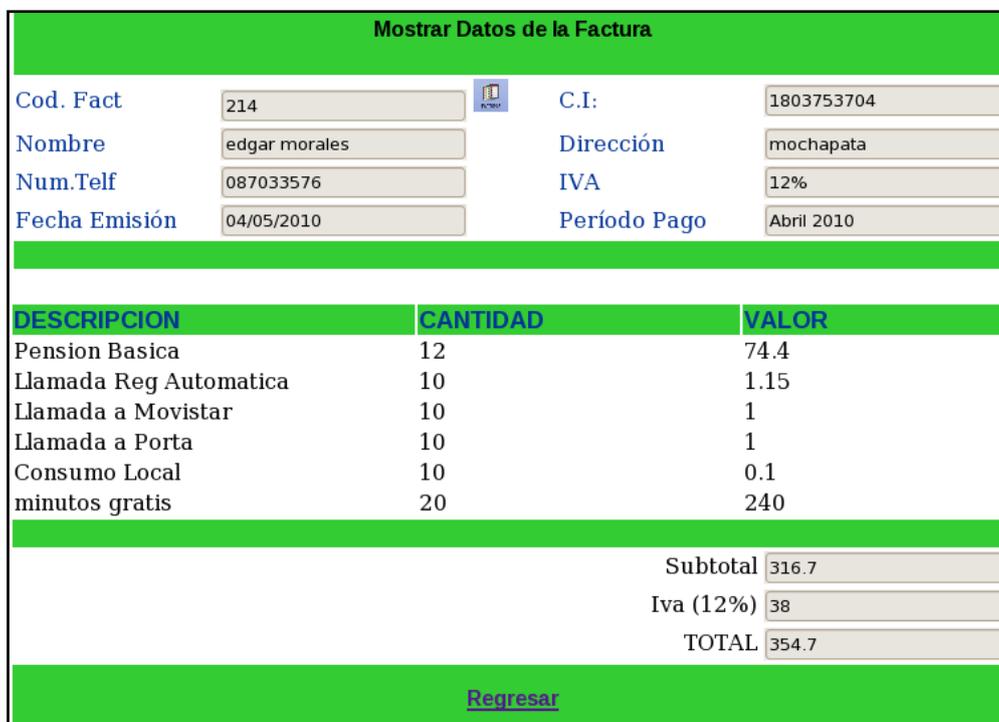


Gráfico IV.63: Pantalla de Búsqueda de Factura

Luego se muestra los datos obtenidos de la búsqueda. Como se muestra en Gráfico IV.64.



DESCRIPCION	CANTIDAD	VALOR
Pension Basica	12	74.4
Llamada Reg Automatica	10	1.15
Llamada a Movistar	10	1
Llamada a Porta	10	1
Consumo Local	10	0.1
minutos gratis	20	240

Subtotal	316.7
Iva (12%)	38
TOTAL	354.7

Gráfico IV.64: Pantalla de muestra de resultados de Factura

#### 4.3.1.5.2 PANTALLA PARA MOSTRAR DATOS DEL CLIENTE

En esta pantalla se escoge la cédula del cliente a mostrar. Como se muestra en el gráfico IV.65

Ingreso del Número de Cédula

CÉDULA ID.: 0603108770

Aceptar Cancelar

[Regresar](#)

Gráfico IV.65: Pantalla de Búsqueda de Cédula del Cliente

En el gráfico IV.66 se muestra la pantalla que despliega la información del cliente

Datos del Cliente

Nombres y Apellidos: Janeth Concha

Numero Telefonico: 032940026

Ciudad: Riobamba

Direccion: La Dolorosa

[Regresar](#)

Gráfico IV.66: Pantalla de muestra de Datos de Cliente

#### 4.3.1.5.3 PANTALLA PARA MOSTRAR DATOS DE SERVICIOS

En el Gráfico IV.67 muestra la pantalla para listar los servicios

Datos de los Servicios

DESCRIPCION	VALOR
intenet inalambrico	12
Lineas Telefonicas	45
Internet	10

[Regresar](#)

Gráfico IV.67: Pantalla de muestra de Datos de Cliente

## 4.3.2 DESARROLLO DE LA APLICACIÓN TELEFÓNICA

### 4.3.2.1 SCRIPT DE PRESENTACIÓN

El primer espacio de código de la aplicación VXML permite realizar la presentación de la aplicación al usuario a partir de un saludo y la información de la hora actual.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
xml:lang="es-ES">
<form>
  <block name="time">
    <var name="horas"/>
    <var name="minutos"/>
    <var name="saludo"/>
    <script>
      var d = new Date();
      horas=d.getHours();
      minutos=d.getMinutes();
      if(horas > 11)
      {
        if(horas > 18)
          saludo="Buenas noches"
        else
          saludo="Buenas tardes";
      }
      else
      {
        saludo="Buenas dias";
      }
    </script>
    <prompt> <value expr="saludo"/>
      Son las <value expr="horas"/>horas,
      <value expr="minutos"/> minutos
    </prompt>
  </block>
  <block>
    <prompt bargein="false">
      Bienvenidos al Sistema de Facturacion VoiceXML. a
      continuacion se presentan las siguientes opciones.
    </prompt>
    <goto next="menu.vxml"/>
  </block>
</form>
</vxml>
```

**Espacio Código IV.28: Archivo bienvenida.vxml**

#### 4.3.2.2 SCRIPT DE SELECCIÓN DEL MENÚ PRINCIPAL

El script menú permite al usuario mediante comando de voz escoger una de las opciones disponibles de nuestra aplicación VXML.

```
?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
<form id="menu">
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="12s"/>
<field name="item">
  <grammar mode="voice" type="application/verbio-text">
    saldos
    servicios
    ayuda
    salir
  </grammar>
  <prompt bargein="true">
    ,Menu principal.
    , saldos.
    , servicios.
    , ayuda.
    , salir.
    escoja su opcion por favor
  </prompt>
  <filled>
    <if cond="item =='saldos'">
      <goto next="planilla.vxml"/>
    </if>
    <if cond="item =='servicios'">
      <goto next="servicios.vxml"/>
    </if>
    <if cond="item =='ayuda'">
      <goto next="ayuda.vxml"/>
    </if>
    <if cond="item =='salir'">
      <goto next="salir.vxml"/>
    </if>
    <clear namelist="item"/>
  </filled>
</field>
</form>
</vxml>
```

Espacio Código IV.29: Archivo menu.vxml

#### 4.3.2.3 SCRIPT DE GENERACIÓN DE FACTURA

La opción saldos muestra un submenú con dos opciones, saldo total y detalle.

```
<?xml version="1.0" encoding="iso-8859-1"?>

<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">

<form id="menu">
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="10s"/>
  <field name="item">
    <grammar mode="voice" type="application/verbio-text">
      saldo total
      detalle
    </grammar>
    <prompt bargein="true">
      Este es un sub menu que tiene dos opciones:
      uno, saldo total.
      2, detalle.
      Escoja su opcion por favor.
    </prompt>
    <filled>
      <if cond="item =='saldo total'">
        <goto next="planillauno.vxml"/>
      </if>
      <if cond="item =='detalle'">
        <goto next="planillados.vxml"/>
      </if>
      <clear namelist="item"/>
    </filled>
  </field>
</form>
</vxml>
```

**Espacio Código IV.30: Archivo planilla.vxml**

#### 4.3.2.3.1 SCRIPT DE GENERACIÓN DE FACTURA TOTAL

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <form>
    <property name="inputmodes" value="voice"/>
    <property name="timeout" value="15s"/>
    <var name="caller" expr="session.connection.remote.uri"/>
    <block><prompt>Diga su numero telefonico, recuerde que su numero telefonico tiene 9
digitos</prompt></block>
    <field name="pin">
      <grammar src="builtin:grammar/digits"/>
      <filled>
        <prompt>Tu dijiste
          <value expr="pin"/>
        </prompt>
      </filled>
      <nomatch>Lo sentimos, no entendi eso. Intentelo de nuevo.
      <reprompt/>
      </nomatch>
    </field>
    <block>
      <submit next="planuno.php" method="post" namelist="caller pin"/>
    </block>
  </form>
</vxml>
```

#### **Espacio Código IV.31: Archivo planillauno.vxml**

El script anterior hace uso del script planuno.php para acceder a la base de datos.

```
<?php echo '<?xml version="1.0" encoding="iso-8859-1"?>';?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <property name="prompturl2" value="http://titus.i6net.org/tts/verbio/tts.php"/>
  <property name="promptposition" value="00" />
  <form>
    <block>
      <prompt bargein="false" xml:lang="text">
        NUMEROS CONSULTADOS;
      </prompt>
      <?php
        $caller = $_REQUEST["caller"];
        $telefono= $_REQUEST["pin"];
        $conexion = mysql_connect("localhost","root","");
        mysql_select_db("telefonica",$conexion);
        $sql ="select * from t_factura where txt_id_numero_tel='".$telefono.'" and int_estado=1";
        $temp = mysql_query($sql);
        $dat=mysql_fetch_assoc($temp);
        $fil= mysql_num_rows($temp);
        ?>
      </prompt>
      <prompt bargein="false">
        el saldo total a pagar por su planilla telefonica es:
      </prompt>
      <prompt bargein="false">
      <?php
        if($fil == 0 )
        {
          echo "Lo siento, ese numero no esta registrado en nuestra base de datos";
        }
        else
        {
          echo $dat["dl_total"]."dolares";
        }
      }?>
      </prompt>
      <goto next="planillatres.vxml"/>
    </block>
  </form>
</vxml>
```

**Espacio Código IV.32: Archivo planuno.php**

#### 4.3.2.3.2 SCRIPT DE GENERACIÓN DE FACTURA A DETALLE

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <form>
    <property name="inputmodes" value="voice"/>
    <property name="timeout" value="15s"/>
    <var name="caller" expr="session.connection.remote.uri"/>
    <field name="pin">
      <grammar src="builtin:grammar/digits"/>
      <prompt>Dime tu numero telefonico, por favor</prompt>
      <filled>
        <prompt>Tu dijiste
          <value expr="pin"/>
        </prompt>
      </filled>
      <nomatch count="1">Lo sentimos, no entendi eso. Intentelo de nuevo.
      <reprompt/>
      </nomatch>
    </field>
  </block>
  <submit next="plandos.php" method="post" namelist="caller pin"/>
</form>
</vxml>
```

#### **Espacio Código IV.33: Archivo planillados.vxml**

Para realizar la función correctamente se utiliza el siguiente código:

```
<?php echo '<?xml version="1.0" encoding="iso-8859-1"?>';?>
<xml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">

<property name="prompturl2" value="http://titus.i6net.org/tts/verbio/tts.php"/>
<property name="promptposition" value="00" />

<form>
  <block>
    <prompt bargein="false" xml:lang="text">
      DETALLE;

<?php
$caller = $_REQUEST["caller"];
$telefono= $_REQUEST["pin"];

$conexion = mysql_connect("localhost","root","");
mysql_select_db("telefonica",$conexion);
$sql="select * from t_factura where txt_id_numero_tel='".$telefono."'";
$temp = mysql_query($sql);
$dat=mysql_fetch_assoc($temp);
$num_fact=$dat["txt_id_factura"];

$sql1="select t_servicios.txt_descripcion as descripcion, t_items.cantidad as cantidad, t_items.total_item as total from
t_servicios inner join t_items on ($num_fact=txt_id_fact and t_items.txt_id_ser=t_servicios.txt_id_servicio);

$temp1= mysql_query($sql1);
$fila=mysql_num_rows($temp1);
?>
</prompt>
  <prompt bargein="false">
    Detalle de su planilla telefonica
  </prompt>
  <prompt bargein="false">
    <?php
    if($fila==0)
    {
      echo"Lo siento, ese numero no esta registrado en nuestra base de datos";
    }
    else
    {
      while ($row = mysql_fetch_array($temp1))
      {
        echo "\n".$row["descripcion"]."\n";
        echo "\n".$row["cantidad"]." minutos";
        echo "\n".$row["total"]." dolares\n";
      }
      echo "el valor del Subtotal es:";
      echo "\n".$dat["dl_subtotal"]." dolares \n";
      echo "el valor del iva es:";
      echo "\n".$dat["dl_iva"]." dolares \n";
      echo "el saldo total a pagar por su planilla telefonica es:";
      echo "\n".$dat["dl_total"]." dolares";
    }
  ?>
</prompt>
  <goto next="planillatres.vxml"/>
</block>
</form>
</vxml>
```

#### Espacio Código IV.34: Archivo plandos.php

Tanto el planuno.vxml y el plandos.vxml permita ir a un submenú el cual permitirá regresar al realizar la consulta de otro numero o salir del sistema.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
<form id="menu">
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="30s"/>
<field name="item">
  <grammar mode="voice" type="application/verbio-text">
    saldo
    menu
    salir
  </grammar>
  <prompt bargein="false">
    Si deseas realizar la consulta de otro numero telefonico, pronuncia
  </prompt>
  <prompt bargein="true">
    saldo
  </prompt>
  <prompt bargein="false">
    Si deseas regresar al menu principal, pronuncia
  </prompt>
  <prompt bargein="true">
    menu
  </prompt>
  <prompt bargein="false">
    Si deseas salir del sistema, pronuncia
  </prompt>
  <prompt bargein="true">
    salir
  </prompt>
  <emphasis>Escoje la opcion por favor</emphasis>
  <filled>
    <if cond="item =='saldo'">
      <goto next="planilla.vxml"/>
    </if>
    <if cond="item =='menu'">
      <goto next="menu.vxml"/>
    </if>
    <if cond="item =='salir'">
      <goto next="salir.vxml"/>
    </if>
    <clear namelist="item"/>
  </filled>
</field>
</form>
</vxml>
```

**Espacio Código IV.35: Archivo planillatres.vxml**

#### 4.3.2.4 CONSULTA A SERVICIOS

A continuación el script de la opción servicios.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <form>
    <block>
      <submit next="servicios.php" method="post" namelist="caller"/>
    </block>
  </form>
</vxml>
```

#### **Espacio Código IV.36: Archivo servicios.vxml**

El script anterior invoca el script service .php para acceder a la base de datos y proporcionar la información al usuario.

```
<?php echo '<?xml version="1.0" encoding="iso-8859-1"?>';?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <property name="prompturl2" value="http://titus.i6net.org/tts/verbio/tts.php"/>
  <property name="promptposition" value="00" />
  <form>
    <block>
      <prompt bargein="false" xml:lang="text">
        Servicios;
      </prompt>
      <?php
        $caller = $_REQUEST["caller"];
        $conexion = mysql_connect("localhost","root","");
        mysql_select_db("telefonica",$conexion);
        $sql = "select * from t_servicios where estado=1";
        $temp1= mysql_query($sql);

        ?>
      </prompt>
      <prompt bargein="false">
        La empresa VooiceXML ofrece los siguiente servicios:
      </prompt>
      <prompt bargein="false">
      <?php
        while ($row = mysql_fetch_array($temp1))
        {
          echo "\n".$row["txt_descripcion"]."\n";
          echo "el valor del servicio es";
          echo "\n".$row["dl_valor"]."\n";
          echo "dolares";
        }
        ?>
      </prompt>
      <prompt bargein="false">
        Gracias por visitarnos, proxicamente estaremos lanzando nuevos servicios. Regresamos al
      </prompt>
      <goto next="menu.vxml"/>
    </block>
  </form>
</vxml>
```

**Espacio Código IV.37: Archivo servicios.php**

#### 4.3.2.5 NAVEGACIÓN POR LA AYUDA

El script de la ayuda nos permite brindarle al usuario la forma de interactuar con el sistema.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
<form id="ayuda">
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="30s"/>
  <block>
    <prompt bargein="false">
      Hola, esta es la ayuda del Sistemas de Planillas Telefonicas Voice,xml. Donde existen cuatro
      opciones:

      Opcion uno del Menu Principal) Saldos,
      Esta es una opcion que se divide en un submenu:

      Opcion Uno del Submenu) Saldo Total,
      Tu puedes digitar el numero telefonico que deseas consultar y a continuacion se darÃ¡ a
      conocer el costo a pagar,nunca olvides que el numero telefonico tiene 9 digitos.

      Opcion dos del Submenu) Detalle,
      Usted puede conocer el detalle del consumo telefonico digitando el numero telefonico.

      Opcion dos del Menu Principal)Servicios,
      En esta opcion se da a conocer los servicios que posee la empresa, ANIMATE Y ENTERATE
      DE TODAS LAS NOVEDADES DE NUESTRA EMPRESA.

      Opcion tres del Menu Principal) Ayuda,
      Con esta opcion se puede tener una guia del sistema.

      Opcion cuatro del Menu Principal) Salir,
      Con esta opcion automaticamente tu puedes salir del sistema.
    </prompt>
    <goto next="opcion_ayuda.vxml"/>
  </block>
</form>
</vxml>
```

**Espacio Código IV.38: Archivo ayuda.vxml**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">

  <form id="menu">
    <property name="inputmodes" value="voice"/>
    <property name="timeout" value="30s"/>

    <field name="item">
      <grammar mode="voice" type="application/verbio-text">
        menu
        salir
      </grammar>
      <prompt bargein="false">
        Si deseas regresar al menu principal, pronuncia
      </prompt>
      <prompt bargein="true">
        menu
      </prompt>
      <prompt bargein="false">
        Si deseas salir del sistema, pronuncia
      </prompt>
      <prompt bargein="true">
        salir
      </prompt>
      <emphasis>Escoje la opcion por favor</emphasis>
      <filled>
        <if cond="item =='menu'">
          <goto next="menu.vxml"/>
        </if>
        <if cond="item =='salir'">
          <goto next="salir.vxml"/>
        </if>
        <clear namelist="item"/>
      </filled>
    </field>
  </form>
</vxml>
```

**Espacio Código IV.39: Archivo opcion\_ayuda.vxml**

#### 4.3.2.6 SALIR DEL SISTEMA

La opción salir permite al usuario finalizar el uso del sistema.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
<form id="salir">
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="30s"/>
  <block>
    <prompt bargein="false">
      Gracias por visitar nuestra aplicacion.Esperamos un pronto retorno.Adios
    </prompt>
  </block>
  <disconnect/>
</form>
</vxml>
```

**Espacio Código IV.40: Archivo salir.vxml**

## **CAPITULO V**

### **COMPROBACIÓN DE LA HIPOTESIS**

**5 4**

#### **5.1 INTRODUCCIÓN**

En esta sección se procederá a realizar la comprobación de hipótesis basado en los objetivos planteados en el proceso de la investigación, se evaluarán los requerimientos hardware y software mínimo de los clientes, necesarios para utilizar tanto las aplicaciones Web y VoiceXML, los mismos que serán calificados con valores enteros con estos resultados probaremos la validez de nuestra hipótesis.

Para ello se ha utilizado el estudio comparativo de la estadística descriptiva, que permitirá organizar y resumir los datos obtenidos de la realidad, e inferir conclusiones respecto de ellos.

## 5.2 DEMOSTRACIÓN DE LA HIPOTESIS

Mediante el estudio comparativo de la estadística descriptiva que permite decidir cuáles son los aspectos, las características o los atributos importantes que se desea observar y registrar para cada uno de los casos. En esta ocasión el análisis de los recursos hardware y software utilizados por la aplicaciones Web y VoiceXML.

Para evaluar los indicadores primero se definirá los valores que serán asignados a los elementos a evaluarse.

VALOR CUALITATIVO	VALOR CUANTITATIVO
Aplica	1
No aplica	0

**Tabla V.17: Valorización para el Análisis**

- **Aplica**

Este valor permite indicar que es utilizable el elemento

- **No aplica**

Con este valor indica la no utilización del elemento

## 5.3 ACCESO A LA INFORMACIÓN VÍA WEB

En la actualidad existen dos formas de acceso a la información como son: la web móvil y la web fija.

### 5.3.1 WEB MÓVIL <sup>15</sup>

Actualmente, existe una demanda cada vez mayor por parte de los usuarios en lo referente a una disponibilidad incondicional de la Web, pero la realidad en el mercado es otra ya que aunque la oferta de dispositivos móviles está creciendo de forma asombrosa en los últimos años, ofreciéndonos infinidad de dispositivos que permiten realizar operaciones que normalmente se hacen desde el equipo de sobremesa, existen limitaciones a la hora de acceder a los servicios desde esos dispositivos móviles. En la mayoría de las ocasiones, el resultado es una experiencia de usuario poco

---

<sup>15</sup> Web Móvil  
<http://www.w3c.es/Divulgacion/GuiasBreves/webmovil>

satisfactoria al encontrarnos con numerosos problemas para acceder a la Web desde los dispositivos móviles.

### 5.3.2 WEB FIJA

Una de las formas mas comunes para el acceso al internet son las conexiones físicas las mismas que utilizan un alto numero de recursos hardware y software.

- **Adaptadores:** Modems y otros dispositivos que permiten la conexión eléctrica entre el ordenador y el exterior. Desde el punto de vista que aquí nos interesa, generalmente este "exterior" está constituido por la red de voz y/o datos de la compañía telefónica, aunque en ocasiones, puede ser una red de TV por cable, una conexión vía radio con una estación terrestre, o incluso vía satélite.
- **Portadores** ("Carriers"): Líneas, generalmente telefónicas, que comunican el adaptador con el proveedor de servicios Internet. También incluimos en este apartado a las compañías que prestan este servicio de comunicación de datos.
- **Software:** Programas de comunicaciones de diversos tipos que, utilizando los protocolos adecuados, permiten conectar a la Red y utilizar los servicios que ofrece. Los más emblemáticos son los Navegadores (Browsers) que permiten acceder a la Web, aunque hay de muchos tipos (clientes de correo electrónico, de Chat, de transferencia de ficheros, etc).

#### 5.4 ACCESO A LA INFORMACION VIA VOZ

Las aplicaciones VoiceXML permiten ser accedidas mediante la voz desde distintos dispositivos, los cuales pueden ser fijos, móviles o teléfonos IP (hardphone o softphone).

<b>ALTERNATIVAS</b> <b>RECURSOS</b>	<b>APLICACIONES WEB</b>	<b>APLICACIONES VOICEXML</b>
<i>Web Fija</i>	Aplica	No aplica
<i>Web Móvil</i>	Aplica	No aplica
<i>Telefonía Fija</i>	No aplica	Aplica
<i>Telefonía Móvil</i>	No aplica	Aplica
<i>Hardphone</i>	No aplica	Aplica
<i>Softphone</i>	No Aplica	Aplica

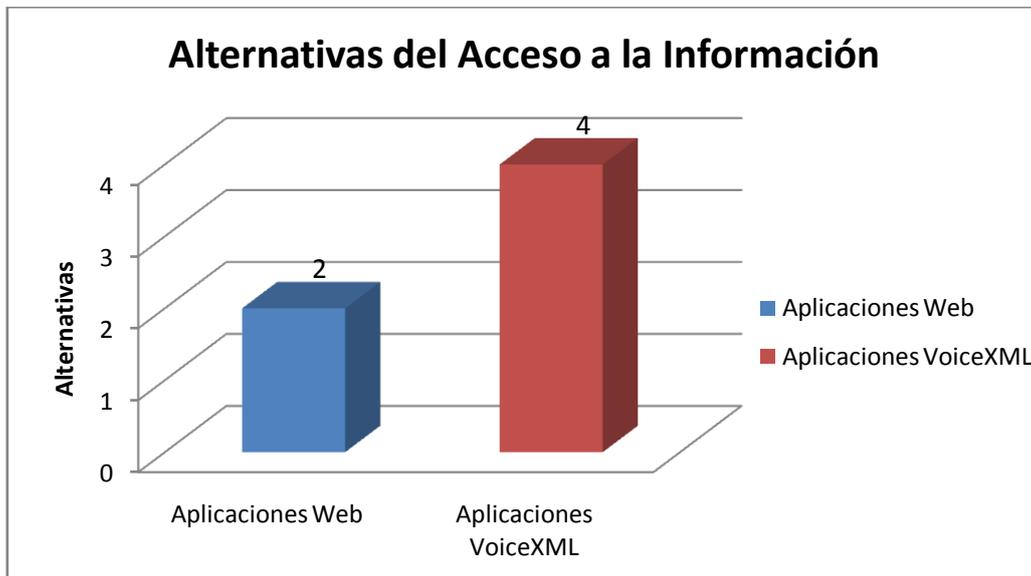
**Tabla V.18: Alternativas para el acceso a las Aplicaciones Web y VoiceXML**

En la siguiente tabla se asigna los valores cuantitativos de acuerdo a la calificación asignada anteriormente.

<b>ALTERNATIVAS</b> <b>RECURSOS</b>	<b>APLICACIONES WEB</b>	<b>APLICACIONES VOICEXML</b>
<i>Web Fija</i>	1	0
<i>Web Movil</i>	1	0
<i>Telefonia Fija</i>	0	1
<i>Telefonia Movil</i>	0	1
<i>Hardphone</i>	0	1
<i>Softphone</i>	0	1
<b>TOTAL</b>	<b>2</b>	<b>4</b>

**Tabla V.19: Valores Cuantificados de los Recursos Hardware**

Un agregado de las aplicaciones VoiceXML es la movilidad de los clientes tanto móviles como IP. Los clientes móviles permiten una movilidad geográfica dentro de las zonas de cobertura mientras tanto que los Hardphone y Softphone permiten una movilidad mundial.



**Gráfico V.68: Representación Gráficas de las alternativas de Acceso a la Información**

#### **5.4.1 REQUERIMIENTOS MINIMOS PARA CONECTARSE A INTERNET**

Para conectarse a Internet se requiere que su Computadora (PC) cumpla con ciertos requisitos pero como un mínimo las siguientes especificaciones:

##### **REQUERIMIENTOS MÍNIMOS HARDWARE**

Los requerimientos hardware mínimos necesarios para navegar a través de internet son los siguientes:

- Teclado
- Pantalla
- Mouse
- Procesador
- Memoria Ram
- Disco Duro
- Módem
- Línea telefónica

- Proveedor de acceso

#### **REQUERIMIENTOS MÍNIMOS SOFTWARE**

Los requerimientos software mínimos necesarios para que un cliente pueda navegar en internet son los siguientes:

- Sistema Operativo
- Navegado.
- Cliente de Correo Electrónico (Opcional)
- Cliente de Ftp (Opcional)
- Cliente de News (Opcional)

#### **5.4.2 REQUERIMIENTOS HARDWARE Y SOFTWARE DE LOS CLIENTES PARA ACCESO A LAS APLICACIONES VOICEXML**

- **TELÉFONO ANALÓGICO**

Para acceder a una aplicación VoiceXML desde un teléfono analógico únicamente se necesita el aparato telefónico y la línea.

- **TELÉFONO CELULAR**

Los elementos hardware de los teléfonos móviles son la pantalla, el teclado, la tarjeta de red.

- **TELÉFONO HARDPHONE**

El hardphone es un teléfono físico que posee una interfaz de red, sistema operativo y memoria de almacenamiento en cantidades pequeñas.

- **TELEFONO SOFTPHONE**

El softphone funciona al igual que un programa de computadora

#### **5.5 EVALUACIÓN DE RECURSOS HARDWARE**

A continuación evaluaremos los recursos hardware que utilizan los dispositivos que permiten acceder a las aplicaciones Web y VoiceXML.

<b>APLICACIONES</b> <b>RECURSOS HW</b>	<b>WEB</b>	<b>VoiceXML</b>
<i>Teclado</i>	Aplica	Aplica
<i>Mouse</i>	Aplica	No Aplica
<i>Pantalla</i>	Aplica	No Aplica
<i>Procesador</i>	Aplica	No Aplica
<i>Memoria Ram</i>	Aplica	No Aplica
<i>Disco Duro</i>	Aplica	No Aplica
<i>Proveedor ISP</i>	Aplica	No Aplica
<i>Línea Telefónica</i>	Aplica	Aplica
<i>ADSL</i>	Aplica	No Aplica
<i>Tarjeta de Red</i>	Aplica	No Aplica

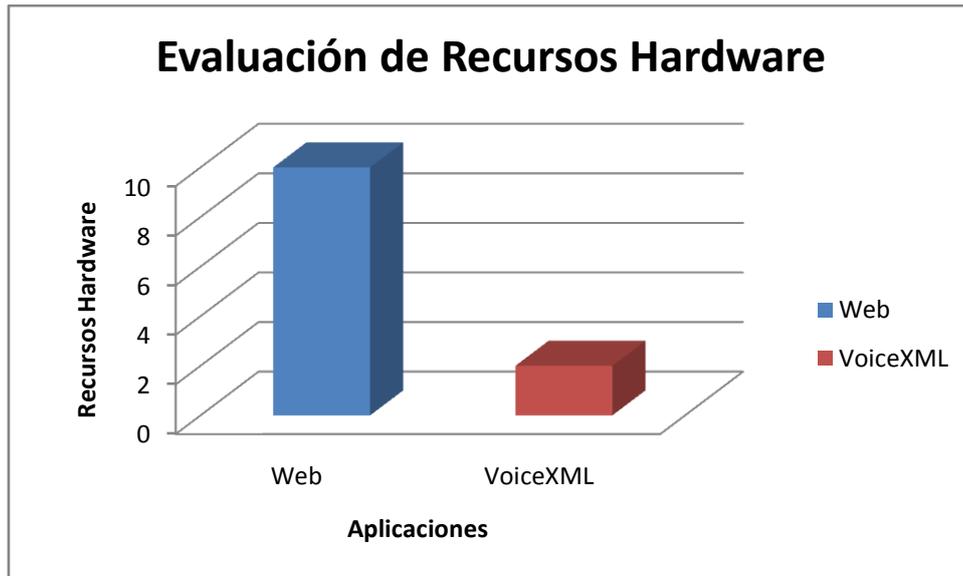
**Tabla V.20: Recursos hardware para el acceso a Aplicaciones**

En la tabla V.21 se asigna los valores cuantitativos de acuerdo a la calificación asignada anteriormente.

<b>APLICACIONES</b> <b>RECURSOS HW</b>	<b>WEB</b>	<b>VoiceXML</b>
<i>Teclado</i>	1	1
<i>Mouse</i>	1	0
<i>Pantalla</i>	1	0
<i>Procesador</i>	1	0
<i>Memoria Ram</i>	1	0
<i>Disco Duro</i>	1	0
<i>Proveedor ISP</i>	1	0
<i>Línea Telefónica</i>	1	1
<i>ADSL</i>	1	0
<i>Tarjeta de Red</i>	1	0
<b>TOTAL</b>	10	2

**Tabla V.21: Valores Cuantificados de los Recursos Hardware**

Basado en el estudio comparativo de la estadística descriptiva se determina que las aplicaciones VXML minimizan la utilización de recursos hardware para su utilización.



**Gráfico V.69: Evaluación de los Recursos Hardware**

### 5.6 EVALUACIÓN DE RECURSOS SOFTWARE

A continuación evaluaremos los recursos software que utilizan los dispositivos que permiten acceder a las aplicaciones Web y VoiceXML.

RECURSOS SW \ APLICACIONES	APLICACIONES	
	WEB	VoiceXML
<i>Sistema Operativo</i>	Aplica	No Aplica
<i>Navegador Web</i>	Aplica	No Aplica
<i>Cliente de Correo Electrónico</i>	Aplica	No Aplica
<i>Cliente FTP</i>	Aplica	No Aplica
<i>Cliente New</i>	Aplica	No Aplica
<i>Softphone</i>	No Aplica	Aplica

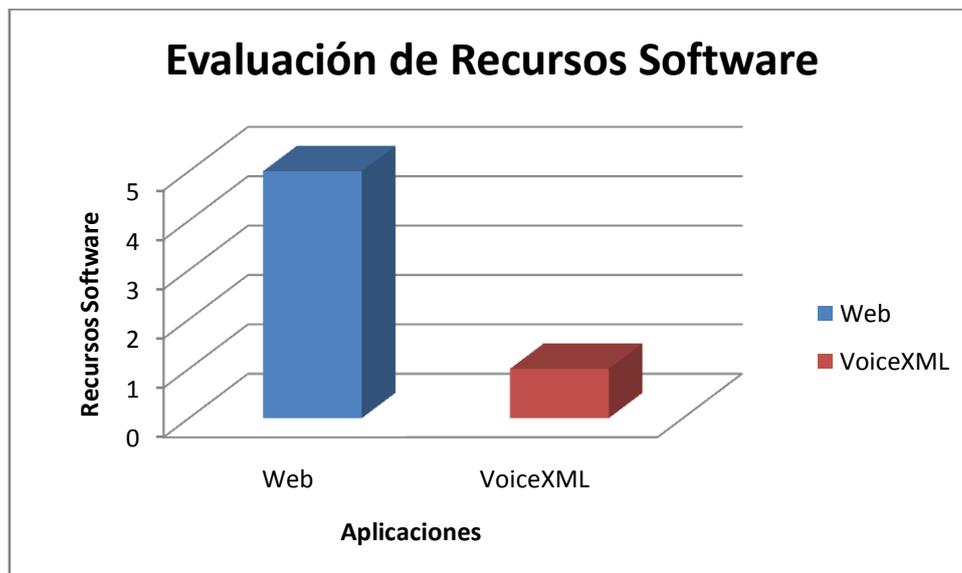
**Tabla V.22: Recursos Software para el Acceso de Aplicaciones**

En la tabla V.23 se asigna los valores cuantitativos de acuerdo a la calificación asignada anteriormente.

RECURSOS SW \ APLICACIONES	WEB	VoiceXML
	<i>Sistema Operativo</i>	1
<i>Navegador Web</i>	1	0
<i>Cliente de Correo Electrónico</i>	1	0
<i>Cliente FTP</i>	1	0
<i>Cliente New</i>	1	0
<i>Softphone</i>	0	1
<i>Total</i>	5	1

**Tabla V.23: Valores Cuantificados de los Recursos Software**

Basado en el estudio comparativo de la estadística descriptiva se determina que las aplicaciones VXML minimizan la utilización de recursos software para su utilización.

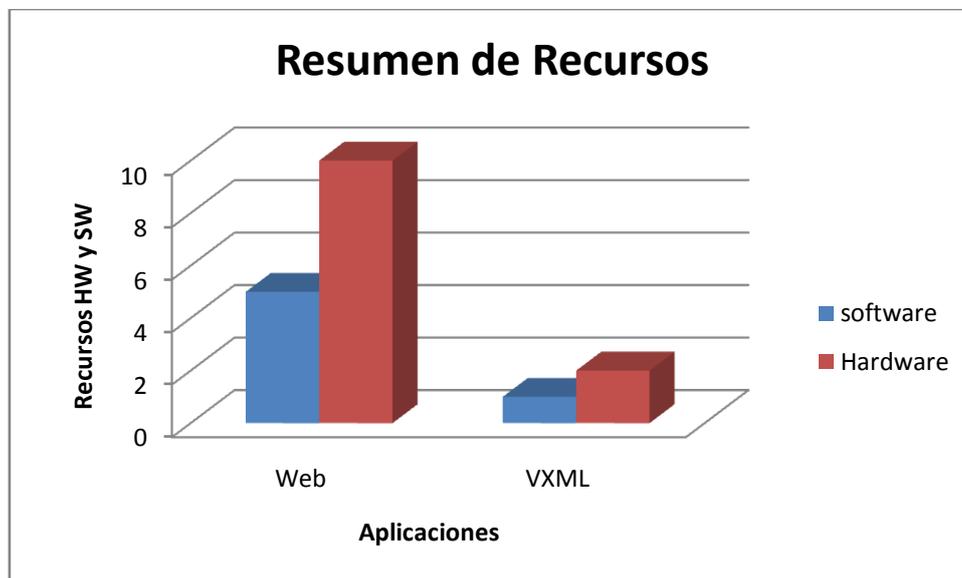


**Gráfico V.70: Evaluación de los Recursos Software**

## 5.7 RESULTADOS OBTENIDO

Las Aplicaciones VoiceXML permiten un mayor número de alternativas para el acceso a la información frente a las aplicaciones Web, como se puede verificar en grafico V.66 que únicamente tienen dos alternativas que permiten llegar a sus contenidos, los cuales se obtuvieron del estudio comparativo realizado anteriormente. Los datos para obtener estos resultados fueron investigados y tabulados previamente.

Una ventaja primordial en la difusión de las aplicaciones VoiceXML, es la disminución del empleo de recursos hardware y software para su utilización, como lo indica el grafico V.71, donde se presenta un resumen.

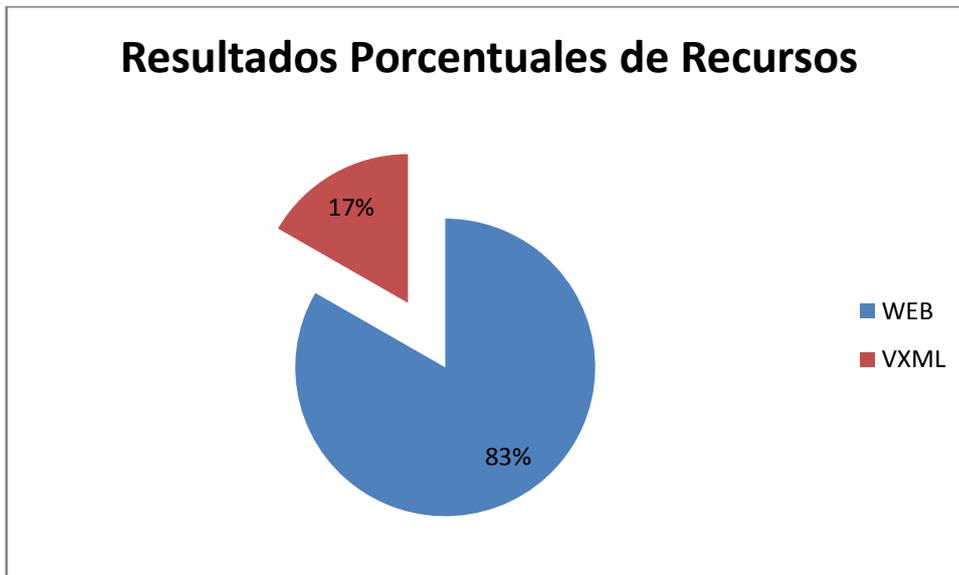


**Gráfico V.71: Resumen de Recursos**

Luego de haber realizado el resumen estadístico de recursos hardware y software, se desarrollará la representación porcentual de resultados de aplicaciones web y vxml. Tomando como muestra todos datos utilizados en el análisis anterior se ha realizado una suma de recursos hardware y software tanto de las aplicaciones Web como VoiceXML. El total de recursos evaluados son 18 de los cuales se han valorizado como un 100% para este análisis, siendo el total de la muestra. Como se muestra en la Tabla V.24.

<b>APLICACIÓN</b>	<b>RECURSO</b>	<b>PORCENTAJE (%)</b>
Web	15	83%
VoiceXML	3	17 %
<b>TOTAL</b>	<b>18</b>	<b>100 %</b>

**Tabla V. 24: Tabla Porcentual de Recursos**



**Gráfico V.72: Resumen Porcentual**

## CONCLUSIONES

- Mediante el estudio del estándar VoiceXML permitió desarrollar un prototipo para consulta de planillas telefónicas, funcional y eficiente.
- El estudio de la arquitectura del estándar VoiceXML ha permitido comprender las diferentes alternativas de uso en el desarrollo de aplicaciones VXML.
- El estudio de la programación en VXML permitió el desarrollo del prototipo telefónico planteado inicialmente.
- Con la adquisición de la Tarjeta Telefonica se logro intergrar el prototipo a la Red de Telefonía Pública y Móvil.
- El uso del estándar VoiceXML permite el desarrollo de aplicaciones informáticas que brindan al usuario información dinámica en formato de voz.
- El reconocimiento de voz ayuda a realizar aplicaciones que permiten simular un proceso de interacción hombre-maquina mediante palabras naturales proporcionadas por parte del usuario.
- La representación de la información en formatos de voz ayuda a que esta sea entendida de una mejor manera por los usuarios.
- La implantación de una aplicación VoiceXML permite al usuario acceder a la información por tres medios distintos teléfono convencional, teléfono celular y por ultimo teléfonos IP.
- En el desarrollo de aplicaciones VoiceXML se pueden combinar herramientas de software libre y propietario, minimizando costos de producción.
- La realización del presente estudio permite dejar abiertas nuevos campos de aplicación de este estándar como pueden ser aplicaciones comerciales, medicina y la mas importante el desarrollo de aplicaciones para personas con capacidades especiales.
- Las aplicaciones VoiceXML permiten un mayor número de alternativas de consulta de datos con la disminución del empleo de recursos hardware y software,

## RECOMENDACIONES

- Basados en el presente estudio, se recomienda realizar investigaciones mas detalladas sobre la manipulación de la información en un sistema gestor de base de datos que incluyan operaciones de ingreso, modificación y eliminación de la información aprovechando las bondades del estándar VoiceXML.
- Se recomienda migrar las aplicaciones Web a aplicaciones VoiceXML, esto ayudaría a que más usuarios accedan a la información con la utilización mínima de recursos hardware y software.
- Para obtener resultados exactos y con mayor productividad en el reconocimiento de la voz se recomienda el estudio de dos o mas motores de reconocimiento del habla que determinen el mas idóneo a utilizar en ambientes de producción.
- El estándar VoiceXML debe ser un referente en la formación de los ingenieros en sistemas informáticos por lo que se recomienda incluir el estudio del mismo en la malla curricular.

## RESUMEN

El objetivo de la investigación es estudiar el estándar VoiceXML, que representa un lenguaje de programación basado en etiquetas, permite realizar aplicaciones basadas en navegación por voz, reproducción de archivos de audio pre-grabados, reconocimiento voz, presentar al usuario información dinámica extraídas desde motores de base de datos y reproducir en forma de voz.

Para el desarrollo de esta investigación, se realizó la implementación de una centralita telefónica, que describe el proceso de instalación y configuración de la plataforma de pruebas para el estándar VoiceXML con todos elementos necesarios para su correcto funcionamiento, se utilizó motores para el reconocimiento y síntesis de voz, que permite una excelente funcionalidad del estándar. La aplicación consta de un menú que presenta opciones al usuario en forma de voz, el usuario escoge la opción mediante comandos de voz, la aplicación reconoce la información y procede a realizar nuevas actividades como navegar a otra opción o acceder a la base de datos del sistema para presentar la información al usuario mediante sonidos vocales.

Luego de realizar la implementación del prototipo de acceso a datos mediante el estándar VoiceXML, se determinó que las aplicaciones VoiceXML utilizan un 17% de recursos frente a las aplicaciones Web con un 83%, lo cual determina que el acceso a la información con VoiceXML utiliza recursos mínimos. Se recomienda la utilización de este estándar para futuras aplicaciones de voz.

## **SUMMARY**

This investigation deals with studying the standard VoiceXML which represents a programming language based on labels. It permits to carry out applications based on surfing through voice, prerecorded audio file reproduction, voice recognition and presenting the user dynamic information extracted from the database motors and reproducing in the voice form. For this investigation development, the implementation of a little telephone exchange describing the installation process and the test platform configuration were carried out for the standard VoiceXML with all the necessary elements for its correct functioning. Motors for voice recognition and synthesis permitting an excellent standard functioning were used. The application consists of a menu presenting the user options in the form of voice; the user chooses the option through voice commands; the application recognizes the information and carried out new activities such as surfing to other option or entering into the system database to present the user the information through vocal sounds. After carrying out the prototype implementation of data access through the standard VoiceXML, it was determined that the applications of VoiceXML use a 17% resources against the Web applications with 83% which determines that the access to the information with VoiceXML uses minimum resources. It is recommended to use this standard for future voice applications.

## GLOSARIO

IVR	Consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de transformación de texto a voz y el reconocimiento de respuestas simples, como "sí", "no" u otras
VoiceXML	Lenguaje extensible de marca de voz que proporciona un interface de programación de alto nivel de recursos de voz y telefonía para los desarrolladores de aplicaciones, proveedores de servicios y fabricantes de equipamientos. El lenguaje sigue todas las reglas sintácticas de XML con las semánticas que son las que soportan la creación de aplicaciones de voz interactivas
PSTN	Es el sistema estándar utilizado por las líneas de telefonía terrestre, o sea alambres de cobre que transportan señales de voz análogas
Teléfono IP	Dispositivo telefónico que parece un teléfono común, se conecta a Public Switched Telephone Network (PSTN o red común de telefonía) y tiene un puerto Ethernet para permitir las comunicaciones a una red de computadoras TCP/IP
Servidor Apache	El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual
Servidor Mysql	El servidor de bases de datos MySQL es la base de datos de fuente abierta más popular en el mundo. Su arquitectura lo hace extremadamente rápido y fácil de adaptar. Este servidor de bases de datos potente es usado por muchos programas de origen abierto incluyendo phpBB, osCommerce, y phpNuke.[
HTML	HTML (HyperText Markup Language) es el lenguaje de autor utilizado para crear documentos en la Red Global (World Wide Web). HTML define la estructura y estilo de una página Web a través del uso de una variedad de "tags" y atributos (atributes)

Text Speech	To	La conversión texto-voz es la generación, por medios automáticos, de una voz artificial que genera idéntico sonido al producido por una persona al leer un texto cualquiera en voz alta. Es decir, son sistemas que permiten la conversión de textos en voz sintética.
Automatic Specch Recognition		Es una parte de la Inteligencia Artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras electrónicas
PBX		Un PBX se refiere al dispositivo que actúa como una ramificación de la red primaria pública de teléfono, por lo que los usuarios no se comunican al exterior mediante líneas telefónicas convencionales, sino que al estar el PBX directamente conectado a la RTC (red telefónica pública).
Asterisk		Es un programa de software libre (bajo licencia GPL) que proporciona funcionalidades de una central telefónica (PBX). Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una RDSI tanto básicos como primarios.
Dahdi-linux		Contiene los módulos de kernel necesario para poder utilizar las tarjetas de comunicaciones
Dahdi-tools		Son las aplicaciones necesarias para cargar la configuración hacer tests a algunas tarjetas, y algunas cosas más que se irán añadiendo poco a poco.
Gramáticas		La Gramática es el estudio de las reglas y principios que regulan el uso de las lenguas y la organización de las palabras dentro de una oración
PHP		PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz

de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

#### Openvxi

OpenVXI es una fuente abierta de herramientas portables VoiceXML. Es un componente de un navegador de voz, y proporciona APIs para servicios de la plataforma: reconocimiento de voz, síntesis de voz, y servicios de telefonía.

## BIBLIOGRAFÍA

- **Analog Phone, PC**

[http://www.pcmag.com/encyclopedia\\_term/0,2542,t=analog+phone&i=37753,00.asp](http://www.pcmag.com/encyclopedia_term/0,2542,t=analog+phone&i=37753,00.asp)

(2009/10/02)

- **Asterisk**

<http://www.asterisk.org/>

(2009/08/29)

- **Asterisk The Software PBX, Knotic**

[http://www.knotik.com/fich\\_documentos/7\\_arquitectura\\_asteriskpbx](http://www.knotik.com/fich_documentos/7_arquitectura_asteriskpbx)

(2009/08/10)

- **Centos 5.4**

<http://www.alcancelibre.org/article.php/disponible-centos-5-4>

(2009/10/28)

- **Central Telefonía IP, Tics con software libre**

<http://portal2.conlinux.net/centralIP>

(2009/10/21)

- **Cómo funcionan los teléfonos celulares, monografias.com**

<http://www.monografias.com/trabajos14/celularhist/celularhist.shtml>

(2009/10/05)

- **Configuración de una Pc para acceder a Internet**

<http://www.unsl.edu.ar/guia/guia.htm#hardware>

(2010/04/29)

- **Dispositivos de Acceso a Internet**

<http://www.zator.com/Internet/A7.htm>

(2010/04/29)

- **Ejemplos Voicexml**

<http://www.w3c.es/presentaciones/2004/1118-seminariommi-ja/22.html>

(2009/12/10)

- **Estudio Comparativo**

<http://www2.uiah.fi/projects/metodi/272.htm>

(2010/05/06)

- **Getting started with VoiceXML 2.0, Dave Raggett, revised 14th November 2001.**

<http://www.w3.org/Voice/Guide/#examples>

(2009/12/14)

- **Guía Breve de Web Móvil**

<http://www.w3c.es/Divulgacion/GuiasBreves/webmovil>

(2010/04/29)

- **Guía de instalación de los motores ASR/TTS de Verbio, Verbio Technologies S.L**

<http://www.verbio.com/webverbio3/html/reference/README-INSTALL-VERBIO.txt>

(2009/09/10)

- **Instalación y Configuración de Asterisk, Moise Silva**

[http://www.moythreads.com/manual\\_de\\_asterisk.pdf](http://www.moythreads.com/manual_de_asterisk.pdf)

(2009/08/08)

- **I6NET ofrece VXi\* para crear servicios de videollamada 3G/IP**

<http://noticias.aecoma.org/aplicaciones-soluciones/video/i6net-ofrece-vxi-para-crear-servicios-de-videollamada-3gip/>

(2010/03/21)

- **LA NUEVA ERA DE LA VOZ**

[http://www.informatizate.net/articulos/la nueva era de la voz parte 02 12052004.html](http://www.informatizate.net/articulos/la_nueva_era_de_la_voz_parte_02_12052004.html)

(2009/08/27)

- **Manual de Instalación y Configuración de un Servidor Asterisk, Luis Felipe Martínez, Wilson Teran T.**

[http://www.uninorte.edu.co/divisiones/ingenierias/Dpto\\_Sistemas/lab\\_redes/upload/file/MANUAL%20DE%20INSTALACION%20Y%20CONFIGURACION%20DE%20UN%20SERVIDOR%20ASTERISK.pdf](http://www.uninorte.edu.co/divisiones/ingenierias/Dpto_Sistemas/lab_redes/upload/file/MANUAL%20DE%20INSTALACION%20Y%20CONFIGURACION%20DE%20UN%20SERVIDOR%20ASTERISK.pdf)

(2009/11/05)

- **Manual de Referencia de Verbio, VoiceWeb, Verbio Technologies S.L**

[http://www.verbio.com/webverbio3/html/reference/pdf/vvoiceweb\\_guide\\_es.pdf?PHPSESSID=a38dbd5824c6e92ebd99a77dcf664138](http://www.verbio.com/webverbio3/html/reference/pdf/vvoiceweb_guide_es.pdf?PHPSESSID=a38dbd5824c6e92ebd99a77dcf664138)

(2009/09/12)

- **MySQL**

<http://www.mysql.com/>

(2009/11/24)

- **Panorama de los Sistemas de Texto a Voz**

[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/moreno\\_a\\_ga/capitulo1.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/moreno_a_ga/capitulo1.pdf)

(2009/08/24)

- **Pattern-oriented Software Architecture, Michael Kircher, Prashant Jain**

<http://elvex.ugr.es/decsai/java/pdf/AE-Telco.pdf>

(2010/01/9)

- **PBX**

[http://www.902112505.com/andramari/curso\\_telecomunicaciones/zdescargar/documentacion.pdf](http://www.902112505.com/andramari/curso_telecomunicaciones/zdescargar/documentacion.pdf)

(2010/01/12)

- **PHP**

<http://www.php.net/>

(2009/12/30)

- **Probando las tarjetas openvox, Julián Menéndez**

<http://www.julianmenendez.es/probando-tarjetas-openvox-a400p-b200p/>

(2009/12/15)

- **PSTN e ISDN, Abril Estrella de la Rosa Miranda, Edgar Raúl Carballo Vega, César Arenas Avilés**

[http://hosting.udlap.mx/profesores/luisg.guerrero/Cursos/IE445/Presentaciones/Pres\\_1\\_PSTNISDN\\_Wapeches.ppt](http://hosting.udlap.mx/profesores/luisg.guerrero/Cursos/IE445/Presentaciones/Pres_1_PSTNISDN_Wapeches.ppt)

(2010/02/14)

- **Sistema de la Información basadas en VoiceXML, Pineda Raul**

<http://www.wisis.ufg.edu.sv/www.wisis/documentos/TE/005.74-M79d/005.74-M79d-Capitulo%20II.pdf>

(2009/08/29)

- **Tarjetas Openvox A400P, Fonologic,**

[http://www.fonologic.net/Tarjetas\\_Openvox\\_A400P.html](http://www.fonologic.net/Tarjetas_Openvox_A400P.html)

(2009/10/12)

- **Teléfono Móvil**

<http://www.bilmo.com/articulos/2005/10/telefono-movil.html>

(2010/04/29)

- **Telefono, wikipedia**

<http://es.wikipedia.org/wiki/Tel%C3%A9fono>

(2010/02/10)

- **VoiceXML Browser Manual Version 4.3, i6net**

<http://www.i6net.com/download/76/>

(2009/09/14)

- **VoiceXML Tutorial, BeVocal**

<http://cafe.bevocal.com/docs/tutorial/tutorial.pdf>

(2010/02/20)

- **VXlasterisk Interprete de VoiceXML**

<http://www.i6net.com>

(2009/10/14)

# **ANEXO 1**

Archivos de Configuración

## extensions.conf

```
; Static extension configuration file, used by
; the pbx_config module. This is where you configure all your
; inbound and outbound calls in Asterisk.
;
; The "General" category is for certain variables.
;
[general]
static=yes
writeprotect=no

[globals]
;VXML_URL2=file:///root/index.vxml
VXML_VIDEO_CODEC=h263p
VXML_VIDEO_SIZE=qcif
VXML_VIDEO_PROFILE=${VXML_VIDEO_CODEC}@${VXML_VIDEO_SIZE}/fps=7/kb=40/qmin=4
/qmax=8/gs=10

[default]
; Echo test
exten => 600,1,Wait(1)
exten => 600,2,Answer
exten => 600,3,BackGround(beepbis)
exten => 600,4,Echo

; Netann support
exten => dialog,1,Answer
exten => dialog,2,Vxml(${URL})
exten => dialog,3,Hangup

; SIP users
;exten => _user.,1,Dial(SIP/${EXTEN})
exten => _user.,1,Set(__VXML_NUMBER=${EXTEN})
exten => _user.,n,Set(__VXML_LOCAL=@dialpeer)
exten => _user.,n,vxml(@transcoder)

; Hosting
;exten => h,1,Hangup()
exten => s,1,Set(__VXML_LOCAL=0000)
exten => s,n,vxml()
```

```

exten => _X.,1,Set(__VXML_LOCAL=${EXTEN})
;exten => _X.,1,NoOp(${CHANNEL(transfercapability)})
exten => _X.,n,vxml(@transcoder)
exten => vxml,1,NoOp(${CHANNEL(transfercapability)})
exten => vxml,n(setid),GotoIf("${VXML_ID}" != "")?setparam
exten => vxml,n,Set(__VXML_ID=${UNIQUEID})
exten => vxml,n(setparam),GotoIf("${VXML_PARAM}" != "AMD"?capacity)
exten => vxml,n,Amd(5000)
exten => vxml,n,Set(__VXML_PARAM=${AMDSTATUS})
exten => vxml,n(capacity),GotoIf("${CHANNEL(transfercapability)}=VIDEO"?video:digital)
exten => vxml,n(digital),GotoIf("${CHANNEL(transfercapability)}=DIGITAL"?video:audio)
exten => vxml,n(video),Answer()
exten => vxml,n,NoCDR()
exten => vxml,n,h324m_gw(vxml_gw@default/n)
exten => vxml,n(audio),NoOp()
;exten => vxml,n,Set(CALLERID(number)=${CALLERID(number):-4})
exten => vxml,n,Vxml()
exten => vxml_gw,1,Answer()
;exten => vxml_gw,n,Set(CALLERID(number)=${CALLERID(number):-4})
exten => vxml_gw,n,h324m_gw_answer()
exten => vxml_gw,n,Set(CHANNEL(transfercapability)=VIDEO)
exten => vxml_gw,n,WaitForVideo(5)
exten => vxml_gw,n,Vxml()
;exten => transcoder,1,goto(default,transcoder,transcodeflash)
exten => transcoder,1,NoOp(${PEER_VIDEO_PROFILE})
exten => transcoder,n,GotoIf("${PEER_VIDEO_PROFILE}" != "" &
${PEER_VIDEO_PROFILE}" != "${VXML_VIDEO_PROFILE}")?transcodepeer)
exten => transcoder,n,GotoIf("${CHANNEL(channeltype)}" == "RTMP"?transcodeflash)
exten => transcoder,n,GotoIf("${VXML_VIDEO_CODEC}" == "h263sorenson"?transcodeflash)
exten => transcoder,n,GotoIf("${CHANNEL(videonativeformat)}" ==
"${VXML_VIDEO_CODEC}")?goto)
exten => transcoder,n,GotoIf("${CHANNEL(videonativeformat)}" == "h263"?transcodeh263)
exten => transcoder,n,GotoIf("${CHANNEL(videonativeformat)}" ==
"h263p"?transcodeh263p)
exten => transcoder,n,GotoIf("${CHANNEL(videonativeformat)}" == "h264"?transcodeh264)
exten => transcoder,n(goto),goto(default,vxml,1)
exten => transcoder,n(transcodepeer),Set(__VXML_TYPE=VIDEO)
exten => transcoder,n,NoCDR()
exten =>
transcoder,n,transcode(${VXML_VIDEO_PROFILE},vxml@default,${PEER_VIDEO_PROFILE})
exten => transcoder,n,Hangup()
exten => transcoder,n(transcodeflash),Set(__VXML_TYPE=VIDEO)
exten => transcoder,n,NoCDR()
exten =>
transcoder,n,transcode(${VXML_VIDEO_PROFILE},vxml@default,h263sorenson@${VXML_VID

```

```

EO_SIZE}/fps=7/kb=40/qmin=4/qmax=8/gs=10)
exten => transcoder,n,Hangup()
exten => transcoder,n(transcodeh263),Set(__VXML_TYPE=VIDEO)
exten => transcoder,n,NoCDR()
exten =>
transcoder,n,transcode(${VXML_VIDEO_PROFILE},vxml@default,h263@${VXML_VIDEO_SIZE}/
fps=7/kb=40/qmin=4/qmax=8/gs=10)
exten => transcoder,n,Hangup()
exten => transcoder,n(transcodeh263p),Set(__VXML_TYPE=VIDEO)
exten => transcoder,n,NoCDR()
exten =>
transcoder,n,transcode(${VXML_VIDEO_PROFILE},vxml@default,h263p@${VXML_VIDEO_SIZE}
/fps=7/kb=40/qmin=4/qmax=8/gs=10)
exten => transcoder,n,Hangup()
exten => transcoder,n(transcodeh264),Set(__VXML_TYPE=VIDEO)
exten => transcoder,n,NoCDR()
exten =>
transcoder,n,transcode(${VXML_VIDEO_PROFILE},vxml@default,h264@${VXML_VIDEO_SIZE}/
fps=10/kb=150/qmin=4/qmax=8/gs=10)
exten => transcoder,n,Hangup()
;exten => recorder,1,mp4recorder(mp4,,V2,transcoder@default/n)
exten => recorder,1,mp4recorder(mp4,,V2,vxml@default/n)
exten => recorder,n,Hangup()
exten => dial3g,1,GotoIf("${VXML_VIDEO_CODEC}" != "h263p"?transcode)
exten => dial3g,n,GotoIf("${VXML_VIDEO_SIZE}" != "qcif"?transcode)
exten => dial3g,n(goto),goto(default,dial3g_call,1)
exten =>
dial3g,n(transcode),transcode(h263p@qcif/fps=5/kb=40/qmin=4/qmax=8/gs=10,dial3g_call@
default,${VXML_VIDEO_PROFILE})
exten => dial3g_call,1,h324m_call(dial3g_gw@default)
exten => dial3g_gw,1,NoCDR()
exten => dial3g_gw,n,Set(CHANNEL(transfercapability)=VIDEO)
exten => dial3g_gw,n,Dial(DAHDI/g1/${VXML_NUMBER},${VXML_CONNECTTIMEOUT})
exten => dialsip,1,transcode(,dialsip_gw@default/n,noamr)
exten => dialsip_gw,1,NoCDR()
exten => dialsip_gw,n,Dial(SIP/${VXML_NUMBER},${VXML_CONNECTTIMEOUT})
exten =>
dialpeer,1,Set(PEER_VIDEO_PROFILE=${SIPPEER(${VXML_NUMBER},chanvar[PEER_VIDEO_PRO
FILE]))
exten => dialpeer,n,GotoIf("${PEER_VIDEO_PROFILE}" == "")?goto)
exten => dialpeer,n,GotoIf("${PEER_VIDEO_PROFILE}" !=
"${VXML_VIDEO_PROFILE}")?transcode)

```

```
exten => dialpeer,n(goto),goto(default,dialpeer_gw,2)
exten =>
dialpeer,n(transcode),transcode(${PEER_VIDEO_PROFILE},dialpeer_gw@default/n,${VXML_VIDEO_PROFILE})
exten => dialpeer_gw,1,NoCDR()
exten => dialpeer_gw,2,Dial(SIP/${VXML_NUMBER},${VXML_CONNECTTIMEOUT})
exten =>
dialh263,1,transcode(h263@qcif/fps=5/kb=40/qmin=4/qmax=8/gs=10,dialh263_gw@default/n,${VXML_VIDEO_PROFILE})
exten => dialh263_gw,1,NoCDR()
exten => dialh263_gw,n,Dial(${VXML_NUMBER},${VXML_CONNECTTIMEOUT})
exten => dialcamera,1,transcode(dialcamera_gw@default/n,${VXML_VIDEO_PROFILE})
exten => dialcamera_gw,1,NoCDR()
;exten => dialcamera_gw,n,Set(VXML_NUMBER=rtsp://192.168.1.3:554:80)
exten => dialcamera_gw,n,Answer()
exten => dialcamera_gw,n,camera(${VXML_NUMBER})
```

```
[LINEA]
;FXO
exten => s,1,Answer()
exten => s,n,Wait(1)
exten => s,n,Vxml(http://localhost/bienvenida.vxml)
exten => s,n,Hangup()
```

## chan\_dahdi.conf

```
[channels]
language=es
context=incoming
switchtype=euroisdn
usercallid=yes
hidecallerid=no
musiconhold=default
callwaiting=yes
usecallingpres=yes
threewaycalling=yes
transfer=yes
inmediate=no
canpark=yes
cancallforward=yes
callreturn=yes
echocancel=no
echocancelwhenbridged=no
;rxgain=0.0
;txgain=0.0
callerid=asreceived
overlapdial=no
pridialplan=local
prilocaldialplan=local
nationalprefix=0
internationalprefix=00
;group=1
;signalling=pri_cpe
;context=default
;channel=1-15
;channel=17-31
;group=2
;signalling=pri_cpe
;context=default
;channel=32-46
;channel=48-62
#include dahdi-channels.conf
```

## **dahdi-channels.conf**

```
; Autogenerated by /usr/sbin/dahdi_genconf on Tue Apr 13 13:41:42 2010
; If you edit this file and execute /usr/sbin/dahdi_genconf again,
; your manual changes will be LOST.
; Dahdi Channels Configurations (chan_dahdi.conf)
;
; This is not intended to be a complete chan_dahdi.conf. Rather, it is intended
; to be #include-d by /etc/chan_dahdi.conf that will include the global settings
;
; Span 1: WCTDM/4 "Wildcard TDM400P REV E/F Board 5" (MASTER)
;;; line="2 WCTDM/4/1 FXSKS"
signalling=fxs_ks
callerid=asreceived
group=0
context=LINEA
channel => 2
callerid=
group=
context=LINEA
```

## vxml.conf

; VoiceXML Configuration

[general]

autoanswer=yes

wavcodec=gsm

videosilence=

audiosilence=

dialformat=SIP/%s

speechprovider=verbio

speechscore=50

[control]

forward=#

reverse=\*

stop=123456789

pause=

restart=0

skipms=5000

[license]

max=1

video=yes

speech=yes

texttospeech=yes

externals=yes

key=evaluation

[account0]

name=demos

number=\*

url=http://192.168.1.4/bienvenida.vxml

max=1

speech=yes

dialformat=SIP/%s@myaccount

[account1]

name=audiovideo

number=1234

url=http://localhost/vxml/audio.vxml

urlvideo=http://localhost/vxml/video.vxml

max=1

speech=no

dialformat=SIP/%s

# verbio.conf

```
;
; Verbio ASR and TTS engines Configuration
;
;
;
; General options ;
;
;

[general]
;
; Primary Voxserver (default: 127.0.0.1)
;
primary_vox_server      = 127.0.0.1
;
; Backup voxserver
;
backup_vox_server       = 127.0.0.1
;
; Network Timeout (seconds)
; default 5
;
net_timeout              = 5
;
; TTS Engine Options ;
;
[tts]
;
; Default TTS language (depends on your verbio setup)
; If you do not want to load TTS, comment this option.
default_language        = es      ; check Verbio sdk documentation
                           ; for available languages.
;
; Default TTS speaker (depends on your verbio setup)
;
;default_speaker        = laura
;
; Default path to prompt files (text files that you want to be readed by Verbio TTS)
;
text_prompts_path       = /var/lib/asterisk/verbio/text
;
; Misc options
; Delays in ms
;
init_delay               = 300
```





```
; Path to recorded files
; default /var/lib/asterisk/verbio/audio
mark_recorded_files=0
recorded_files_path    = /var/lib/asterisk/verbio/audio
recorded_files_exten=wav
```

## modules.conf

```
;
; Module Loader Configuration
;
[modules]
autoload=yes
load => res_speech.so
noload => res_speech_lumenvox.so
load => res_speech_verbio.so
load => chan_dahdi.so
noload => chan_gtalk.so
noload => res_jabber.so
noload => chan_ss7.so
```

# client.cfg

```
# =====  
#  
# Copyright (c) 2001-3 SpeechWorks International  
#  
# Configuration file for SpeechBrowser client application  
#  
# Rules:  
# - Lines beginning with '#' are considered comments and ignored  
# - No comments are supported within a line (following other items)  
# - Each line is made of three items, a Name, a Type and a Value  
# - These three items (strings) can be separated by tabs or spaces  
# - Only the Value string can contain spaces, except trailing spaces  
#   which are ignored  
# - All items are case-sensitive  
# - Supported types are 'Environment', 'VXIString', 'VXIInteger',  
#   'VXIFloat' and 'VXIPtr'  
# - The 'Environment' type is used to set and remove an environment  
#   variable  
# - Types other than 'Environment' indicate you want to set a map  
#   property  
# - All properties will be passed as a single map argument to SB  
#   functions  
# - The value for the 'VXIPtr' type is ignored and considered as NULL  
# - Environment variables set here will apply not only to the script  
#   environment, but to the real application as well  
# - To remove a variable from the environment, supply no Value for it  
# - To use a variable within the script, use the syntax '$(VARIABLENAME)'  
# - Variables can only be used within Value items, not in Names and Types  
#  
# Examples:  
#           Name           Type      Value  
#           -----  
# Set an integer property: myModule.myIntegerKey  VXIInteger 1234  
# Set a string property:  myModule.myStringKey   VXIString Any string  
# Set an env. variable:   MY_VARIABLE           Environment C:\TEMP;D:\  
# Remove an env. variable: MY_EX_VARIABLE      Environment  
# Use an env. variable:   myModule.myEnvKey    VXIString $(MY_VARIABLE)  
#  
# You can use several variables within a Value: $(TYPE)//$(DRIVE)/$(PATH)  
#  
# =====
```

```

#####
# Overridden environment variables #
#####
# SWISBSDK Environment C:\Progra~1\SpeechWorks
#####
# Base client configuration #
#####
#####
### REC, PROMPT and TEL implementation #
#####
client.rec.implementation      VXIString voicebrowser
client.tel.implementation      VXIString voicebrowser
client.prompt.implementation   VXIString voicebrowser
### Inet and Write-back cache
client.cache.cacheDir          VXIString /tmp/cacheContent
client.cache.cacheTotalSizeMB  VXIInteger 200
client.cache.cacheLowWaterMarkMB  VXIInteger 180
client.cache.cacheEntryMaxSizeMB  VXIInteger 20
client.cache.cacheEntryExpTimeSec  VXIInteger 3600
client.cache.unlockEntries      VXIInteger 1
### Logging
client.log.filename             VXIString /tmp/log.txt
client.log.fileMimeType         VXIString text/plain;charset=ISO-8859-1
#client.log.fileMimeType       VXIString text/plain;charset=UTF-8
client.log.maxLogSizeMB         VXIInteger 50
client.log.contentDir           VXIString /tmp/logContent
client.log.contentTotalSizeMB   VXIInteger 50
# The default is to log to standard out as well as to a file (set to 1)
# set to 0 to disable logging to standard out
client.log.logToStdout          VXIInteger 0
# The default is to keep the log file open between writes for faster
# logging (set to 1), set to 0 to close between writes to allow
# manually rotating the log file by merely moving it aside while the
# platform continues running
client.log.keepLogFileOpen      VXIInteger 1
# The default is to report the error text for each error, as contained
# in the XML error mapping files defined below
client.log.reportErrorText      VXIInteger 1
### Internet fetch, extension rules defined separately below
#client.inet.proxyServer        VXIString 172.17.100.1
#client.inet.proxyPort          VXIInteger 8080
client.inet.userAgent           VXIString OpenVXI/3.1
client.inet.acceptCookies       VXIInteger 1
client.inet.postContinueTimeout  VXIInteger 1

```

```

#### Proxy rules
#client.inet.proxyRule.0      VXIString .speechworks.com/specialPath |
proxyServer1:123
#client.inet.proxyRule.1      VXIString i6net.org |
#client.inet.proxyRule.2      VXIString .com | proxyServer2:456
#client.inet.proxyRule.3      VXIString | proxyServer3:789
#### JavaScript
client.jsi.runtimeSizeBytes    VXIInteger 16384000
client.jsi.contextSizeBytes    VXIInteger 131072
client.jsi.maxBranches         VXIInteger 100000
#client.jsi.globalScriptFile    VXIString http://greenland/misc/test.js
#### QA Prompt cache
client.prompt.enableCache      VXIInteger 1
#### Session connection variables
client.session.connection.local.uri  VXIString telephone.dnis
client.session.connection.remote.uri  VXIString telephone.ani
client.session.connection.protocol.name  VXIString OpenVXI_VXML
client.session.connection.protocol.version  VXIString 1.0.0
client.session.connection.aai          VXIString telephone.param
client.session.connection.originator    VXIString connection.remote
# redirect array: each element of the array has 4 properties:
# uri, pi, si and reason
client.session.connection.redirect.0.uri  VXIString http://www.company1.com/redirect
client.session.connection.redirect.0.pi  VXIString presentation information 0
client.session.connection.redirect.0.si  VXIString screening information 0
client.session.connection.redirect.0.reason  VXIString unknown
client.session.connection.redirect.1.uri  VXIString http://www.company2.com/redirect
client.session.connection.redirect.1.pi  VXIString presentation information 1
client.session.connection.redirect.1.si  VXIString screening information 1
client.session.connection.redirect.1.reason  VXIString unknown
#### TRD utilities
# The stack size in bytes to use when creating new threads. If set to
# zero or left undefined it means 'use the default (OS-specific) size',
# which will usually be the same stack size as the parent process.
#client.trd.threadStackSize          VXIInteger 0
#### SSFT's Recognizer configuration
#client.rec.initURI                  VXIString $(SWISRSDK)/config/Baseline.xml
#### VoiceXML Interpreter
client.vxi.beepURI                    VXIString uri:beepbis.gsm
# Uncomment the following to override the interpreter defaults
#client.vxi.defaultsURI              VXIString file:///etc/opensvxi/defaults.xml
#client.vxi.maxLoopIterations        VXIInteger 50
#client.vxi.maxDocuments             VXIInteger 20
#client.vxi.maxEvents                VXIInteger 30
#client.vxi.optimization             VXIInteger 1
#####
# Base diagnostic tag offset for each interface #
#####
client.cache.diagLogBase              VXIInteger 2000
client.inet.diagLogBase               VXIInteger 3000
client.jsi.diagLogBase                VXIInteger 4000
client.prompt.diagLogBase            VXIInteger 5000

```

```

client.rec.diagLogBase          VXIInteger 6000
client.tel.diagLogBase          VXIInteger 7000
client.vxi.diagLogBase          VXIInteger 8000
client.object.diagLogBase       VXIInteger 9000
client.client.diagLogBase       VXIInteger 10000
#####
# Diagnostic tags: 0 to disable, 1 to enable #
#####
# API/general log traces for each component
client.log.diagTag.0            VXIInteger 0
client.log.diagTag.2000         VXIInteger 0
client.log.diagTag.2001         VXIInteger 0
client.log.diagTag.2002         VXIInteger 0
client.log.diagTag.3000         VXIInteger 0
client.log.diagTag.3001         VXIInteger 0
client.log.diagTag.3002         VXIInteger 0
client.log.diagTag.3003         VXIInteger 0
client.log.diagTag.3004         VXIInteger 0
client.log.diagTag.3005         VXIInteger 0
client.log.diagTag.3006         VXIInteger 0
client.log.diagTag.3007         VXIInteger 0
client.log.diagTag.3008         VXIInteger 0
client.log.diagTag.3010         VXIInteger 0
client.log.diagTag.4000         VXIInteger 0
client.log.diagTag.4001         VXIInteger 0
client.log.diagTag.4002         VXIInteger 0
client.log.diagTag.5000         VXIInteger 0
client.log.diagTag.5001         VXIInteger 0
client.log.diagTag.6000         VXIInteger 0
client.log.diagTag.6001         VXIInteger 0
client.log.diagTag.6002         VXIInteger 0
client.log.diagTag.7000         VXIInteger 0
client.log.diagTag.9000         VXIInteger 0
# VXi logging, the first is for application diagnostics/errors, the
# second is the output from the <log> element
client.log.diagTag.8000         VXIInteger 0
client.log.diagTag.8001         VXIInteger 0
client.log.diagTag.8002         VXIInteger 0
client.log.diagTag.8003         VXIInteger 0
# SBclient API, component, and generic logging respectively
client.log.diagTag.10000        VXIInteger 0
client.log.diagTag.10001        VXIInteger 0
client.log.diagTag.10002        VXIInteger 0
# testClient logging
client.log.diagTag.60001        VXIInteger 0
# Pass-through of OpenSpeech Recognizer diagnostic messages as enabled
# in the configured OSR diagnostic tag map file, always leave this
# enabled
client.log.diagTag.79999        VXIInteger 0
#####
# Error mapping files #
#####

```

```

client.log.errorMapFile.1      VXIString /usr/lib/opensvxi/VXIclientErrors.xml
client.log.errorMapFile.2      VXIString /usr/lib/opensvxi/VXIErrors.xml
client.log.errorMapFile.3      VXIString /usr/lib/opensvxi/VXIobjectErrors.xml
client.log.errorMapFile.4      VXIString /usr/lib/opensvxi/VBpromptErrors.xml
client.log.errorMapFile.5      VXIString /usr/lib/opensvxi/VBrecErrors.xml
client.log.errorMapFile.6      VXIString /usr/lib/opensvxi/VBtelErrors.xml
client.log.errorMapFile.7      VXIString /usr/lib/opensvxi/SBcacheErrors.xml
client.log.errorMapFile.8      VXIString /usr/lib/opensvxi/SBinetErrors.xml
client.log.errorMapFile.9      VXIString /usr/lib/opensvxi/SBjsiErrors.xml
client.log.errorMapFile.10     VXIString /usr/lib/opensvxi/SBblogErrors.xml
#####
# TTS server configuration #
#####
client.prompt.resource.0.uri    VXIString http://localhost/tts/verbio/tts.php
#client.prompt.resource.0.uriVideo  VXIString http://localhost/tts/video/ttv.php
client.prompt.resource.0.method  VXIString POST
client.prompt.resource.0.cacheDir  VXIString /tmp/cacheContent
client.prompt.resource.0.format    VXIString alaw
client.prompt.resource.0.formatVideo  VXIString 3gp
client.prompt.resource.0.maxage     VXIInteger -1
client.prompt.resource.0.checkBreak  VXIInteger 1
client.prompt.resource.0.cutPrompt   VXIInteger 0
#####
# ASR server configuration #
#####
client.rec.resource.0.cacheDir    VXIString /tmp/cacheContent
client.rec.resource.0.format      VXIString bnf
#####
# File extension to MIME type mapping rules #
#####
client.inet.extensionRule.xml     VXIString text/xml
client.inet.extensionRule.txt     VXIString text/plain
client.inet.extensionRule.ulaw    VXIString audio/basic
client.inet.extensionRule.wav     VXIString audio/x-wav
client.inet.extensionRule.alaw    VXIString audio/x-alaw-basic
client.inet.extensionRule.srgs    VXIString application/srgs+xml
client.inet.extensionRule.grxml   VXIString application/srgs+xml
client.inet.extensionRule.grm     VXIString application/isolated
client.inet.extensionRule.jsgf    VXIString application/x-jsgf
client.inet.extensionRule.bnf     VXIString application/verbio-abnf
client.inet.extensionRule.gram    VXIString application/lumenvox-abnf
client.inet.extensionRule.ssmml   VXIString application/synthesis+ssml
client.inet.extensionRule.vxml    VXIString application/voicexml+xml
#####
# Objects initialisations #
#####
#client.object.initialisation.demo  VXIString UP

```

## **ANEXO 2**

Aplicación VoiceXML

## bienvenida.vxml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
<form>
  <block name="time">
    <var name="horas"/>
    <var name="minutos"/>
    <var name="saludo"/>
    <script>
      var d = new Date();
      horas=d.getHours();
      minutos=d.getMinutes();
      if(horas > 11)
      {
        if(horas > 18)
          saludo="Buenas noches"
        else
          saludo="Buenas tardes";
      }
      else
      {
        saludo="Buenas dias";
      }
    </script>
    <prompt> <value expr="saludo"/>
      Son las <value expr="horas"/>horas,
      <value expr="minutos"/> minutos
    </prompt>
  </block>
  <block>
    <prompt bargein="false">
      Bienvenidos al Sistema de Facturacion VoiceXML. a continuacion se presentan las
      siguientes opciones.
    </prompt>
    <goto next="menu.vxml"/>
  </block>
</form>
</vxml>
```

## menu.vxml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
<form id="menu">
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="12s"/>
  <field name="item">
    <grammar mode="voice" type="application/verbio-text">
      saldos
      servicios
      ayuda
      salir
    </grammar>
    <prompt bargein="true">
      ,Menu principal.
      , saldos.
      , servicios.
      , ayuda.
      , salir.
      escoja su opcion por favor
    </prompt>
    <filled>
      <if cond="item =='saldos'">
        <goto next="planilla.vxml"/>
      </if>
      <if cond="item =='servicios'">
        <goto next="servicios.vxml"/>
      </if>
      <if cond="item =='ayuda'">
        <goto next="ayuda.vxml"/>
      </if>
      <if cond="item =='salir'">
        <goto next="salir.vxml"/>
      </if>
      <clear namelist="item"/>
    </filled>
  </field>
</form>
</vxml>
```

## planilla.vxml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
<form id="menu">
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="10s"/>
<field name="item">
  <grammar mode="voice" type="application/verbio-text">
    saldo total
    detalle
  </grammar>
  <prompt bargein="true">
    Bienvenidos a este es un sub menu que tiene dos opciones:
    uno, saldo total.
    2, detalle.
    Escoja su opcion por favor.
  </prompt>
  <filled>
    <if cond="item =='saldo total'">
      <goto next="planillauno.vxml"/>
    </if>
    <if cond="item =='detalle'">
      <goto next="planillados.vxml"/>
    </if>
    <clear namelist="item"/>
  </filled>
</field>
</form>
</vxml>
```

## planillauno.vxml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <form>
    <property name="inputmodes" value="voice"/>
    <property name="timeout" value="15s"/>
    <var name="caller" expr="session.connection.remote.uri"/>
    <block><prompt>Dime tu numero telefonico,por favor,recuerda que el numero telefonico
tiene 9 digitos.</prompt></block>
    <field name="pin">
      <grammar src="builtin:grammar/digits"/>
      <filled>
        <prompt>Tu dijiste
          <value expr="pin"/>
        </prompt>
      </filled>
      <nomatch>Lo sentimos, no entendi eso. Intentelo de nuevo.
      <reprompt/>
      </nomatch>
    </field>
  </block>
  <submit next="planuno.php" method="post" namelist="caller pin"/>
</form>
</vxml>
```

## planuno.php

```
<?php echo '<?xml version="1.0" encoding="iso-8859-1"?>';?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <property name="prompturl2" value="http://titus.i6net.org/tts/verbio/tts.php"/>
  <property name="promptposition" value="00" />
  <form>
    <block>
      <prompt bargein="false" xml:lang="text">
        NUMEROS CONSULTADOS;
      </prompt>
      <?php
        $caller = $_REQUEST["caller"];
        $telefono= $_REQUEST["pin"];
        $conexion = mysql_connect("localhost","root","");
        mysql_select_db("telefonica",$conexion);
        $sql ="select * from t_factura where txt_id_numero_tel='".$telefono.'" and int_estado=1";
        $temp = mysql_query($sql);
        $dat=mysql_fetch_assoc($temp);
        $fil= mysql_num_rows($temp);
        ?>
      </prompt>
      <?php
        if($fil == 0 )
        {
          echo "Lo siento, ese numero no esta registrado en nuestra base de datos";
        }
        else
        {
          echo "el saldo total a pagar por su planilla telefonica es:";
          echo $dat["dl_total"]."dolares";
        }?>
      </prompt>
      <goto next="planillatres.vxml"/>
    </block>
  </form>
</vxml>
```

## planillados.vxml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <form>
    <property name="inputmodes" value="voice"/>
    <property name="timeout" value="15s"/>
    <var name="caller" expr="session.connection.remote.uri"/>
    <field name="pin">
      <grammar src="builtin:grammar/digits"/>
      <prompt>Dime tu numero telefonico, por favor, recuerda que el numero telefonico tiene 9
digitos.
</prompt>
      <filled>
        <prompt>Tu dijiste
          <value expr="pin"/>
        </prompt>
      </filled>
      <nomatch count="1">Lo sentimos, no entendi eso. Intentelo de nuevo.
<reprompt/>
      </nomatch>
    </field>
  </block>
  <submit next="plandos.php" method="post" namelist="caller pin"/>
</block>
</form>
</vxml>
```

## plandos.php

```
<?php echo '<?xml version="1.0" encoding="iso-8859-1"?>';?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <property name="prompturl2" value="http://titus.i6net.org/tts/verbio/tts.php"/>
  <property name="promptposition" value="00" />
  <form>
    <block>
      <prompt bargein="false" xml:lang="text">
        DETALLE;
      </prompt>
      <?php
        $caller = $_REQUEST["caller"];
        $telefono= $_REQUEST["pin"];
        $conexion = mysql_connect("localhost","root","");
        mysql_select_db("telefonica",$conexion);
        $sql ="select * from t_factura where txt_id_numero_tel='".$telefono."'";
        $temp = mysql_query($sql);
        $dat=mysql_fetch_assoc($temp);
        $num_fact=$dat["txt_id_factura"];
        $sql1 ="select t_servicios.txt_descripcion as descripcion, t_items.cantidad as cantidad,
        t_items.total_item as total from t_servicios inner join t_items on ($num_fact=txt_id_fact and
        t_items.txt_id_ser=t_servicios.txt_id_servicio)";
        $temp1= mysql_query($sql1);
        $fila=mysql_num_rows($temp1);
      ?>
    </prompt>
    <prompt bargein="false">
      <?php
        if($fila==0)
        {
          echo"Lo siento, ese numero no esta registrado en nuestra base de datos";
        }
        else
        {
          echo"Detalle de su planilla telefonica";
          while ($row = mysql_fetch_array($temp1))
          {
            echo "\n".$row["descripcion"]."\n";
            echo "\n".$row["cantidad"]." minutos";
          }
        }
      </?php
    </prompt>
  </form>
</vxml>
```

```
    echo "\n".$row["total"]."dolares\n";
}
echo "el valor del Subtotal es:";
echo "\n".$dat["dl_subtotal"]."dolares \n";
echo "el valor del iva es:";
echo "\n".$dat["dl_iva"]."dolares \n";
echo "el saldo total a pagar por su planilla telefonica es:";
echo "\n".$dat["dl_total"]."dolares";
}
?>
</prompt>
    <goto next="planillatres.vxml"/>
</block>
</form>
</vxml>
```

## servicios.vxml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <form>
    <block>
      <submit next="servicios.php" method="post" namelist="caller"/>
    </block>
  </form>
</vxml>
```

## servicios.php

```
<?php echo '<?xml version="1.0" encoding="iso-8859-1"?>';?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
  <property name="prompturl2" value="http://titus.i6net.org/tts/verbio/tts.php"/>
  <property name="promptposition" value="00" />
  <form>
    <block>
      <prompt bargein="false" xml:lang="text">
        Servicios;
      </prompt>
      <?php
        $caller=$_REQUEST["caller"];
        $conexion = mysql_connect("localhost","root","");
        mysql_select_db("telefonica",$conexion);
        $sql ="select * from t_servicios where estado=1";
        $temp1= mysql_query($sql);
        ?>
      </prompt>
      <prompt bargein="false">
        La empresa VooiceXML ofrece los siguiente servicios:
      </prompt>
      <prompt bargein="false">
      <?php
        while ($row = mysql_fetch_array($temp1))
        {
          echo "\n".$row["txt_descripcion"]."\n";
          echo "el valor del servicio es";
          echo "\n".$row["dl_valor"]."\n";
          echo "dolares";
        }
        ?>
      </prompt>
      <prompt bargein="false">
        Gracias por visitarnos, proxicamente estaremos lanzando nuevos servicios. Regresamos al
      </prompt>
      <goto next="menu.vxml"/>
    </block>
  </form>
</vxml>
```

## ayuda.vxml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
<form id="ayuda">
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="30s"/>
  <block>
    <prompt bargein="false">
      Hola, esta es la ayuda del Sistemas de Planillas Telefonicas Voice,xml. Donde existen
      cuatro opciones:
      Opcion uno del Menu Principal) Saldos,
      Esta es una opcion que se divide en un submenu:
      Opcion Uno del Submenu) Saldo Total,
      Tu puedes decir el numero telefonico que deseas consultar y a continuacion se dará a
      conocer el costo a pagar,nunca olvides que el numero telefonico tiene 9 digitos.
      Opcion dos del Submenu) Detalle,
      Usted puede conocer el detalle del consumo telefonico diciendo el numero telefonico.
      Opcion dos del Menu Principal)Servicios,
      En esta opcion se da a conocer los servicios que posee la empresa, ANIMATE Y ENTERATE
      DE TODAS LAS NOVEDADES DE NUESTRA EMPRESA.
      Opcion tres del Menu Principal) Ayuda,
      Con esta opcion se puede tener una guia del sistema.
      Opcion cuatro del Menu Principal) Salir,
      Con esta opcion automaticamente tu puedes salir del sistema.
    </prompt>
    <goto next="opcion_ayuda.vxml"/>
  </block>
</form>
</vxml>
```

## salir.vxml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="es-ES">
<form id="salir">
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="30s"/>
  <block>
    <prompt bargein="false">
      Gracias por visitar nuestra aplicacion.Esperamos un pronto retorno.Adios
    </prompt>
  </block>
  <disconnect/>
</form>
</vxml>
```

# **ANEXO 3**

Aplicación Web

# index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>VXML</title>
<style type="text/css">
<!--
.Estilo1 {font-style: italic}
.Estilo2 {
    font-style: italic;
    font-size: 18px;
}
-->
</style>
</head>
<body>
<table align="center" width="600">
  <br><br><br><br>
<tr height="50"><td height="42" colspan="2" bgcolor="#33CC33"><div align="center"><span
class="Estilo1">
</span><h4 align="center"><span class="Estilo2"><strong><font color="#003399"
face="Geneva, Arial, Helvetica, sans-serif">Bienvenidos al Sistema de
Facturaci&oacute;n</font></strong></span></h4>
<h4 align="center" class="Estilo2"><em><strong><font color="#003399" face="Geneva, Arial,
Helvetica, sans-serif">VXML</font></strong></em></h4>
</div></td></tr>
<tr>
  <td></td>
  <td><table bgcolor="#33CC33">
    <form method="post" action="">
    <tr><td width="120" align="center" ><div align="left"><strong><font color="#003399"
face="Geneva, Arial, Helvetica, sans-serif">Login</font></strong>:</div></td>
      <td width="281" align="center"><input type="text" name="login"></td>
    </tr>
    <tr>
      <td width="120" align="left" ><strong><font color="#003399" face="Geneva, Arial,
Helvetica, sans-serif">Password:</font></strong></td> <td width="281" align="center">
<input type="password" name="pass">
```



## validar.php

```
<?php
include("conec.php");
$link=Conectarse();
$user=$_POST["login"];
$pass=$_POST["pass"];
if((isset($user))&& (isset($pass))&&(($user!="")&& ($pass!="")))
{
    $sql="select * from t_administrador where login='".$user.'" and
password='".$pass.'";
    $result=mysql_query($sql,$link);
    $row=mysql_fetch_array($result);
    if($row)
    {
        header("location:menu.php");
    }
    else
    {
        header("location:index.html");
    }
    session_destroy();
}
else
{
    header("location:index.html");
}
?>
```

## menu.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>VXML</title>
<style type="text/css">
<!--
.Estilo1 {font-size: 18px}
.Estilo4 {font-weight: bold; font-size: 18px;}
.Estilo6 {font-size: 12px}
-->
</style>
</head>
<body>
<table align="center" width="600">
<br><br><br><br><br><br>
<tr height="50" ><td height="42" colspan="2" bgcolor="#33CC33"><div align="center"
class="Estilo1"><span class="Estilo4"><font color="#003399" face="Geneva, Arial, Helvetica,
sans-serif">Men&uacute; Principal </font></span><br>
<br>
</div></td></tr>
<tr><td align="center"></td>
<td>
<table>
<tr><td align="right"><div align="left"><strong><a href="ingresar.php">Ingreso de
Datos</a></strong></div></td></tr>
<tr><td align="right"><div align="left"><strong><a href="administra.php">Ingreso de
Usuarios</a></strong></div></td></tr>
<tr><td align="right"><div align="left"><strong><a href="elegir_factura.php">Mostrar
Factura</a></strong></div></td></tr>
<tr><td align="right"><div align="left"><strong><a href="elegir_cliente.php">Mostrar
Clientes</a></strong></div></td></tr>
<tr><td align="right"><div align="left"><strong><a href="mostrar_servicios.php">Mostrar
Servicios</a></strong></div></td></tr>
<tr height="30" ><td height="22" bgcolor="white"><div align="center"><span class="Estilo1">
&nbsp;
</span></div>
</td>
</tr>
</table>
```

```
</td>
</tr>
<tr height="50" width="600"><td height="42" colspan="2" bgcolor="#33CC33"><div
align="center"><span class="Estilo1">
<span class="Estilo6"><font color="#003399" face="Geneva, Arial, Helvetica, sans-
serif">Copyright (c)2010 ESPOCH:EIS</font></span><br>
</span></div>
</td>
</tr>
</table>
</body>
</html>
```





## client.php

```
<?php
    include("conec.php");
    $link=Conectarse();
    $Sql="insert into
t_cliente(txt_cedula,txt_nombres,txt_apellidos,txt_direccion,txt_ciudad) values
('".$_POST["cedula"]."', '".$_POST["nombre"]."', '".$_POST["apellido"]."',
 '".$_POST["direccion"]."', '".$_POST["ciudad"]."'");
    mysql_query($Sql,$link);
    header("Location: confirmar.php");
?>
```

## cliente.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<style type="text/css">
<!--
.Estilo1 {
    font-family: Geneva, Arial, Helvetica, sans-serif;
    font-weight: bold;
}
.Estilo2 {font-family: Geneva, Arial, Helvetica, sans-serif}
-->
</style>
<body>
<table align="center" width="600">
<br><br><br><br><br>
<tr height="50" ><td height="42" colspan="3" bgcolor="#33CC33"><div align="center"><span
class="Estilo1"><span class="Estilo2"><font color="#003399" face="Geneva, Arial, Helvetica,
sans-serif"> Insertar Cliente</font></span><br>
    <br>
</span></div>
</td>
</tr>
<form method="post" action="client.php">
    <tr><td width="123" rowspan="5"></td>
    <td width="93"><font color="#003399"><span class="Estilo2"><br>
    Cedula</span></font></td>
    <td width="368"><br>
    <input type="text" name="cedula" width="200"></td>
</tr>
<tr>
<td ><font color="#003399"><span class="Estilo2">Nombres:</span></font></td>
<td><input type="text" name="nombre" width="300"></td>
</tr>
<tr>
<td><font color="#003399"><span class="Estilo2">Apellidos:</span></font></td>
<td><input type="text" name="apellido" width="300"></td>
</tr>
<tr>
<td><font color="#003399"><span
class="Estilo2">Direcci&oacute;n:</span></font></td>
    <td><input type="text" name="direccion" width="300"></td>
```

```

        </tr>
        <tr>
        <td><font color="#003399"><span class="Estilo2">Ciudad:</span></font></td>
        <td><input type="text" name="ciudad" width="300"></td>
        </tr>
        <tr align="center">
        <td align="center" colspan="3"><br><input type="submit" name="aceptar"
value="Aceptar"/>
        <input type="reset" name="cancelar" value="Cancelar"/>    </td>
        </tr>
        </form>
<tr height="50" ><td height="42" colspan="4" bgcolor="white"><div align="center"><span
class="Estilo1">
<br><br>
</span></div>
</td>
</tr>
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1">
<br><a href="ingresar.php" class="Estilo2">Regresar</a>
<br>
</span></div>
</td>
</tr>

</table>
</body>
</html>

```

## confirmar.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<style type="text/css">
<!--
.Estilo1 {
    font-family: Geneva, Arial, Helvetica, sans-serif;
    font-weight: bold;
}
-->
</style>
<body>
<table align="center" width="600">
<br><br><br><br><br><br><br><br>
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1">
CONFIRMACION<br><br>
</span></div>
</td>
</tr>
    <tr><td>&nbsp;</td></tr>
    <tr><td>&nbsp;</td></tr>
    <tr align="center">
        <td align="justify"><div align="center"><font color="BLACK" size="5" face="Geneva,
Arial, Helvetica, sans-serif">ij Los datos han sido ingresado correctamente
!!</font></div></td>
    </tr>
    <tr><td>&nbsp;</td></tr>
    <tr><td>&nbsp;</td></tr>
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1"><a href="ingresar.php">Regresar</a>
<br><br>
</span></div>
</td>
</tr>
</table>
</body>
</html>
```



```
    ?>
</select>
    </td>
</tr>
<tr height="15" ><td height="10" colspan="4" bgcolor="white"><div align="center">
</div></td>
</tr>
<tr align="center">
<td align="center" colspan="2"><input type="submit" name="Buscar" value="Aceptar"/>
<input type="reset" name="cancelar" value="Cancelar"/><br><br>
</td>
</tr>
</form>
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1">
<br><a href="ingresar.php" class="Estilo2">Regresar</a>
<br>
</span></div>
</td>
</tr>
</table>
</body>
</html>
```

## factura.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<style type="text/css">
<!--
.Estilo1 {
    font-family: Geneva, Arial, Helvetica, sans-serif;
    font-weight: bold;
}
.Estilo2 {font-size: 18px}
.Estilo3 {font-family: Geneva, Arial, Helvetica, sans-serif; font-weight: bold; font-size: 18px; }
-->
</style>
<body>
<table align="center">
<br><br><br><br><br><br><br><br><br>
<tr height="50" width="400"><td height="42" colspan="4" bgcolor="#33CC33"><div
align="center"><span class="Estilo3">
<font color="#003399" face="Geneva, Arial, Helvetica, sans-serif"> Ingreso del
C.I.</font></span><br>
</span></div>
</td>
</tr>
<form method="post" action="factura_b.php">
<tr align="center">
    <td align="center"><font color="#003399"><br>
    Cedula Id.:</font></td>
    <td align="center"><br>
    <input type="text" name="id_cedula" width="200"></td>
</tr>
<tr height="15" ><td height="10" colspan="4" bgcolor="white"><div align="center"><span
class="Estilo1">
</span></div></td>
</tr>
<tr align="center">
<td align="center" colspan="2"><input type="submit" name="Buscar" value="Aceptar"/>
<input type="reset" name="cancelar" value="Cancelar"/><br><br>
</td>
</tr>
</form>
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
```

```
class="Estilo1">  
<br><a href="ingresar.php" class="Estilo2">Regresar</a>  
<br>  
</span></div>  
</td>  
</tr>  
</table>  
</body>  
</html>
```

## factura\_b.php

```
<?php
include("conec.php");
$link=Conectarse();
include("libreria.php");
$user=$_POST["id_cedula"];
$result2=getServicios($link);
$row=getUser($user,$link);
$date=date("d/m/Y");
if($row)
{
    $nom=$row["txt_nombres"];
    $ape=$row["txt_apellidos"];
    $dir=$row["txt_direccion"];
    $result1=getTelefono($user,$link);
    if($result1){
?>
<html>
<style type="text/css">
<!--
.Estilo1 {
    font-family: Geneva, Arial, Helvetica, sans-serif;
    font-weight: bold;
}
.Estilo2 {font-size: 18px}
-->
</style>
<body>
<table align="center">
<br>
<br><br>
<form action="insertar.php" method="post">
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1"><span class="Estilo2"><font color="#003399" face="Geneva, Arial, Helvetica,
sans-serif"> Insertar Factura</font></span><br>
    <br>
</span></div>
</td>
</tr>
<tr height="2" ><td colspan="4" bgcolor="white">&nbsp;  </td>
</tr>
<tr>
<td width="150"><font color="#003399" >
Cod. Fact</font></td><td width="250">
```

```

<input type="text" name="id_factura" width="50">
 </td>
<td width="150"><font color="#003399" >
C.I:</font> </td><td width="150"><input type="text" name=cedula value="<?php echo
$_POST["id_cedula"];?>" READONLY></td>
</tr>
<input type="hidden" name="val" value="0">
<input type="hidden" name="val1" value="1">
<tr>
<td width="150"><font color="#003399" >
Nombre</font> </td><td width="250"><input type="text" name=nombre value="<?php echo
$nom.' '.$ape;?>" READONLY> </td>
<td width="150"><font color="#003399" >
Direcci&oacute;n </font></td>
<td width="150"><input type="text" name=direccion value="<?php echo $dir;?>"
READONLY></td>
</tr>
<tr>
<td width="150" height="23"><font color="#003399" >
Num.Telf </font></td><td width="250">
<select name="numero">
  <?php
    while($row = mysql_fetch_array($result1))
    {
      printf("<option value='%s'>%s</option>", $row['txt_numero_telefonico'],
$row['txt_numero_telefonico']);
    }
  ?>
</select>
</td>
<td width="150"><font color="#003399" >
IVA </font></td><td width="150"><input type="text" name="iva" value="12%"
READONLY></td>
</tr>
<tr>
<td width="150"><font color="#003399" >
Fecha Emisi&oacute;n </font></td>
<td width="250"><input type="text" name=fecha value="<?php echo $date; ?>" READONLY>
</td>
<td width="150"><font color="#003399" >
Per&iacute;odo Pago</font></td>
<td width="150">
<select name="periodo" >
  <option value="Enero">Enero</option>
  <option value="Febrero">Febrero</option>

```

```

        <option value="Marzo">Marzo</option>
        <option value="Abril">Abril</option>
        <option value="Mayo">Mayo</option>
        <option value="Junio">Junio</option>
        <option value="Julio">Julio</option>
        <option value="Agosto">Agosto</option>
        <option value="Noviembre">Noviembre</option>
        <option value="Diciembre">Diciembre</option>
    </select>
    <select name="anio" >
        <option value="2010">2010</option>
        <option value="2011">2011</option>
        <option value="2012">2012</option>
        <option value="2013">2013</option>
        <option value="2009">2014</option>
    </select></td>
</tr>
<tr height="2" ><td colspan="4" bgcolor="white">&nbsp;&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td colspan="4" align="center"><br>
<input type="submit" name="guardar" value="Guardar"><input type="reset" name="cancelar"
value="Cancelar">
<br><br></td>
</tr>
</form>
<tr height="2" >
    <td colspan="4" bgcolor="white"><div align="center"></div></td>
</tr>
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1">
</span></div></td>
</tr>
</table>
</body>
<?php
}}
else
{
    header("location:error.php");
}
?>
</html>

```

## guardar.php

```
<?php
    include("conec.php");
    $link=Conectarse();
    $cont=$_POST["contador"];
    $i=1;
    $suma=0;
    $iva=0;
    $total=0;
    while($i<=$cont)
    {
        $sql="select * from t_servicios where txt_descripcion='".$_POST['x'].$i.'"and
estado=0";
        $sql1=mysql_query($sql,$link);
        $sql2=mysql_fetch_array($sql1);
        $total_item=$_POST['y'].$i * $sql2["dl_valor"];
        $total_item=round($total_item,2);
        $sql3="insert into t_items(txt_id_ser,txt_id_fact,cantidad,total_item) values ('".$sql2
["txt_id_servicio"]."',".$_POST["fact"]."',".$_POST['y'].$i."',".$_total_item."')";
        mysql_query($sql3,$link);
        $i++;
    }
?>
<html>
<style type="text/css">
<!--
.Estilo1 {
    font-family: Geneva, Arial, Helvetica, sans-serif;
    font-weight: bold;
}
.Estilo8 {font-size: 18px; font-family: Geneva, Arial, Helvetica, sans-serif; font-weight: bold; }
-->
</style>
<body>
<table align="center">
<br>
<br><br>
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1">
    Insertar Factura<br><br>
</span></div>
</td>
</tr>
<tr>
```

```

<td width="150"><font color="#003399" >
Cod. Fact</font></td><td width="250">
<input type="text" name="fact1" value="<?php echo $_POST["fact"];?>" READONLY>
<input type="hidden" name="fact1" value="<?php echo $_POST["fact"];?>">
 </td>
<td width="150"><font color="#003399" >
C.I:</font></td><td width="150"><input type="text" name="cedula1" value="<?php echo
$_POST["cedula"];?>" READONLY></td>
</tr>
<input type="hidden" name="val" value="0">
<input type="hidden" name="val1" value="1">
<tr>
<td width="150"><font color="#003399" >
Nombre </font></td><td width="250"><input type="text" name="nombre1" value="<?php echo
$_POST["nombre"];?>" READONLY> </td>
<td width="150"><font color="#003399" >
Direcci&oacute;n </font></td>
<td width="150"><input type="text" name="direccion1" value="<?php echo
$_POST["direccion"];?>" READONLY></td>
</tr>
<tr>
<td width="150" height="23"><font color="#003399" >
Num.Telf </font></td><td width="250">
<input type="text" name="telefono1" value="<?php echo $_POST["telefono"];?>"
READONLY></td>
<td width="150"><font color="#003399" >
IVA</font></td><td width="150"><input type="text" name="iva" value="12%"
READONLY></td>
</tr>
<tr>
<td width="150"><font color="#003399" >
Fecha Emisi&oacute;n</font></td>
<td width="250"><input type="text" name="fecha1" value="<?php echo $_POST["fecha"];?>"
READONLY> </td>
<td width="150"><font color="#003399" >
Per&iacute;odo Pago</font></td>
<td width="150">
<input type="text" name="periodo1" value="<?php echo $_POST["periodo"];?>" READONLY>
</td>
</tr>
<tr height="2" ><td colspan="4" bgcolor="#33CC33">&nbsp;</td>
</tr>
<tr height="2" ><td colspan="4" bgcolor="white">&nbsp;</td>
</tr>

```

```

</table>
<table align="center">
<tr>
<td bgcolor="#33CC33" width="300"><span class="Estilo8"><font color="#003399"
>DESCRIPCION</font></span></td>
<td bgcolor="#33CC33" width="240"><span class="Estilo8"><font color="#003399"
>CANTIDAD</font></span></td>
<td bgcolor="#33CC33" width="200"><span class="Estilo8"><font color="#003399"
>VALOR</font></span></td>
</tr>
<?php
    $result=mysql_query("select t_servicios.txt_descripcion as descripcion, t_items.cantidad as
cantidad, t_items.total_item as total from t_servicios inner join t_items on
(txt_id_fact='".$$_POST["fact"]."' and t_items.txt_id_ser=t_servicios.txt_id_servicio)", $link);
    while($r = mysql_fetch_array($result))
    {
printf("<tr><td>%s</td><td>%s</td><td width=>%s </td></tr>", $r["descripcion"],
$r["cantidad"], $r["total"]);
        $suma=$suma+$r["total"];
    }
    $iva=round($suma*0.12,2);
    $total=round($iva+$suma,2);
    mysql_query("update t_factura set dl_subtotal= '".$suma."',dl_iva= '".$iva."',dl_total=
'".$total.'" WHERE txt_id_factura='".$$_POST["fact"]."'");
?>
<tr height="2" ><td colspan="4" bgcolor="#33CC33">&nbsp;</td>
</tr>
<tr height="2" align="right"><td colspan="4" bgcolor="white">Subtotal <input type="text"
name="subtotal" value="<?php echo $suma;?>" READONLY></td>
</tr>
<tr height="2" align="right"><td colspan="4" bgcolor="white">Iva (12%) <input type="text"
name="iva" value="<?php echo $iva;?>" READONLY></td>
</tr>
<tr height="2" align="right"><td colspan="4" bgcolor="white">TOTAL <input type="text"
name="total" value="<?php echo $total;?>" READONLY></td>
</tr>
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1"><a href="ingresar.php">Regresar</a>
</span></div></td>
</tr>
</table>
</body>
</html>

```

## insertar.php

```
<?php
    include("conec.php");
    $link=Conectarse();
    include("libreria.php");
    $result2=getServicios($link);
    $i=1;
    $Sql="insert into
    t_factura(txt_id_factura,txt_cedula,txt_id_numero_tel,dt_periodo_fact,dt_fecha_emision,dl_s
    ubtotal,dl_iva,dl_total,int_estado) values ('".$_POST["id_factura"]."',".$_POST["cedula"]."',
    '".$_POST["numero"]."',".$_POST["periodo"].'
    '.$_POST["anio"]."',".$_POST["fecha"]."',".$_POST["val"]."',
    '".$_POST["val"]."',".$_POST["val"]."',".$_POST["val1"].'");'
    mysql_query($Sql,$link);
    $Sql1="select * from t_factura where txt_cedula='".$_POST["cedula"]."' and int_estado=1";
    $d=mysql_query($Sql1,$link);
    $row=mysql_fetch_array($d);
    $Sql2="select * from t_cliente where txt_cedula='".$_POST["cedula"].'";
    $d1=mysql_query($Sql2,$link);
    $row1=mysql_fetch_array($d1);
    ?>
<html>
<style type="text/css">
<!--
.Estilo1 {
    font-family: Geneva, Arial, Helvetica, sans-serif;
    font-weight: bold;
}
-->
</style>
<body>
<table align="center">
<br><br><br>
<form action="guardar.php" method="post">
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1"><span class="Estilo2"><font color="#003399" face="Geneva, Arial, Helvetica,
sans-serif"> Insertar Factura</font></span></div><br>
    <br>
</span></div>
</td>
</tr>
<tr>
```







```

        <tr>
            <td><h4><font size="3" face="Geneva, Arial, Helvetica, sans-serif"
color="#003399"><a href="items.php">Insertar Items </a></font></h4></td>
        </tr>
        <tr>
            <td><h4><font size="3" face="Geneva, Arial, Helvetica, sans-serif"
color="#003399"><a href="factura.php">Insertar Factura</a>
            </font></h4></td>
        </tr>
    </table>
</td>
<td width="759" valign="top" align="left">
    <table width="462" background="imagenes/menu.jpg">
        <form method="post" action="admin.php">
            <tr>
                <td colspan="2"><h2 align="center"><font face="Geneva, Arial, Helvetica, sans-serif"
color="#003399">Ingresar datos Items</font></h2></td>
            </tr>
            <tr>
                <td width="100" ><h4 align="left"><font color="#003399" face="Geneva, Arial,
Helvetica, sans-serif">Id. Servicio:</font></h4></td>
                <td width="404" ><input type="text" name="id_servicio" width="200"></td>
            </tr>
            <tr>
                <td width="100" ><h4 align="left"><font color="#003399" face="Geneva, Arial,
Helvetica, sans-serif">Id. Factura:</font></h4></td>
                <td width="404" ><input type="text" name="id_factura" width="300"></td>
            </tr>
            <tr>
                <td width="100" ><h4 align="left"><font color="#003399" face="Geneva, Arial,
Helvetica, sans-serif">Cantidad:</font></h4></td>
                <td width="404" ><input type="text" name="cantidad" width="300"></td>
            </tr>
            <tr>
                <td width="100" ><h4 align="left"><font color="#003399" face="Geneva, Arial,
Helvetica, sans-serif">Total:</font></h4></td>
                <td width="404" ><input type="text" name="total" width="300"></td>
            </tr>
            <tr align="center">
                <td width="140" align="center" colspan="2"><input type="submit" name="aceptar"
value="Aceptar"/>
                <input type="reset" name="cancelar" value="Cancelar"/>
            </td>
        </tr>
    </form>

```

```
</table>
</td>
</tr>
</table>
</body>
</html>
```

## mostrar\_factura.php

```
<?php
    include("conec.php");
    $link=Conectarse();
    $sql="select * from t_factura where txt_id_factura='".$_POST['factura']."'";
    $sql1=mysql_query($sql,$link);
    $sql2=mysql_fetch_array($sql1);
    $sql3=$sql2["txt_cedula"];
    $sql4="select * from t_cliente where txt_cedula='".$sql3."'";
    $sql5=mysql_query($sql4,$link);
    $sql6=mysql_fetch_array($sql5);
?>
<html>
<style type="text/css">
<!--
.Estilo1 {
    font-family: Geneva, Arial, Helvetica, sans-serif;
    font-weight: bold;
}
.Estilo8 {font-size: 18px; font-family: Geneva, Arial, Helvetica, sans-serif; font-weight: bold; }
-->
</style>
<body>
<table align="center">
<br><br><br>
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1">
Mostrar Datos de la Factura<br><br>
</span></div>
</td>
</tr>
<tr>
<td width="150"><font color="#003399" >
Cod. Fact</font></td><td width="250">
<input type="text" name="fact1" value="<?php echo $_POST["factura"];?>" READONLY>
 </td>
<td width="150"><font color="#003399" >
C.I:</font> </td><td width="150"><input type="text" name=cedula1 value="<?php echo
$sql2["txt_cedula"];?>" READONLY></td>
</tr>
<input type="hidden" name="val" value="0">
<input type="hidden" name="val1" value="1">
<tr>
<td width="150"><font color="#003399" >
```

```

Nombre </font></td><td width="250"><input type="text" name=nombre1 value="<?php echo
$sql6["txt_nombres"].'.'. $sql6["txt_apellidos"];?>" READONLY> </td>
<td width="150"><font color="#003399" >
Direcci&oacute;n </font></td>
<td width="150"><input type="text" name=direccion1 value="<?php echo
$sql6["txt_direccion"];?>" READONLY></td>
</tr>
<tr>
<td width="150" height="23"><font color="#003399" >
Num.Telf </font></td><td width="250">
<input type="text" name="telefono1" value="<?php echo $sql2["txt_id_numero_tel"];?>"
READONLY></td>
<td width="150"><font color="#003399" >
IVA</font> </td><td width="150"><input type="text" name="iva" value="12%"
READONLY></td>
</tr>
<tr>
<td width="150"><font color="#003399" >
Fecha Emisi&oacute;n</font> </td>
<td width="250"><input type="text" name=fecha1 value="<?php echo
$sql2["dt_fecha_emision"];?>" READONLY> </td>
<td width="150"><font color="#003399" >
Per&iacute;odo Pago</font></td>
<td width="150">
<input type="text" name="periodo1" value="<?php echo $sql2["dt_periodo_fact"];?>"
READONLY>
</td>
</tr>
<tr height="2" ><td colspan="4" bgcolor="#33CC33">&nbsp;</td>
</tr>
<tr height="2" ><td colspan="4" bgcolor="white">&nbsp;</td>
</tr>
</table>
<table align="center">
<tr>
<td bgcolor="#33CC33" width="300"><span class="Estilo8"><font color="#003399"
>DESCRIPCION</font></span></td>
<td bgcolor="#33CC33" width="240"><span class="Estilo8"><font color="#003399"
>CANTIDAD</font></span></td>
<td bgcolor="#33CC33" width="200"><span class="Estilo8"><font color="#003399"
>VALOR</font></span></td>
</tr>

```

```

<?php
    $result=mysql_query("select t_servicios.txt_descripcion as descripcion, t_items.cantidad as
cantidad, t_items.total_item as total from t_servicios inner join t_items on
(txt_id_fact='".$$_POST["factura"]." and t_items.txt_id_ser=t_servicios.txt_id_servicio)", $link);
    while($r = mysql_fetch_array($result))
    {
printf("<tr><td>%s</td><td>%s</td><td width=>%s </td></tr>", $r["descripcion"],
$r["cantidad"], $r["total"]);
    }
?>
<tr height="2" ><td colspan="4" bgcolor="#33CC33">&nbsp;  </td>
</tr>
<tr height="2" align="right"><td colspan="4" bgcolor="white">Subtotal <input type="text"
name="subtotal" value="<?php echo $sql2["dl_subtotal"];?>" READONLY></td>
</tr>
<tr height="2" align="right"><td colspan="4" bgcolor="white">Iva (12%) <input type="text"
name="iva" value="<?php echo $sql2["dl_iva"];?>" READONLY></td>
</tr>
<tr height="2" align="right"><td colspan="4" bgcolor="white">TOTAL <input type="text"
name="total" value="<?php echo $sql2["dl_total"];?>" READONLY></td>
</tr>
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1"><a href="menu.php">Regresar</a>
</span></div></td>
</tr>
</table>
</body>
</html>

```



## servicios.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<style type="text/css">
<!--
.Estilo1 {
    font-family: Geneva, Arial, Helvetica, sans-serif;
    font-weight: bold;
}
.Estilo3 {font-family: Geneva, Arial, Helvetica, sans-serif}
.Estilo4 {
    font-size: 18px;
    font-weight: bold;
}
-->
</style>
<body>
<table align="center" width="600">
<br><br><br><br><br><br>
<tr height="50" ><td height="42" colspan="3" bgcolor="#33CC33"><div align="center"><span
class="Estilo3"><span class="Estilo4"><font color="#003399" face="Geneva, Arial, Helvetica,
sans-serif">Insertar Servicios</font></span></span><span class="Estilo1"><br>
    <br>
</span></div></td></tr>
<form method="post" action="service.php">
<tr><td width="114">&nbsp;</td>
<td width="113"><br>
    Id. Servicio </td><td width="357"><br>
    <input type="text" name="id" width="200"></td></tr>
<tr><td>&nbsp;</td><td>Descripcion </td><td><input type="text" name="descripcion"
width="300"></td></tr>
<tr><td>&nbsp;</td><td>Valor </td><td><input type="text" name="valor"
width="300"></td></tr>
<tr><td>&nbsp;</td><td>Estado </td><td><input type="text" name="estado"
width="300"></td></tr>
<tr align="center"><td align="center" colspan="3"><br>
    <input type="submit" name="aceptar" value="Aceptar"/><input type="reset"
name="cancelar" value="Cancelar"/></td></tr>
</form>
<tr height="30" ><td height="22" colspan="4" bgcolor="white"><div align="center"><span
class="Estilo1">
&nbsp;</span></div></td></tr>
```

```
<tr height="50" ><td height="42" colspan="4" bgcolor="#33CC33"><div align="center"><span
class="Estilo1">
<br><a href="ingresar.php" class="Estilo2">Regresar</a>
<br>
</span></div></td></tr>
</table>
</body>
</html>
```

## conec.php

```
<?php
function Conectarse()
{
    if (!($link=mysql_connect("localhost","root","")))
    {
        exit();
    }
    if (!mysql_select_db("telefonica",$link))
    {
        exit();
    }
    return $link;}
?>
```

# **ANEXO 3**

Script de la Base de Datos

```

-- MySQL dump 10.11
--
-- Host: localhost  Database: telefonica
-----
-- Server version      5.0.77
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO,NO_KEY_OPTIONS,NO_TABLE_OPTIONS,NO_FIELD_
OPTIONS' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
--
-- Table structure for table `t_administrador`
--
DROP TABLE IF EXISTS `t_administrador`;
SET @saved_cs_client  = @@character_set_client;
SET character_set_client = utf8;
CREATE TABLE `t_administrador` (
  `txt_cedula` varchar(10) NOT NULL default "",
  `txt_nombres` varchar(50) default NULL,
  `txt_apellidos` varchar(50) default NULL,
  `login` varchar(20) default NULL,
  `password` varchar(20) default NULL,
  PRIMARY KEY (`txt_cedula`)
);
SET character_set_client = @saved_cs_client;
--
-- Dumping data for table `t_administrador`
--
LOCK TABLES `t_administrador` WRITE;
/*!40000 ALTER TABLE `t_administrador` DISABLE KEYS */;
INSERT INTO `t_administrador` VALUES
('','','',''),('045555566','jjjklll','lñlññ','kkk','ñññ'),('0603108770','janet','concha','jane','jane'),('18
03753704','Edgar','Morales','edgar','edgar');
/*!40000 ALTER TABLE `t_administrador` ENABLE KEYS */;
UNLOCK TABLES;
--
-- Table structure for table `t_cliente`
--
DROP TABLE IF EXISTS `t_cliente`;
SET @saved_cs_client  = @@character_set_client;
SET character_set_client = utf8;

```

```

CREATE TABLE `t_cliente` (
  `txt_cedula` varchar(10) NOT NULL default "",
  `txt_nombres` varchar(50) default NULL,
  `txt_apellidos` varchar(50) default NULL,
  `txt_direccion` varchar(50) default NULL,
  `txt_ciudad` varchar(50) default NULL,
  PRIMARY KEY (`txt_cedula`)
);
SET character_set_client = @saved_cs_client;
--
-- Dumping data for table `t_cliente`
--
LOCK TABLES `t_cliente` WRITE;
/*!40000 ALTER TABLE `t_cliente` DISABLE KEYS */;
INSERT INTO `t_cliente` VALUES ('0603108770','Janeth','Concha','La
Dolorosa','Riobamba'),('1803753704','edgar','morales','mochapata','quero');
/*!40000 ALTER TABLE `t_cliente` ENABLE KEYS */;
UNLOCK TABLES;
--
-- Table structure for table `t_factura`
--
DROP TABLE IF EXISTS `t_factura`;
SET @saved_cs_client = @@character_set_client;
SET character_set_client = utf8;
CREATE TABLE `t_factura` (
  `txt_id_factura` varchar(5) NOT NULL default "",
  `txt_cedula` varchar(10) NOT NULL,
  `txt_id_numero_tel` varchar(9) default NULL,
  `dt_periodo_fact` varchar(20) default NULL,
  `dt_fecha_emision` varchar(20) default NULL,
  `dl_subtotal` varchar(20) default NULL,
  `dl_iva` varchar(20) default NULL,
  `dl_total` varchar(20) default NULL,
  `int_estado` tinyint(4) default '0',
  PRIMARY KEY (`txt_id_factura`),
  KEY `t_factura_FK1` (`txt_id_numero_tel`),
  KEY `t_factura_FK2` (`txt_cedula`),
  CONSTRAINT `t_factura_FK1` FOREIGN KEY (`txt_id_numero_tel`) REFERENCES
`t_linea_telefonica` (`txt_numero_telefonico`),
  CONSTRAINT `t_factura_FK2` FOREIGN KEY (`txt_cedula`) REFERENCES `t_cliente`
(`txt_cedula`)
);
SET character_set_client = @saved_cs_client;
--

```

```

-- Dumping data for table `t_factura`
--
LOCK TABLES `t_factura` WRITE;
/*!40000 ALTER TABLE `t_factura` DISABLE KEYS */;
INSERT INTO `t_factura` VALUES ('12','0603108770','032940026','Enero
2010','05/05/2010','92.15','11.06','103.21',1),('214','1803753704','087033576','Abril
2010','04/05/2010','316.7','38','354.7',1);
/*!40000 ALTER TABLE `t_factura` ENABLE KEYS */;
UNLOCK TABLES;
--
-- Table structure for table `t_items`
--
DROP TABLE IF EXISTS `t_items`;
SET @saved_cs_client = @@character_set_client;
SET character_set_client = utf8;
CREATE TABLE `t_items` (
  `txt_id_ser` varchar(5) NOT NULL default "",
  `txt_id_fact` varchar(5) NOT NULL default "",
  `cantidad` int(11) NOT NULL,
  `total_item` double default NULL,
  PRIMARY KEY (`txt_id_fact`,`txt_id_ser`),
  KEY `t_items_FK2` (`txt_id_ser`),
  CONSTRAINT `t_items_FK1` FOREIGN KEY (`txt_id_fact`) REFERENCES `t_factura`
(`txt_id_factura`),
  CONSTRAINT `t_items_FK2` FOREIGN KEY (`txt_id_ser`) REFERENCES `t_servicios`
(`txt_id_servicio`)
);
SET character_set_client = @saved_cs_client;
--
-- Dumping data for table `t_items`
--
LOCK TABLES `t_items` WRITE;
/*!40000 ALTER TABLE `t_items` DISABLE KEYS */;
INSERT INTO `t_items` VALUES
('1','12',5,31),('2','12',5,0.1),('3','12',5,0.5),('4','12',5,0.5),('5','12',5,0.05),('9','12',5,60),('1','214',
12,74.4),('2','214',10,1.15),('3','214',10,1),('4','214',10,1),('5','214',10,0.1),('9','214',20,240);
/*!40000 ALTER TABLE `t_items` ENABLE KEYS */;
UNLOCK TABLES;
--
-- Table structure for table `t_linea_telefonica`
--
DROP TABLE IF EXISTS `t_linea_telefonica`;
SET @saved_cs_client = @@character_set_client;
SET character_set_client = utf8;
CREATE TABLE `t_linea_telefonica` (

```

```

`txt_numero_telefonico` varchar(9) NOT NULL default "",
`txt_cedula_propietario` varchar(10) default NULL,
PRIMARY KEY (`txt_numero_telefonico`),
KEY `t_linea_telefonica_FK2` (`txt_cedula_propietario`),
CONSTRAINT `t_linea_telefonica_FK2` FOREIGN KEY (`txt_cedula_propietario`) REFERENCES
`t_cliente` (`txt_cedula`)
);
SET character_set_client = @saved_cs_client;
--
-- Dumping data for table `t_linea_telefonica`
--
LOCK TABLES `t_linea_telefonica` WRITE;
/*!40000 ALTER TABLE `t_linea_telefonica` DISABLE KEYS */;
INSERT INTO `t_linea_telefonica` VALUES
('032940026','0603108770'),('087033576','1803753704');
/*!40000 ALTER TABLE `t_linea_telefonica` ENABLE KEYS */;
UNLOCK TABLES;
--
-- Table structure for table `t_servicios`
--
DROP TABLE IF EXISTS `t_servicios`;
SET @saved_cs_client = @@character_set_client;
SET character_set_client = utf8;
CREATE TABLE `t_servicios` (
  `txt_id_servicio` varchar(5) NOT NULL default "",
  `txt_descripcion` varchar(50) default NULL,
  `dl_valor` double default NULL,
  `estado` tinyint(4) default '0',
  PRIMARY KEY (`txt_id_servicio`)
);
SET character_set_client = @saved_cs_client;
--
-- Dumping data for table `t_servicios`
--
LOCK TABLES `t_servicios` WRITE;
/*!40000 ALTER TABLE `t_servicios` DISABLE KEYS */;
INSERT INTO `t_servicios` VALUES ('002','Intenet inalambrico',12,1),('1','Pension
Basica',6.2,0),('2','Llamada Reg Automatica',0.02,0),('22','Lineas Telefonicas',45,1),('3','Llamada
a Movistar',0.1,0),('4','Llamada a Porta',0.1,0),('5','Consumo
Local',0.01,0),('8','Internet',10,1),('9','minutos gratis',12,0);
/*!40000 ALTER TABLE `t_servicios` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

```

```
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;  
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;  
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;  
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;  
-- Dump completed on 2010-05-10 20:11:39
```