

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**



**Facultad de Ingeniería en Electricidad y Computación**

**“Diseño e Implementación de un Sistema Telefónico  
para la Lectura de Repositorios Digitales de  
Información con la Finalidad de Prestar el Servicio a  
Personas con Discapacidades Visuales”**

**INFORME DE MATERIA DE GRADUACIÓN**

Previa a la obtención del Título de:

**INGENIERO EN TELEMÁTICA**

Presentada por:

**MARIA FERNANDA MOLINA MIRANDA**

**LUIS ENRIQUE SÁNCHEZ LOOR**

**GUAYAQUIL – ECUADOR**

**AÑO**

**2011**

## AGRADECIMIENTO

Agradezco a Dios, a mis padres, y a mis profesores que siempre confiaron en mí haciendo posible que el sueño de ser profesional se haga realidad.

***Ma. Fernanda Molina Miranda***

Agradezco a Dios, a mis padres y hermanos que han estado conmigo para desarrollar mis objetivos como persona, como estudiante y como profesional.

***Luis Enrique Sánchez Loor***

## DEDICATORIA

A mis padres, tíos, abuelitas, hermana, enamorado y amigos por su amor y apoyo incondicional.

***Ma. Fernanda Molina Miranda***

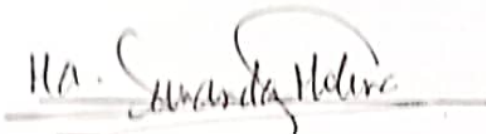
A mi familia, mis padres, hermanos, enamorada y colegas por su amor e interés en mis éxitos.

***Luis Enrique Sánchez Loor***

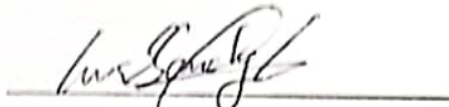
## DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Trabajo de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral".

(Reglamento de Graduación de la ESPOL)



Ma. Fernanda Molina Miranda



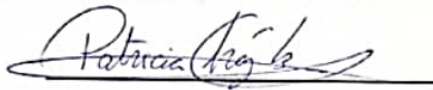
Luis Enrique Sánchez Loo

## TRIBUNAL DE SUSTENTACIÓN



Ing. Gabriel Astudillo

**PROFESOR DE LA MATERIA DE GRADUACIÓN**



Ing. Patricia Chávez

**PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD**

## RESUMEN

El objetivo principal del presente proyecto fue diseñar e implementar un IVR basado en Asterisk, el cual junto con otras tecnologías fue capaz de leer repositorios digitales como fueron las páginas web de periódicos, diccionarios, la wikipedia, blogs y libros.

El sistema inicia recogiendo los datos de páginas web estandarizadas para generar archivos con un formato que permite ser leído por otra herramienta que es capaz de generar archivos de voz, los cuales fueron reproducidos y así el usuario escuchará la información, además de la recepción por medio de comandos de voz para la elección entre los cinco repositorios digitales disponibles.

La central telefónica Asterisk permite la comunicación entre las diferentes aplicaciones como son Festival TTS, VXI y Sphinx mediante scripts en lenguaje PHP y Perl.

Esta herramienta puede ser adquirida por un bajo costo, es de fácil instalación y mantenimiento, por lo cual podría ser implementada sin problema por alguna institución del gobierno y así las personas obtendrían un nuevo medio de

información.

# INDICE DE CONTENIDO

INTRODUCCIÓN .....	XIII
CAPITULO I .....	1
1.1 ANTECEDENTES .....	2
1.2 OBJETIVOS.....	3
1.2.1 OBJETIVO GENERAL.....	3
1.2.2 OBJETIVOS ESPECÍFICOS.....	4
1.3 JUSTIFICACIÓN .....	5
1.4 METODOLOGÍA.....	6
CAPITULO II .....	7
2.1 VOIP .....	8
2.2 ASTERISK .....	9
2.3.1 PROTOCOLOS DE SEÑALIZACIÓN.....	10
2.3 ARQUITECTURA DE ASTERISK.....	11
2.3.1 PLAN DE MARCADO.....	11
2.3.2 IVR.....	11
2.3.3 AGI .....	12
CAPITULO III.....	13
3.1 HARDWARE .....	14
3.1.1 SERVIDOR .....	14
3.1.2 FXS Y FXO .....	15
3.2 SOFTWARE.....	18
3.2.1 SERVIDOR WEB - APACHE.....	19
3.2.2 VOICEXML .....	19
3.2.3 NAVEGADOR VXI .....	21
3.2.4 FESTIVAL TTS.....	26
3.2.5 SPHINX .....	27
3.2.6 EXPRESIONES REGULARES.....	27
CAPITULO IV .....	29



4.1	INSTALACIÓN LIBRERÍAS REQUERIDAS .....	30
4.2	INSTALACIÓN DE ASTERISK .....	31
4.2.1	INSTALACIÓN DE PAQUETES .....	31
4.2.2	CONFIGURACIÓN DE ASTERISK .....	34
4.3	INSTALACIÓN DEL NAVEGADOR VX .....	37
4.3.1	INSTALACIÓN DEL PAQUETE .....	38
4.3.2	CONFIGURACIÓN DEL NAVEGADOR VXI .....	38
4.4	INSTALACIÓN DE FESTIVAL TTS .....	42
4.4.1	FESTIVAL Y ASTERISK .....	43
4.4.2	FESTIVAL EN ESPAÑOL .....	44
4.5	INSTALACIÓN DE SPHINX .....	46
4.5.1	INSTALACIÓN DE PAQUETES .....	47
4.5.2	CONFIGURACIÓN DE COMPONENTES .....	49
4.5.3	VERIFICACIÓN DE RECONOCIMIENTO .....	51
CAPITULO V .....		52
5.1	INICIO DE SERVICIOS.....	53
5.2	FUNCIONAMIENTO PARCIAL.....	54
5.2.1	PERIÓDICO .....	54
5.2.2	DICCIONARIO .....	58
5.2.3	WIKIPEDIA .....	61
5.2.4	BLOG.....	65
5.2.5	LIBROS.....	67
CONCLUSIONES Y RECOMENDACIONES .....		<b>ERROR! BOOKMARK NOT DEFINED.</b>
GLOSARIO .....		3
REFERENCIAS BIBLIOGRÁFICAS.....		55
ANEXOS .....		6

## INDICE DE FIGURAS

FIG. 1.1 ESQUEMA DEL SISTEMA.....	6
FIG. 3.1 FXS/FXO SIN PBX .....	16
FIG. 3.2 FXS/FXO CON PBX .....	17
FIG. 3.3 ARQUITECTURA DEL VOICEXML .....	20
FIG. 5.1 EJECUCIÓN DEL SCRIPT WELCOME.AGI.....	54
FIG. 5.2 PORTADA DEL DIARIO EL UNIVERSO.....	55
FIG. 5.3 EJECUCIÓN DEL ARCHIVO ELUNIVERSO.AGI .....	55
FIG. 5.4 NOTICIA COMPLETA NÚMERO 2.....	56
FIG. 5.5 EJECUCIÓN DE ARCHIVO FULL3UNIVERSO.PHP.....	57
FIG. 5.6 FINALIZA SCRIPT ELUNIVERSO.AGI.....	57
FIG. 5.7 INGRESO DE PALABRA A BUSCAR .....	59
FIG. 5.8 EJECUCIÓN DE SCRIPT DICCIONARIO.AGI .....	59
FIG. 5.9 PÁGINA WEB CON LOS SIGNIFICADOS DE LA PALABRA EJEMPLAR.....	60
FIG. 5.10 FINALIZA EL DICCIONARIO.....	60
FIG. 5.12 CONSULTA LA PALABRA “GORRA” .....	63
FIG. 5.13 INGRESO DE PALABRA “CELULAR” .....	63
FIG. 5.14 “CELULAR” PALABRA AMBIGUA .....	64
FIG. 5.15 PÁGINA WEB CON SIGNIFICADO COMPLETO. ....	65
FIG. 5.15 FINALIZA WIKIPEDIA.....	65
FIG. 5.16 PÁGINA RSS DEL BLOG.....	66
FIG. 5.17 EJECUCIÓN DEL SCRIPT BLOGSALUD.AGI .....	66
FIG. 5.18 CONTENIDO DE SCRIPT FUULBLOGSALUD.PHP .....	67
FIG. 5.19 SALIDA DEL DICCIONARIO.....	67

FIG. 5.20 EJECUCIÓN DE SCRIPT DECODELETRA.AGI .....	68
FIG. 5.21 RESULTADO DE BÚSQUEDA DE AUTOR CON LETRA "R" .....	69
FIG. 5.22 EJECUCIÓN DE SCRIPT FULL3LIBRO.PHP .....	69
FIG. 5.23 CONTENIDO QUE EJECUTA FULL3LIBRO.PHP .....	70
FIG. 5.24 CONTENIDO QUE EJECUTA FULL3PAGINA.PHP .....	71
FIG. 5.25 FINALIZA EL LIBRO .....	72

# INDICE DE TABLAS

TABLA 3.1 CARACTERÍSTICAS MÍNIMAS DEL SERVIDOR.....	14
TABLA 3.2 GUÍA PARA REQUERIMIENTOS DEL SISTEMA .....	15
TABLA 3.3 APLICACIONES DEL SISTEMA.....	18
TABLA 3.4 ELEMENTOS DEL VXML.....	25
TABLA 3.5 EXPRESIONES REGULARES.....	28

# INTRODUCCIÓN

En la actualidad existe un sinnúmero de programas, aplicaciones y dispositivos que las personas usan para permanecer comunicados, pero todos estos conllevan un costo de adquisición y de uso, la mayoría se aprecian mediante la visión, sin embargo existe un porcentaje alto de personas que no pueden gozar de estos avances por varias razones como por ejemplo, el costo económico o que sufren de discapacidad visual.

Por ello, se ha sacado provecho a varias herramientas para implementar un sistema que será capaz de leer de forma clara y comprensible el contenido de cinco repositorios digitales; con la finalidad de mantener informada e integrada a las personas con discapacidad visual leve, media o severa.

La estructura del presente proyecto comprende cinco capítulos que han sido desarrollados de la siguiente manera:

En el capítulo I se dan a conocer los antecedentes y los objetivos que se desea cumplir, en el capítulo II se dan a conocer las características y el funcionamiento de la central telefónica Asterisk, el posterior capítulo se describen los componentes y herramientas que requiere el sistema, el capítulo IV contiene el proceso de instalación e implementación de cada uno de los elementos y

finalmente en el capítulo V se da a conocer el funcionamiento y archivos que genera el sistema.

La finalidad de este proyecto es aprovechar las diferentes tecnologías que se encuentran disponibles para el beneficio de la comunidad, ya que este sistema puede ser de gran utilidad para muchas personas pero sobre todo las personas con discapacidad visual que se encuentran limitadas a solo escuchar su entorno y no tienen acceso a internet.

# **CAPITULO 1**

## **ANTECEDENTES Y PLANTEAMIENTO**

## 1.1 ANTECEDENTES

En las últimas décadas los avances tecnológicos han sido desarrollados velozmente, nuevas tecnologías que han solucionado enormes problemas como también facilitado la vida de los seres vivos. Sobre todo el ser humano ha sido la especie que se ha vuelto dependiente de estos nuevos sistemas creados por el hombre, porque en la mayoría de los casos son de gran utilidad, pero aún así las personas con mayores necesidades como lo son para las personas que sufren alguna discapacidad están insatisfechas con estos avances.

Teclados con sistema braille, computadores que cuentan al usuario no vidente lo que está en pantalla y escriben lo que dicta su voz, o computadores adaptados para que una persona tetrapléjica pueda navegar sólo con dirigir la mirada hacia el monitor, son algunas de la joyas tecnológicas creadas por empresas y universidades para que personas con distintos tipos de discapacidad sigan en el carro de la informática.

Por ello empresas como Freedom Scientific<sup>1</sup> se han dedicado al desarrollo de productos tecnológicos para aquellas personas ciegas o con poca visión a nivel mundial permitiéndoles participar en este gran avance tecnológico.

Según el Primer Estudio Nacional de la Discapacidad en Chile en el año



2004<sup>2</sup>, una de cada ocho personas con discapacidad ha cursado Enseñanza Media completa y sólo una de cada veinte ha logrado acceder a la educación superior. Con el presente proyecto se pretende que las personas con cegueras accedan a información que sea de gran utilidad, disminuyendo un poco el analfabetismo e inculcando a las personas que manteniéndose informados es la mejor forma de educarse.

La ESPOL está siendo parte de esta integración apoyando algunos proyectos como “Mis manos son mis ojos”. Esta idea se enfoca en la creación de negocios propios por parte de las personas discapacitadas, quienes primero aprenden el manejo de las computadoras y luego aplican sus conocimientos en la creación de pequeños negocios que produzcan mayores ingresos.

## **1.2 OBJETIVOS**

### **1.2.1 OBJETIVO GENERAL**

Implementar un sistema telefónico que permita a las personas con discapacidad visual escuchar la información almacenada en repositorios digitales estándar de manera ágil, cómoda y a bajo costo.

## 1.2.2 OBJETIVOS ESPECÍFICOS

- \* Implementar un asistente de llamadas (IVR).
- \* Demostrar la importancia de VoiceXML para el desarrollo de diálogos automatizados.
- \* Desarrollar los scripts PHP necesarios para la conversión a VoiceXML.
- \* Utilizar la Interfaz de Puerta de Enlace (AGI) para la ejecución de scripts.
- \* Integrar software libre para la implementación del sistema, tales como: Open VXI, Festival TTS, Asterisk.
- \* Habilitar el acceso a información web vía telefónica a los siguientes sitios: blog de salud, diccionario, libro, periódico y wikipedia.

### 1.3 JUSTIFICACIÓN

Como muchos dicen "...El discapacitado es una persona común y corriente que necesita las herramientas tecnológicas necesarias para poder desenvolverse con normalidad...", lema al cual se le hace atributo, ya que se desea aportar con un proyecto que beneficiará a muchas personas discapacitadas, logrando así que ellos puedan aprender y aprovechar todos los beneficios que la tecnología puede ofrecer.

Como cualquier persona, un sistema complejo sería incómodo de utilizar. En la solución que se propone se dispone de un asistente de llamadas con las opciones concretas de los distintos repositorios de información que se tiene acceso. A diferencia de otros métodos de acceso ya existentes, para el proyecto no se necesitan grandes costos de capacitación, dado que la utilización de un teléfono es de conocimiento común.

Existen recursos para el aprendizaje del sistema Braille, no obstante de acuerdo a un artículo publicado por Espacio Logopédico.com<sup>3</sup> sólo un diez por ciento de las personas consideradas ciegas o deficientes visuales lo utilizan debido a la falta de destreza en el sentido del tacto o la falta de textos más atractivos para las diferentes edades. Además del escaso respaldo para obtener imprentas que produzcan libros o revistas en Braille, a causa de su alto costo. El software libre que se integra al sistema beneficiará

el ámbito de gastos, ya que los altos costos serán aislados únicamente para el hardware del servidor.

#### 1.4 METODOLOGÍA

Los pasos necesarios a seguir para cumplir con el principal objetivo es instalar Asterisk sobre un servidor con sistema operativo CentOS Linux sobre el cual tomará desarrollo el IVR. Se desarrollarán scripts en lenguaje PHP que reciban la dirección de diferentes páginas web como son: periódico, blog, wikipedia, diccionario, entre otros; luego se traducirán en un lenguaje común para poder ser leídas a través del AGI. De esta forma, el usuario obtendrá la información deseada.

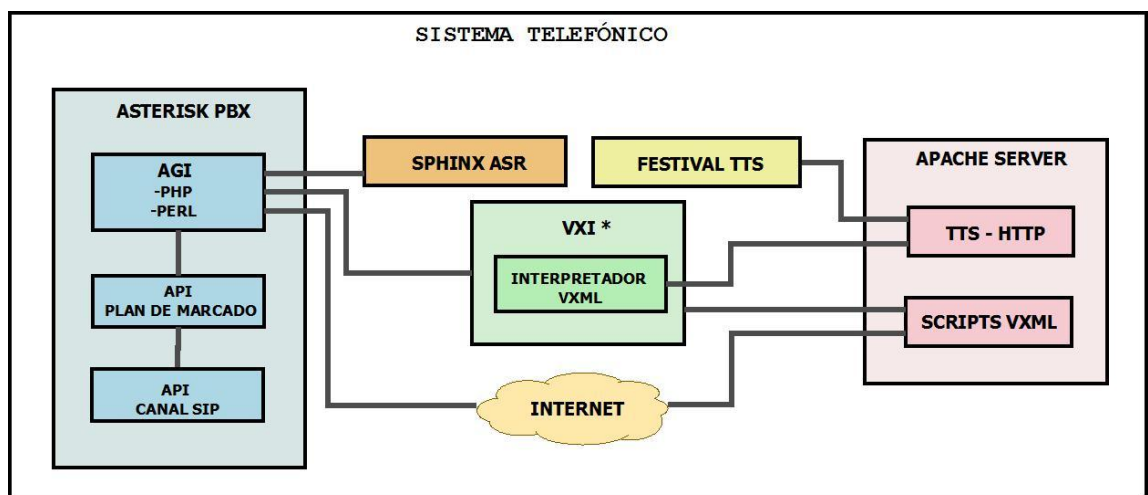


Fig. 1.1 Esquema del sistema

## **CAPITULO 2**

ASTERISK

Como es de conocimiento general, el internet se ha vuelto parte de nuestras vidas porque permite la integración de un sinnúmero de aplicaciones para redes de datos, video y voz. Por ello se considera como un sistema convergente, es decir, se obtiene un medio unificado de intercambio y almacenamiento de información electrónica.

## **2.1 VOIP**

Más conocido como Voz sobre IP, se define así al grupo de normas, dispositivos, recursos que hacen posible que a través de Internet viaje la señal de voz empleando el Protocolo de Internet. La señal de voz se envía en paquetes de datos, es decir, en formato digital y como la señal de voz es de origen analógica se deben usar una serie de protocolos para su adaptación. Estos protocolos permitirán una transmisión de buena calidad, libre de errores, retrasos y sobre todo económica. Por este último beneficio es que se ha vuelto tan solicitado a nivel empresarial.

## 2.2 ASTERISK

Asterisk es un software libre que proporciona funcionalidades de una central telefónica IP (IPBX) conectada directamente a la red pública de teléfono por medio de líneas troncales. Mark Spencer de Digium, desarrolló este sistema que se ha caracterizado por su flexibilidad y además con otros programadores han contribuido en añadir funcionalidades como buzones de voz, conferencias, IVR (Respuesta Automática Interactiva), distribución automática de llamadas, directorios, grabación de llamadas y muchas otras más de manera gratuita y consumiendo pocos recursos de hardware.

Asterisk es un sistema multiplataforma compatible con la mayoría de tecnologías de voz sobre IP (VoIP) como los protocolos de señalización para redes convergentes:

- SIP
- H323
- MGCP
- IAX

Y también los protocolos de telefonía tradicional analógica y digital:

- TDM
- ISDN
- BRI

- PRI

Para utilizar teléfonos analógicos Asterisk requiere usar tarjetas electrónicas FXS (Foreign eXchange Station, Estación de Intercambio Externo) o FXO (Foreign eXchange Office, Oficina de Intercambio Externo) que también son fabricados por Digium y otros proveedores.

### **2.3.1 PROTOCOLOS DE SEÑALIZACIÓN**

Para el establecimiento de las conexiones en un canal de comunicación se deben efectuar una serie de confirmaciones y estándares, conocidos como protocolos de señalización.

- **SIP – Protocolo de Inicio de Sesión**

Protocolo desarrollado por la IETF (Internet Engineering Task Force, Fuerza de Trabajo de Ingeniería de Internet) encargado de iniciar, modificar y eliminar las sesiones o conexiones entre dos o más usuarios. SIP no proporciona flujo de medios ni controla la entrega de flujos. Es un protocolo orientado de extremo a extremo, por ello toda la lógica es almacenada en los puntos finales.



## **2.3 ARQUITECTURA DE ASTERISK**

### **2.3.1 PLAN DE MARCADO**

El plan de marcado o dialplan, como se lo conoce en inglés, es el corazón de Asterisk donde reside la lógica de la central telefónica (PBX). Dado que es único y se configura en el archivo `/etc/asterisk/extensions.conf`, debe seguir una lista de instrucciones que Asterisk emplea como respuesta a peticiones externas.

### **2.3.2 IVR**

La Respuesta de Voz Interactiva de Asterisk, es un conjunto de aplicaciones, contextos y elementos de un plan de marcado. Es un sistema automatizado para la interacción de llamadas entrantes, donde existen grabaciones de voz y reconocimiento de respuestas simples como SI, NO y otros. De acuerdo a su desarrollo, un IVR puede ser orientado a la consulta o ingreso de información, permitiendo acceso a servicios u otras operaciones.

Su principal uso es como asistente automatizado de llamadas, se lo utiliza para varios sistemas telefónicos de empresas, para clientes y para empleados. El alcance que se le pueda dar, requiere de estar correctamente asesorado para evitar incómodos usos por parte de clientes y poder obtener

un mejor perfil.

### **2.3.3 AGI**

La Interfaz de Puerta de Enlace, que agrega funcionalidades a Asterisk mediante diferentes tipos de lenguajes de programación, como Perl, PHP, C, Pascal. Se comunica con Asterisk mediante comandos que controlan las peticiones y respuestas del script. Existen cuatro tipos de AGI:

- AGI: Controla el plan de marcado, llamado desde el archivo `extensions.conf`.
- Async AGI: Permite compilar scripts de forma asíncrona.
- EAGI: Permite controlar y acceder al canal, además de interactuar con el plan de marcado.
- FastAGI: Puede realizar acceso remoto.

## **CAPITULO 3**

COMPONENTES DEL SISTEMA

### 3.1 HARDWARE

Una de las ventajas del sistema es que requiere de tan solo un componente físico, el servidor, el cual debe tener las características necesarias para receptor, procesar y enviar los datos del registro telefónico.

#### 3.1.1 SERVIDOR

Las características mínimas requeridas para el buen funcionamiento del servidor están detalladas en la tabla 3.1.

**Tabla 3.1 Características Mínimas del Servidor**

<b>CPU</b>	Pentium III 2Ghz
<b>MEMORIA RAM</b>	1.5 GB
<b>TARJETA DE RED</b>	10/100Mbps
<b>DISCO DURO</b>	20 GB

De acuerdo al libro Asterisk, The Future of Telephony<sup>4</sup>, estas características deberán ser más robustas y con mayor velocidad dependiendo de la finalidad que el usuario le vaya a dar al sistema. Como se muestra en la tabla 3.2.

**Tabla 3.2 Guía para requerimientos del sistema**

<b>Propósito</b>	<b>Número de Canales</b>	<b>Mínimo Recomendado</b>
<b>Sistema para ocio</b>	No más de 5 canales	400MHz x86, 256MB RAM
<b>Sistema SOHO</b>	De 5 a 10	1 GHz x86, 523MB RAM
<b>Sistema para pequeña empresa</b>	Hasta 25	3 GHz x86, 1GB RAM
<b>Sistema para mediana y grande</b>	Más de 25	Procesador dual, posiblemente múltiples servidores en una arquitectura distribuida.

### **3.1.2 FXS Y FXO**

FXS y FXO<sup>5</sup>, también denominados POTS (Servicio Telefónico Básico y Antiguo), son los nombres de los puertos usados por las líneas telefónicas analógicas. A continuación una breve descripción de cada uno.

### 3.1.2.1 FXS

Es la interfaz de abonado externo, el puerto que envía la línea analógica al abonado. En otras palabras, es el “enchufe de la pared” que envía tono de marcado, corriente para la batería y tensión de llamada.

### 3.1.2.2 FXO

Es la interfaz de la central externa, el puerto que recibe la línea analógica. Es un enchufe del teléfono o de la central telefónica analógica. Envía una indicación de colgado/descolgado (cierre de bucle).

Como el puerto FXO está adjunto a un dispositivo, tal como un teléfono, el dispositivo a menudo se denomina “dispositivo FXO”. Son siempre pares, es decir, similar a un conector macho/hembra.

Sin una centralita, el teléfono se conecta directamente al puerto FXS que brinda la empresa telefónica un ejemplo se muestra en la figura 3.1.

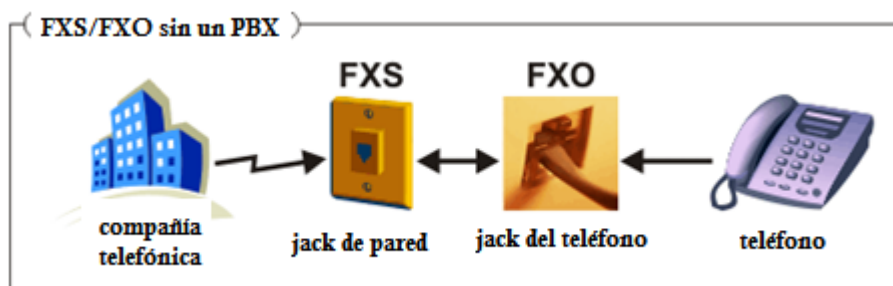
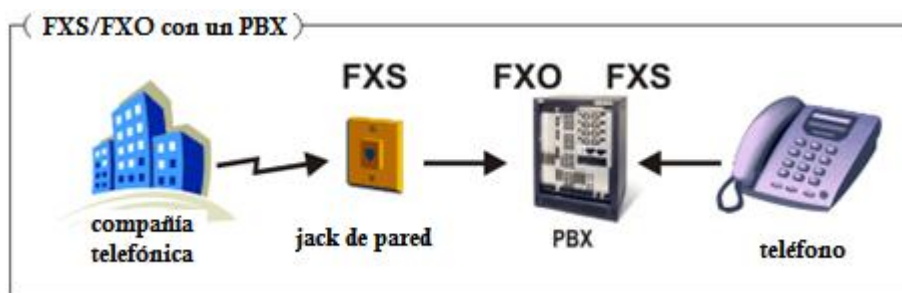


Fig. 3.1 FXS/FXO sin PBX

Si tiene centralita, como se muestra en la figura 3.2 se debe conectar las líneas que suministra la empresa telefónica a la centralita y luego los teléfonos a la centralita. Por lo tanto, la centralita debe tener puertos FXO (para conectarse a los puertos FXS que suministra la empresa telefónica) y puertos FXS (para conectar los dispositivos de teléfono o fax).



**Fig. 3.2 FXS/FXO con PBX**

## 3.2 SOFTWARE

La finalidad del presente proyecto es conocer y aplicar los beneficios que ofrece el software libre, además de disminuir los costos de implementación.

Las aplicaciones que se detallan en la tabla 3.3 fueron elegidas de acuerdo al nivel de compatibilidad con Asterisk, funcionalidades y con licencia gratuita.

**Tabla 3.3 Aplicaciones del sistema**

Plataforma	Linux
Distribución	Centos 5
Servidor Web	Apache
Software IP PBX	Asterisk versión 1.8.4.2
Aplicación Text-to-Speech	Festival TTS
Estándar Voice XML	OpenVXI
Reconocedor de voz	Sphinx



### **3.2.1 SERVIDOR WEB - APACHE**

Apache es el servidor web HTTP de código abierto multiplataforma más usado para enviar páginas web estáticas y dinámicas en aplicaciones web, que son desarrolladas sobre todo en PHP, el cual viene a ser el lenguaje que utilizamos en la implementación de nuestro sistema para desarrollar los scripts que recogen y administran la información desde las páginas web.

### **3.2.2 VOICEXML (VXML - Voice Extensible Markup Language version 2.0, Lenguaje de Marcado Extensible de Voz versión 2.0)**

#### **3.2.2.1 CONCEPTO**

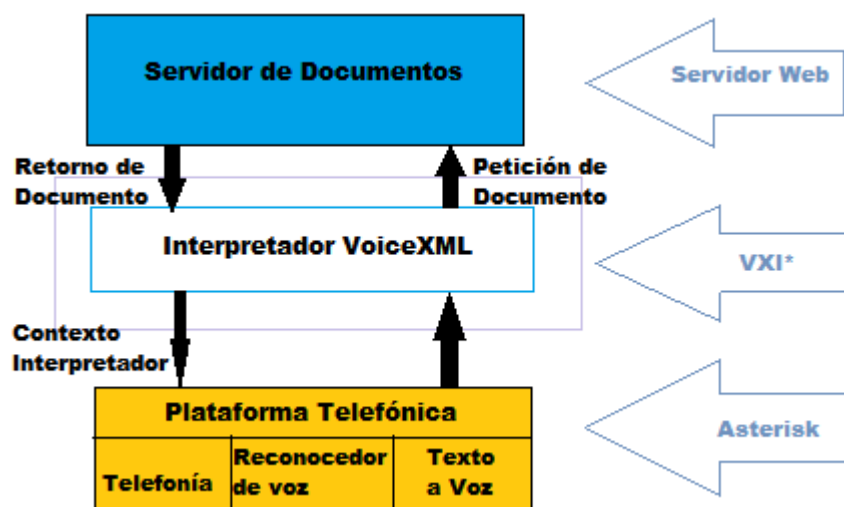
De forma general, VXML es un formato estándar XML de la W3C diseñado para interactuar entre el humano y la computadora creando diálogos de voz mediante un navegador de voz o por teléfono.

Entre sus principales características están: sintetizador de voz, audio digitalizado, reconocimiento de voz, registro de llamadas entrantes y entrada DTMF (marcación por tonos).

Las grandes ventajas del desarrollo de aplicaciones con VXML son: una implementación flexible, el re-uso de infraestructura web existente, plataforma independiente y la existencia de productos disponibles.

### 3.2.2.2 ARQUITECTURA DEL VOICEXML

El modelo de aplicación distribuido del navegador VXI por Asterisk asumido por VoiceXML es un sistema basado en tres niveles con la siguiente arquitectura:



**Fig. 3.3 Arquitectura del VoiceXML**

Cuando una llamada es recibida, la plataforma de implementación (plataforma telefónica, en nuestro caso es Asterisk) envía un evento al interpretador VoiceXML, el cual a su vez se ve en el contexto de la URL del documento inicial a buscar. Una petición es enviada al servidor de documentos para el documento inicial. El servidor de documentos a la vez envía el documento al interpretador VoiceXML que interpreta el documento y lo ejecuta usando los servicios de la plataforma telefónica.

La interpretación de acuerdo a la aplicación puede resultar de varias formas: enviando un mensaje al usuario, esperando una entrada por parte del usuario, grabando una voz, entre otros.

### **3.2.3 Navegador VXi**

Entre los productos que ofrece la compañía europea i6net soluciones y tecnologías está el navegador VXi\* VoiceXML para Asterisk<sup>6</sup>, este software corre sobre VoIP para crear nuestro propio IVR (Respuesta de Voz Interactiva) o plataforma IVVR (Respuesta de Voz y Video Interactiva).

OpenVXi es un kit de herramientas interpretador de VoiceXML portátil de código abierto, es un componente del navegador VXi que provee APIs para servicios de plataforma como reconocimiento de voz, síntesis de habla y servicios de telefonía.

VXi intérprete, elemento del navegador VXi, trabaja directamente con el software PBX Asterisk con el apoyo de Digium y está escrito en lenguaje C sobre el sistema operativo Linux. La licencia gratuita ofrecida para el usuario final solo permite una sesión a la vez, es decir, solo una llamada puede ser atendida; para fines comerciales es necesario tener la licencia comercial que ya tiene un costo.

El navegador VXI\* VoiceXML de Asterisk permite desarrollar aplicaciones de voz y video a través de IP, PSTN y redes 3G-324M, es compatible con el estándar VoiceXML, también tiene integrado reconocimiento automático de voz (ASR) y texto a voz (TTS).

Comúnmente es instalado en servidores con sistema operativo Linux de 32 bits o 64 bits con distribuciones Debian, CentOS, Suse Ubuntu y RedHat. Las soluciones de VXI\* pueden ejecutar en la configuración del servidor principal y en cualquier hardware compatible con Linux, lo que permite desarrollar un sistema de acuerdo a las necesidades del usuario.

### **3.2.3.1 CARACTERÍSTICAS**

- Solución basada en software, empaquetado 100%.
- Disponible para varias distribuciones de Linux OS en 32 o 64 bits aplicaciones interoperables simultánea de voz y de video.
- Soporta 3G-324M que incluye H.263, H.263, H.264.
- VoiceXML 2.0 compatible con etiquetas extendidas añadidas recientemente.
- Transcodificación de video y adaptación de tasa en tiempo real.
- Complementos disponibles para extender su plataforma VXI\*.

### 3.2.3.2 PRINCIPALES CONCEPTOS

Es necesario dar a conocer algunos conceptos que son importantes para el manejo del lenguaje VoiceXML<sup>7</sup>.

- **SESIONES**

Una sesión inicia cuando el usuario empieza a interactuar con el documento VoiceXML.

- **APLICACIONES**

Una aplicación es una colección de documentos VoiceXML, todos los documentos en una aplicación comparten la misma aplicación raíz del documento.

- **FORMULARIOS**

Un formulario en un documento VoiceXML presenta información y recoge la entrada del usuario.

- **MENÚS**

Un menú le da al usuario una lista de opciones para elegir y transiciones a un diálogo diferente o un documento de base en la

elección del usuario.

- **ENLACES**

Un enlace especifica una transición que es común a todos los diálogos en el ámbito de la relación. Cuando una entrada del usuario coincide con el enlace de la gramática, transfiere el control al destino del vínculo.

- **GRAMÁTICAS**

Una gramática especifica una lista de vocabulario permisible (palabras y frases) para que el usuario seleccione con el fin de para interactuar con la aplicación VoiceXML. Cada diálogo tiene uno o más vocabularios y/o gramáticas asociadas con ella.

- **EVENTOS**

Un evento es lanzado por la plataforma VoiceXML para una serie de razones, como por ejemplo cuando un usuario no responde a una entrada, no responde correctamente, o las solicitudes de ayuda, etc. El intérprete VoiceXML también lanza los acontecimientos en el caso de que haya errores semánticos en el documento de VoiceXML.

### 3.2.3.3 ELEMENTOS DEL VXML

A continuación en la tabla 3.4 se detallan los elementos Vxml<sup>8</sup> que se han implementado en la configuración del sistema.

**Tabla 3.4 Elementos del VXML**

<b>Elemento</b>	<b>Descripción</b>
<assign>	Asigna valor a una variable.
<block>	Ejecuta el código que contiene.
<if>	Código condicional.
<else>	Código de condición del elemento <if>.
<exit>	Finaliza la aplicación.
<field>	Recoge la entrada del usuario.
<filled>	Define código luego que el usuario ha ingresado un dato.
<form>	Define diálogos para recoger la entrada del usuario.
<goto>	Salta hacia otro diálogo.
<link>	Define URL para enlazar cuando se lo seleccione.
<prompt>	Define salida para ser reproducida.

<subdialog>	Invoca a otro diálogo y luego retorna.
<return>	Retorna desde un subdiálogo.
<submit>	Invoca otro diálogo abandonando el diálogo previo.
<value>	Invoca a una variable.
<var>	Declara una variable.
<vxml>	Elemento raíz del documento.

### 3.2.4 FESTIVAL TTS

Es nuestro sistema sintetizador de voz, el cual es multilingüístico y fue desarrollado por CSTR (Centro de Investigación de Tecnologías del Lenguaje)<sup>9</sup>, ofrece un completo sistema de conversión de texto a voz mediante API's que se integran a Asterisk.

Festival está escrito en lenguaje C++ y está implementado como un intérprete de comandos para un control general sobre software libre con licencia MIT-X11 para usar el código fuente sin restricciones.

Hemos implementamos Festival en idioma español y agregamos módulos que permiten obtener una voz con un léxico más claro.



### **3.2.5 SPHINX**

Sphinx es un reconocedor de voz que se integra con Asterisk, cuenta con un gran vocabulario en inglés y trabaja bajo la licencia estilo de *Berkeley*<sup>10</sup>.

También es una colección de herramientas de código abierto y recursos que permite a los desarrolladores construir sistemas de reconocimiento de voz. Trabaja en dos módulos, cliente y servidor, los cuales fueron modificados para ser implementados en nuestro proyecto.

### **3.2.6 EXPRESIONES REGULARES**

Se les conoce como patrones, son textos especiales conformados por una serie de signos y caracteres especiales que permiten realizar búsquedas y reemplazar porciones de texto específicos dentro de una cadena de caracteres más grande.

Las expresiones regulares fueron la herramienta clave para obtener la información desde las páginas web, recogiendo los datos dentro de documentos HTML y XML. Muchos lenguajes de programación admiten el uso de expresiones regulares para la manipulación de textos. Los caracteres más usados se encuentran detallados en la tabla 3.5.

**Tabla 3.5 Expresiones Regulares**

<b>Nombre</b>	<b>Signo</b>	<b>Descripción</b>
Barra vertical		Para separar las alternativas.
Signo más	+	Indica que el carácter anterior a él debe aparecer al menos una vez.
Signo de interrogación	?	Indica que el carácter anterior a él debe aparecer como mucho una sola vez.
Signo de multiplicar	*	Indica que el carácter anterior a él debe aparecer ninguna, una o más veces.
Paréntesis	( )	Para definir el ámbito y precedencia de los demás operadores.
Punto	.	Representa cualquier carácter excepto los que representen el salto de línea.
Barra inversa	\	Se combina con otros caracteres y dependiendo de estos adquiere diferentes significados.
Corchetes	[ ]	Sirven para agrupar caracteres en grupos o clases.
Acento circunflejo	^	Tiene doble funcionalidad. Individual representa el inicio de cadena y dentro de corchetes representa a cualquier carácter que no se encuentre dentro del grupo indicado.

## **CAPITULO 4**

### **INSTALACIÓN E IMPLEMENTACIÓN**

## 4.1 INSTALACIÓN DE LIBRERÍAS REQUERIDAS

Se utiliza la terminal de comandos para la instalación de librerías y compiladores, requeridos para la instalación del sistema a implementar en un servidor que ha iniciado sesión como usuario root, el cual tiene todos los permisos necesarios.

Requeridos por Asterisk:

```
# yum install -y bison bison-devel
# yum install -y ncurses ncurses-devel
# yum install -y libtermcap libtermcap-devel
# yum install -y zlib zlib-devel
# yum install -y openssl openssl-devel
# yum install -y gnutls-devel
# yum install -y gcc gcc-c++
# yum install -y newt newt-devel
# yum install -y libtool libtool-devel
```

Requeridos por Festival:

```
# yum install -y perl
# yum install rpm-build
```

## 4.2 INSTALACIÓN DE ASTERISK

Se ubica en la carpeta `/usr/src/` y luego se procede a la descarga de los paquetes de instalación para Asterisk. Estos son: Asterisk, dahdi complete, libpri. Se puede usar el comando `#wget` de la siguiente manera.

```
# cd /usr/src/  
  
# wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-  
1.8.3.2.tar.gz  
  
# wget http://downloads.asterisk.org/pub/telephony/dahdi-linux-  
complete/releases/dahdi-linux-complete-2.4.1.2+2.4.1.tar.gz  
  
# wget http://downloads.asterisk.org/pub/telephony/libpri/releases/libpri-  
1.4.11.5.tar.gz
```

Otra alternativa es dirigirse a la página de descargas de Asterisk, [\*http://www.asterisk.org/downloads\*](http://www.asterisk.org/downloads)<sup>11</sup> donde se podrá descargar de igual manera los paquetes de instalación.

### 4.2.1 INSTALACIÓN DE PAQUETES

Se desempaquetan los paquetes previamente descargados en el directorio `/usr/src/` con el comando `#tar zxvf`.

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

```
#tar zxvf asterisk-1.8.3.2.tar.gz
```

```
#tar zxvf dahdi-linux-complete-2.4.1.2+2.4.1.tar.gz
```

```
#tar zxvf libpri-1.4.11.5.tar.gz
```

Instrucciones para la instalación de cada paquete de instalación en el siguiente orden:

#### **4.2.1.1 LIBPRI**

Librería de uso opcional, es una librería de código abierto que encapsula los protocolos para soporte sobre ISDN.

```
# cd /usr/src/libpri-1.4.11.5
```

```
# make clean
```

```
# make
```

```
# make install
```

#### **4.2.1.2 DAHDI**

Es el software usado por Asterisk para la interacción con la telefonía análoga y digital.

```
# cd /usr/src/dahdi- linux-complete-2.4.1.2+2.4.1
```

```
# make clean  
  
# make  
  
# make install  
  
# make config
```

#### **4.2.1.3 ASTERISK**

Se compilan e instalan los módulos de Asterisk, así como módulos de ejemplo.

```
# cd /usr/src/asterisk-1.8.3.2  
  
# make clean  
  
# ./configure  
  
# make menuselect .- seleccionar los módulos adicionales  
  
# make install  
  
# make samples  
  
# make config
```

Una vez completada la instalación de los paquetes requeridos, se inicia Asterisk con el siguiente comando: `# asterisk -r`

Si no está ejecutándose entonces se ingresa: `# asterisk -cgvvv`

## 4.2.2 CONFIGURACIÓN DE ASTERISK

Los archivos de configuración existentes en la carpeta `/etc/asterisk/` pueden ser modificados con el propósito de cumplir las exigencias de la implementación de una central telefónica. Entre ellos, nos compete los siguientes: `sip.conf`, `extensions.conf` a cuales creamos copias de respaldo. Al finalizar la realización de cambios, para que tomen efecto se ejecutan en la consola de Asterisk los siguientes comandos:

```
CLI> module reload chan_sip.so
```

```
CLI> dialplan reload
```

### 4.2.2.1 ARCHIVO SIP

En el archivo `sip.conf` se configuran las opciones y parámetros por defecto de canal para los usuarios o *peers* registrados en el archivo. Además se configuran de manera particular los parámetros para cada usuario, a continuación se muestra un ejemplo.

```
[general]
```

```
context=noautenticadas
```

```
allowguest=no
```

```
srvlookup=yes
```



udpbindaddr=0.0.0.0

tcpenable=no

qualify=yes

language=es

[lsanchez]

type=friend

context=internos

host=dynamic

nat=yes

secret=1234

dtmfmode=auto

disallow=all

allow=ulaw

[mmolina]

type=friend

context=internos

host=dynamic

nat=yes

secret=1234

dtmfmode=auto

disallow=all

#### 4.2.2.2 ARCHIVO EXTENSIONS

En el archivo extensions.conf, también conocido como el plan de marcado se configuran los contextos y extensiones, los cuales definen el comportamiento de la central telefónica.

[general]

autofallthrough=yes

clearglobalvars=no

static=yes

priorityjumping=no

[internos]

;EXTENSION PRINCIPAL DE BIENVENIDA

exten => 1,1,Answer()

exten => 1,n,Playback(bienvenida)

exten => 1,n,AGI(welcome.agi)

exten => 1,n,WaitExten()

[periodico]

exten

=>

start,1,AGI(eluniverso.agi,http://www.eluniverso.com/rss/portada.xml)

[diccionario]

exten => start,1,Answer()

same => n,AGI(decode.agi)

[blog]

exten => start,1,AGI(blogsalud.agi,http://www.saludynutricion.es/feed/)

[libro]

exten => start,1,Answer()

same => n,AGI(decodeLetra.agi)

same => n,Hangup

[wikipedia]

exten => start,1,Answer()

same => n,AGI(decodewiki.agi)

same => n,Hangup

### **4.3 INSTALACIÓN DEL NAVEGADOR VXI**

Desde la página web de i6net en la sección de descargas en donde se enlistan los paquetes de instalación para las distintas versiones de sistemas operativos Linux. Al encontrar la opción adecuada para CentOS, se procede a descargarlo en el directorio `/usr/src/` y luego a descomprimirlo con:

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

```
#tar zxvf vxml_V6.1_2011-05-28_202_i686_CentOS-5.4.tar.gz
```

### **4.3.1 INSTALACIÓN DEL PAQUETE**

Se empieza con la ubicación del directorio descomprimido del paquete de instalación, y luego ejecutamos lo siguiente:

```
# cd vxml_V6.1_2011-05-28_202_i686_CentOS-5.4  
# ./install.sh
```

Existen módulos de complemento de VXI para versiones de Asterisk más actuales, por lo que se reemplazan las existentes con el siguiente comando.

```
# cp ./modules/app_vxml.so.asterisk_v1.8.4.1 /usr/lib/asterisk/modules
```

Se reinicia Asterisk y luego en su consola se verifica que se hayan cargado los módulos correctamente mediante los siguientes comandos:

```
CLI> core restart now
```

```
CLI> core show help vxml
```

### **4.3.2 CONFIGURACIÓN DEL NAVEGADOR VXI**

Los archivos de configuración a modificar para cumplir con la implementación son: `client.cfg` y `defaults.xml` los cuales se encuentran en el directorio `/etc/openvxi/`. De igual manera es recomendable hacer copias de

respaldo. Luego de la realización de cambios, para que tomen efecto se reinicia la aplicación con los siguientes comandos:

```
# service openvxi restart
```

#### 4.3.2.1 ARCHIVO CLIENT

Contiene la configuración del cliente del Navegador VXi, donde incluyen parámetros base para la interacción con otros componentes y complementos de la aplicación. Modificaremos la sección del TTS Server, reemplazando la ruta donde se encuentra el recurso encargado de convertir el texto en voz, el archivo `tts.php`. VXi trabaja con un servidor web para sus componentes, en nuestro caso utilizando Apache la ruta será `http://localhost/tts.php`.

```
client.prompt.resource.0.uri          VXIString  http://localhost/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
```

#### **4.3.2.2 ARCHIVO TTS**

Es un script creado en PHP, el cual recibe entradas en texto desde el interpretador de VXI y retorna salidas en audio. Este archivo originalmente no viene implementado para trabajar con Festival TTS, pero está basado en script de la documentación de VXI, para una versión más ligera de Festival como es Flite (Festival-light). Partiendo de esto reemplazamos y editamos líneas de código, esencialmente:

```
$text2wave_cmd = sprintf("text2wave -o %s -scale 50 %s -F 8000",  
$wave_file, $speech_file);  
exec($text2wave_cmd);
```

El comando text2wave, es una utilidad de Festival TTS la cual crea un archivo de extensión .wav, a partir de un archivo de texto. Adicionalmente, se especifica la frecuencia (-F) y la escala de volumen (-scale).

#### **4.3.2.3 ARCHIVO DEFAULTS**

Este archivo XML contiene propiedades y opciones por defecto del interpretador de VXML del navegador VXI. Estas propiedades pueden ser

sobrescritas particularmente en cada archivo vxml. Se modificaron los mensajes predeterminados de información y ayuda, así como los mensajes de error adicionando como atributo *xml:lang="es"*.

```
<property name="swirec_extra_nbest_keys" value=""/>
<catch event="cancel"><!-- do nothing --></catch>
<error>
  <if cond="_message != undefined">
    <log>Error: <value expr="_event"/>, <value expr="_message"/></log>
  <else/><log>Error: <value expr="_event"/></log>
  </if>
  Ha ocurrido un error. Saliendo del programa. <exit/>
</error>
<catch event="exit"><exit/></catch>
<help><prompt xml:lang="es">El contexto de ayuda no esta
disponible.</prompt><reprompt/></help>
<help count="5"><exit/></help>
<noinput count="1"><reprompt/></noinput>
<noinput count="2"><prompt xml:lang="es">Lo siento no lo
escuche.</prompt>
<reprompt/></noinput>
<noinput count="3"><prompt xml:lang="es">Podria repetir por
favor.</prompt><reprompt/></noinput>
```

```

<noinput count="4"><prompt xml:lang="es">No lo escuche, repita por
favor.</prompt><reprompt/></noinput>

<noinput count="5"><exit/></noinput>

<nomatch count="1"><prompt xml:lang="es">Disculpe, aun no le
entiendo.</prompt><reprompt/></nomatch>

<nomatch count="2"><prompt xml:lang="es">No entendi, hable fuerte y
claro.</prompt><reprompt/></nomatch>

<nomatch count="3"><prompt xml:lang="es">Por repita por ultima
vez.</prompt><reprompt/></nomatch>

<nomatch count="4"><prompt xml:lang="es">Lo siento mucho, no pude
entender.</prompt><exit/></nomatch>

<catch event="connection.disconnect"><exit/></catch>

<catch><if cond="_message != undefined"><log>Error: <value
expr="_event"/>, <value expr="_message"/></log><else/><log>Error: <value
expr="_event"/></log></if>

<prompt xml:lang="es">Ocurrio un error en la aplicacion. Saliendo del
programa</prompt><exit/></catch>

```

#### 4.4 INSTALACIÓN DE FESTIVAL TTS

El código a continuación descarga e instala las librerías de Festival TTS, así como los módulos para la interacción con Asterisk.



```
# yum install festival festival-devel
```

#### 4.4.1 FESTIVAL Y ASTERISK

Se verifica que se instaló correctamente, accediendo a Asterisk y ejecutando los siguientes comandos:

```
CLI> module unload app_festival
```

```
CLI> module load app_festival
```

Se modifica el archivo **festival.conf** del directorio `/etc/asterisk/` agregando los siguientes parámetros para activar Asterisk con festival. Se recargan los módulos una vez hecho los cambios.

```
[general]
```

```
host=localhost
```

```
port=1314
```

```
usecache=yes
```

```
cachedir=/var/lib/asterisk/festivalcache/
```

```
festivalcommand=(tts_textasterisk "%s" 'file')(quit)\n
```

#### 4.4.2 FESTIVAL EN ESPAÑOL

Festival TTS viene predefinido en el idioma inglés, por lo que se deben realizar cambios en la configuración. Para esto se convierten los paquetes de voz, que vienen para Debian, en paquetes rpm de CentOS. Se descarga el programa Alien<sup>12</sup>:

```
# cd /usr/src
# wget http://ftp.de.debian.org/debian/pool/main/a/alien/alien_8.78.tar.gz
```

Se lo descomprime para luego entrar en la carpeta y se lo instala:

```
# tar zxvf alien_8.78.tar.gz
# cd alien
# perl Makefile.PL
# make
# make install
```

Se descarga el paquete Debian de la voz en español masculina<sup>13</sup> y se lo transforma con Alien a paquetes .rpm:

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

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

Ahora con el comando rpm se ejecuta el paquete transformado:

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

Revisando en el directorio `/usr/share/festival/voices/spanish/` se encontrará la carpeta **JuntaDeAndalucia\_es\_pa\_diphone**. Se debe determinar la nueva voz a usar, para esto se dirige a `/usr/share/festival/festival.scm` y antes de la última línea del archivo se agrega:

```
;(language__spanish)
```

```
(set! voice_default 'voice_JuntaDeAndalucia_es_pa_diphone)
```

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

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

Apply tts to STRING. This function is specifically designed for use in server mode so a single function call may synthesize the string. This function name may be added to the server safe functions."

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

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

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

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

Luego de esto se inicia el servidor de Festival con el comando:

```
# /usr/bin/festival --server
```

#### **4.5 INSTALACIÓN DE SPHINX**

Se realiza la descarga de los siguientes paquetes en el directorio /usr/src/:

- Sphinx2 versión 0.6

[http://sourceforge.net/project/showfiles.php?group\\_id=1904](http://sourceforge.net/project/showfiles.php?group_id=1904)

- Módulos de Perl para Sphinx2

<http://search.cpan.org/CPAN/authors/id/D/DJ/DJHD/Speech-Recognizer-SPX-0.0801.tar.gz>

- Modelo Acústico

<http://www.speech.cs.cmu.edu/sphinx/models/hmm/communicator-2000-11-17-2.tgz>

- Sphinx AGI, sr\_client y sr\_server

<http://www.syednetworks.com/asterisk-sphinx.rar>

- Módulos de Perl para AGI

<http://search.cpan.org/~jamesgol/asterisk-perl-0.09/lib/Asterisk/AGI.pm>

## 4.5.1 INSTALACIÓN DE PAQUETES

Luego de la descarga, se procede a descomprimirlos con el comando *tar* de la siguiente manera:

```
# tar zxvf sphinx2-0.6.tar.gz
```

```
# tar zxvf Speech-Recognizer-SPX-0.0801.tar.gz
```

```
# tar zxvf communicator-2000-11-17-2.tgz
```

```
# tar zxvf asterisk-perl-0.09.tar.gz
```

### 4.5.1.1 SPHINX 2

Se ingresa al directorio descomprimido y se ejecuta el siguiente código para instalar los archivos de código fuente de Sphinx2.

```
# cd sphinx2-0.6
```

```
# ./configure --prefix=/usr/local/sphinx
```

```
# make
```

```
# make install
```

#### **4.5.1.2 SPEECH RECOGNIZER (RECONOCIMIENTO DE DIÁLOGO)**

Se instalan los módulos de Perl para Sphinx2 con los siguientes comandos:

```
# cd Speech-Recognizer-SPX-0.0801
# perl Makefile.PL --sphinx-prefix=/usr/local/sphinx
# make
# make install UNINSTALL=1
```

#### **4.5.1.3 MODELO ACÚSTICO**

Se coloca el modelo acústico de Sphinx en la carpeta hmm de la siguiente manera:

```
#cd communicator-2000-11-17-2/
#mv sphinx_2_format
/usr/local/sphinx/share/sphinx2/model/hmm/communicator
```

#### **4.5.1.4 MÓDULOS PERL AGI**

Se instalan los módulos de la siguiente manera. Luego se reinicia Asterisk para que tomen efecto los cambios.

```
# cd asterisk-perl-0.09/  
  
# perl Makefile.PL  
  
# make  
  
# make install
```

Un método alternativo para la instalación de módulos y complementos en Perl, es usar la utilidad CPAN.

## **4.5.2 CONFIGURACIÓN DE COMPONENTES**

Los componentes a configurar son los archivos `sr_server.dat` y los archivos del diccionario de palabras a reconocer por Sphinx. Luego se verifica la respuesta de interpretación del servidor.

### **4.5.2.1 CONFIGURACIÓN DE SR\_SERVER**

Es el archivo servidor de Sphinx, encargado de reconocer los archivos de audio enviados como parámetros, retornando la palabra interpretada siempre que se encuentre en el diccionario. Se encuentra implementado en Perl, pero produce error de compilación por el encabezado. Por lo que se lo editó reemplazando con:

```
# !/bin/sh
```

```
exec perl -w -x $0
```

```
# !perl
```

Luego de guardar los cambios, se inicia el siguiente script para evitar errores al cargar el server, y dejamos corriendo el sr\_server en espera de algún requerimiento.

```
# export LD_LIBRARY_PATH=/usr/local/sphinx/lib
```

```
#!/usr/local/bin/sr_server &
```

#### 4.5.2.2 CONFIGURACIÓN DEL DICCIONARIO

Se necesitan una serie de archivos para el reconocimiento del audio. Estos se generan con una aplicación web conocida como Imtools. Partiendo por la creación de un archivo de texto conteniendo las palabras deseadas en inglés. Para nuestra implementación serán: OK, SEE, NO, separadas por un salto de línea. La aplicación genera una carpeta con los archivos de extensión **.dic**, **.log\_pronounce**, **.lm**, **.sent** y **.vocab** a cuales se les cambiará en nombre de la carpeta, así como el nombre de los archivos dentro de ella, por “confirm”.

La carpeta confirm deberá ser movida a /usr/local/sphinx/share/sphinx2/model/lm/ de donde el server de Sphinx usa



los archivos para el reconocimiento del habla.

#### **4.5.3 VERIFICACIÓN DE RECONOCIMIENTO**

Se verifica que el sr\_server funciona corriendo el archivo sr\_client con archivos de audio wav como parámetro, y observando la respuesta del server. Los archivos de audio deberán tener grabado palabras pertenecientes al diccionario, y usamos el comando:

```
$ /usr/local/bin/sr_client OK.wav
```

```
$ /usr/local/bin/sr_client NO.wav
```

En el sr\_server deberá respectivamente aparecer:

```
SERVER RESULT: OK
```

```
SERVER RESULT: NO
```

## **CAPITULO 5**

### **FUNCIONAMIENTO**

## 5.1 INICIO DE SERVICIOS

Ingresar al terminal para inicializar los servicios necesarios:

1. `service httpd restart`
2. `service openvxi restart`
3. `/usr/bin/festival –server`
4. `export LD_LIBRARY_PATH=/usr/local/sphinx/lib`
5. `/usr/local/bin/sr_server &`
6. Asterisk –r

La reproducción de los repositorios digitales disponibles se inicia al marcar el número de prueba 1800-12345-6789, continuación en la figura 5.1 se visualiza la ejecución del script `welcome.agi` que espera un comando de voz por parte del usuario para acceder a una de las siguientes opciones:

1. Periódico
2. Diccionario
3. Wikipedia
4. Blog
5. Libros

```

[root@localhost ~]# asterisk -r
Asterisk 1.8.4.2, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public License version 2 and other licenses; you are welcome to redistribute it under certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.8.4.2 currently running on localhost (pid = 3279)
Verbosity is at least 3
== Using SIP RTP CoS mark 5
-- Executing [l@internos:1] Answer("SIP/mmolina-00000000", "") in new stack
-- Executing [l@internos:2] Playback("SIP/mmolina-00000000", "bienvenida") in new stack
-- <SIP/mmolina-00000000> Playing 'bienvenida.slin' (language 'es')
-- Executing [l@internos:3] AGI("SIP/mmolina-00000000", "welcome.agi") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/welcome.agi
-- Playing 'periodico' (escape_digits=) (sample_offset 0)
-- Playing 'beep' (escape_digits=) (sample_offset 0)
welcome.agi: INTENTOS:: 0
welcome.agi: CONFIRM:
-- Playing 'diccionario' (escape_digits=) (sample_offset 0)
-- Playing 'beep' (escape_digits=) (sample_offset 0)
welcome.agi: INTENTOS:: 1
welcome.agi: CONFIRM:
-- Playing 'wikipedia' (escape_digits=) (sample_offset 0)
-- Playing 'beep' (escape_digits=) (sample_offset 0)
welcome.agi: INTENTOS:: 2
welcome.agi: CONFIRM:
-- Playing 'blog' (escape_digits=) (sample_offset 0)
-- <SIP/mmolina-00000000>AGI Script welcome.agi completed, returning -1
localhost*CLI> █

```

**Fig. 5.1 Ejecución del script welcome.agi**

## 5.2 FUNCIONAMIENTO PARCIAL

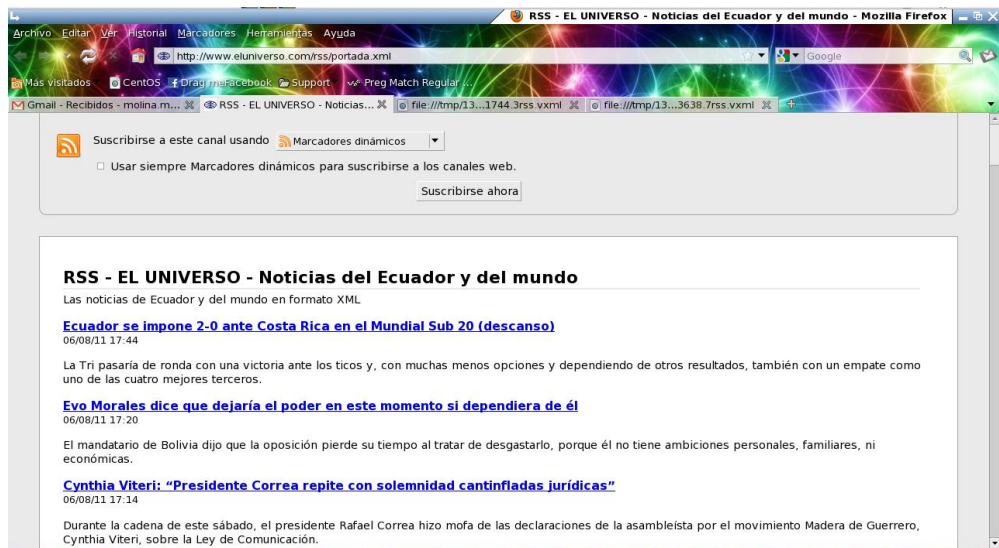
### 5.2.1 PERIÓDICO

La página a consultar es la portada en rss que pertenece al Diario El

Universo, con la siguiente dirección web

<http://www.eluniverso.com/rss/portada.xml>, como se muestra en la

figura 5.2.



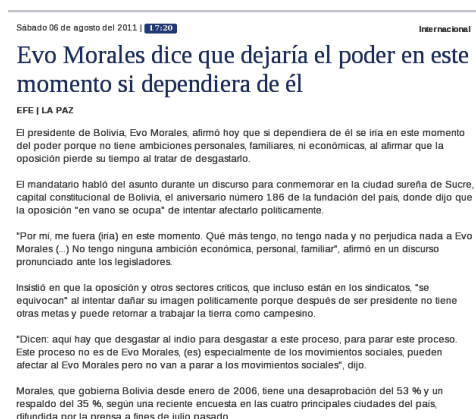
**Fig. 5.2 Portada del Diario El Universo**

El script principal welcome.agi ejecuta el script llamado eluniverso.agi (anexo 2), como se muestra en la figura 5.3, el cual genera el archivo eluniverso.xml que se encuentra en el anexo 3.

```
[root@localhost ~]# asterisk -r
Asterisk 1.8.4.2, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.8.4.2 currently running on localhost (pid = 3078)
Verbosity is at least 3
== Using SIP RTP CoS mark 5
-- Executing [6@internos:1] AGI("SIP/mmolina-00000006", "eluniverso.agi,http://ww
w.eluniverso.com/rss/portada.xml") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/eluniverso.agi
-- AGI Script Executing Application: (Vxml) Options: (file:///tmp/1312672216.6rss
.vxml)
-- <SIP/mmolina-00000006> Playing '/tmp/cacheContent/default/en-GB/cache_VCU11cCT
fPMLCJ5AdaNQ80==.slin' (language 'es')
-- <SIP/mmolina-00000006> Playing '/tmp/cacheContent/default/es-ES/cache_nm9H10DV
oyuqofBeNe0UD==.slin' (language 'es')
-- <SIP/mmolina-00000006> Playing '/tmp/cacheContent/default/es-ES/cache_KiIscsqh
7s+I5657-sjwBy==.slin' (language 'es')
-- <SIP/mmolina-00000006> Playing '/tmp/cacheContent/default/es-ES/cache_zm53d0HB
a+I0L7cRoHQGDq==.slin' (language 'es')
-- <SIP/mmolina-00000006> Playing '/tmp/cacheContent/default/en-GB/cache_wiKio7Ti
9SEs43yyZ4f9DP==.slin' (language 'es')
localhost*CLI> █
```

**Fig. 5.3 Ejecución del archivo eluniverso.agi**

El usuario durante la lectura puede elegir entre avanzar a la siguiente noticia o leer la noticia completa, en nuestro ejemplo se ha decidido leer la noticia completa. Como se muestra en la figura 5.4.



**Fig. 5.4 Noticia completa número 2**

Para leer las noticias de manera completa se ejecuta un archivo xml temporal creado por el script full3eluniverso.php. Ver contenido en anexo 4 y ejecución del script full3eluniverso.php en la figura 5.5.

```

This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.8.4.2 currently running on localhost (pid = 3078)
Verbosity is at least 3
== Using SIP RTP CoS mark 5
-- Executing [6@internos:1] AGI("SIP/mmolina-0000006", "eluniverso.agi,http://ww
w.eluniverso.com/rss/portada.xml") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/eluniverso.agi
-- AGI Script Executing Application: (Vxml) Options: (file:///tmp/1312672216.6rss
fPMLCJ5AdaNQBO==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/default/es-ES/cache_nm9H1QDV
oyuqofBeNeOUD==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/default/es-ES/cache_KiIscsqh
7s+I5G57-sjwBy==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/default/es-ES/cache_zm53d0HB
a+I0L7cRoHQGDq==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/default/en-GB/cache_wiKio7Ti
9SEs43yyZ4f9DP==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/default/es-ES/cache_lguzl53u
77l2eErbkH7mBK==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/default/es-ES/cache_22M1E5IA
j-3BYuPSJI+DBa==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/default/en-GB/cache_wiKio7Ti
9SEs43yyZ4f9DP==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/localhost/en-GB/cache_XbF6id
ZkUIyqo0-yZgQwAY==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/localhost/en-GB/cache_oIMKv8
c0GFLLEyI0jPCW==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/default/en-GB/cache_VCUi1cCTFF
MLCJ5AdaNQBO==.slin' (language 'es')
-- <SIP/mmolina-0000006> Playing '/tmp/cacheContent/default/es-ES/cache_nm9H1QDVoy
uqofBeNeOUD==.slin' (language 'es')
localhost*CLI> █

```

**Fig 5.5 Ejecución de archivo full3universo.php**

Luego de haber reproducido la noticia, el usuario puede decidir en regresar a la portada del diario El Universo o salir del sistema. Se ha optado en salir del sistema y automáticamente se ejecuta el script principal llamado welcome.agi para escuchar de nuevo las opciones del sistema, puede visualizarse en la figura 5.6.

```

-- <SIP/mmolina-0000007>AGI Script eluniverso.agi completed, returning 0
-- Executing [1@internos:1] Answer("SIP/mmolina-0000007", "") in new stack
-- Executing [1@internos:2] Playback("SIP/mmolina-0000007", "bienvenida") in new s
tack
-- <SIP/mmolina-0000007> Playing 'bienvenida.slin' (language 'es')
== Spawn extension (internos, 1, 2) exited non-zero on 'SIP/mmolina-0000007'
-----*CLI> █

```

**Fig. 5.6 Finaliza script eluniverso.agi**

## 5.2.2 DICCIONARIO

Se ha escogido el diccionario de la Real Academia Española consultando su página web <http://www.rae.es/rae.html>.

Primero se ejecuta el script `decode.agi` (anexo 5) donde el usuario debe ingresar mediante el teclado numérico la palabra a buscar en el diccionario, como por ejemplo, para consultar la palabra “tierno”, debe teclear los siguientes números:

T = una vez el número 8

I = tres veces el número 4

E = dos veces el número 3

R = tres veces el número 7

N = dos veces el número 6

0 = número cero para distinguir las letras que se encuentran en el mismo número.

O = tres veces el número 6

\* = para finalizar la palabra

Este proceso se puede visualizar en la figura 5.7



```

-- Executing [2@internos:1] Answer("SIP/mmolina-00000008", "") in new stack
-- Executing [2@internos:2] AGI("SIP/mmolina-00000008", "decode.agi") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/decode.agi
-- Playing 'busqueda' (escape_digits=) (sample_offset 0)
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter8fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter4fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter4fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter4fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter3fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter3fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter7fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter7fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter6fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter6fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter0fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')

```

**Fig. 5.7 Ingreso de palabra a buscar**

Luego de terminar el ingreso, se pedirá la confirmación de su entrada para ejecutar el script diccionario.agi (anexo 6) caso contrario el sistema le permitirá el ingreso de una nueva palabra. En la figura 5.8 se aprecia la palabra ingresada.

```

decode.agi: este es el caracter3fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter7fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter7fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter6fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter6fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter0fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter6fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter6fin
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
decode.agi: este es el caracter*fin
decode.agi: Code: 000444033077066006660*
decode.agi: palabra a buscar TIERNO
-- Playing 'palabraing' (escape_digits=) (sample_offset 0)
-- AGI Script Executing Application: (Festival) Options: (TIERNO)
== Parsing '/etc/asterisk/festival.conf': == Found
-- Playing 'confirmacion' (escape_digits=) (sample_offset 0)
-- <SIP/mmolina-00000008> Playing 'silence/5.ulaw' (language 'es')
-- Playing 'esperexfavor' (escape_digits=) (sample_offset 0)
-- AGI Script Executing Application: (AGI) Options: (diccionario.agi,TIERNO,TIERNO)
-- Launched AGI Script /var/lib/asterisk/agi-bin/diccionario.agi
-- AGI Script Executing Application: (Vxml) Options: (file:///tmp/dic.vxml)

```

**Fig. 5.8 Ejecución de script diccionario.agi**

El script diccionario.agi generará un archivo xml llamado diccionario.xml (anexo 7) el contenido de la página web que contiene el o los significados de la palabra que consulta como se puede ver en la figura 5.9.

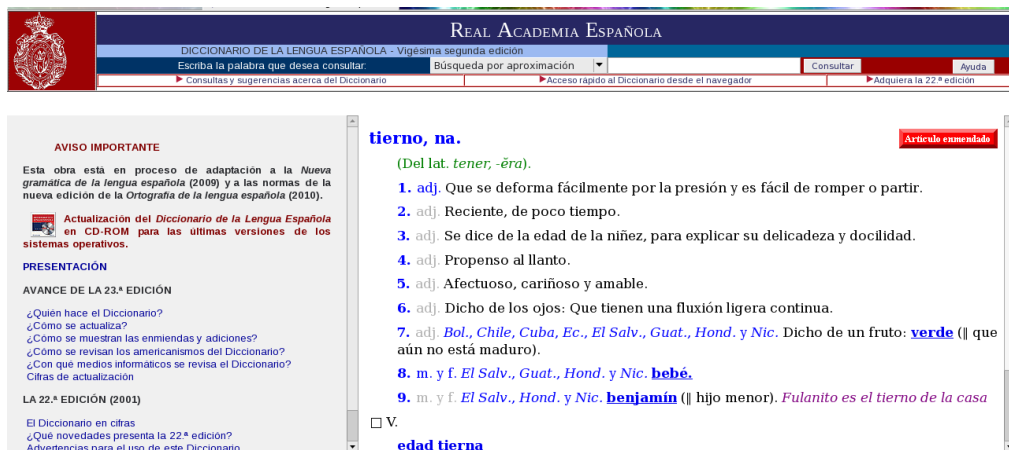


Fig. 5.9 Página web con los significados de la palabra ejemplar

Como se puede apreciar en el archivo diccionario.xml sólo se encontraron 6 definiciones porque el sistema no leerá las definiciones que contengan vínculos hacia otras páginas.

Para salir del diccionario el usuario ingresará la opción cero cuando desee y volverá al script principal como se puede apreciar en la figura 5.10.

```
-- <SIP/mmolina-00000008>AGI Script diccionario.agi completed, returning 0
-- <SIP/mmolina-00000008>AGI Script decode.agi completed, returning 0
-- Executing [1@internos:1] Answer("SIP/mmolina-00000008", "") in new stack
-- Executing [1@internos:2] Playback("SIP/mmolina-00000008", "bienvenida") in new s
tack
-- <SIP/mmolina-00000008> Playing 'bienvenida.slin' (language 'es')
localhost*CLI> █
```

Fig. 5.10 Finaliza el diccionario

### 5.2.3 WIKIPEDIA

La página de la wikipedia en idioma español es <http://es.wikipedia.org/wiki/Wikipedia:Portada>, en la cual basaremos nuestro sistema.

El usuario, al igual que en el diccionario, deberá ingresar la palabra a buscar mediante el teclado numérico, la cual será recibida por el script `decodewiki.agi` (ver anexo 8).

Se presentan dos casos:

- I. Cuando la palabra tiene un solo significado.
- II. Cuando la palabra tiene varios significados.

Caso I

Para el siguiente ejemplo se ha consultado la palabra “gorra” como se aprecia en la figura 5.11.

```

-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter6fin
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter6fin
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter6fin
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter7fin
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter7fin
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter7fin
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter7fin
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter7fin
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter7fin
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter2fin
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
jcodewiki.agi: este es el caracter*fin
jcodewiki.agi: Code: 040666077700777020*
jcodewiki.agi: palabra a buscar GORRA
-- Playing 'palabraing' (escape_digits=) (sample_offset 0)
-- AGI Script Executing Application: (Festival) Options: (gorra)
== Parsing '/etc/asterisk/festival.conf': == Found
[Aug 7 22:49:41] WARNING[4818]: : : festival_cli
t: connect to server failed
-- Playing 'confirmacion' (escape_digits=) (sample_offset 0)
-- <SIP/mmolina-00000000> Playing 'silence/5.ulaw' (language 'es')
-- AGI Script Executing Application: (AGI) Options: (jcodewiki.agi)
-- Launched AGI Script /var/lib/asterisk/agi-bin/jcodewiki.agi
-- Playing 'busquedawiki' (escape_digits=) (sample_offset 0)

```

Fig. 5.11 Visualización de palabra a buscar en wikipedia

Luego de confirmar el ingreso, se ejecuta el script wiki.agi (anexo 9), el cual genera el archivo wiki1.xml (anexo 10). Para consultar la página que se muestra en la figura 5.12 se genera un script php llamado full3wiki.php (anexo 11).



**Fig. 5.12 Consulta la palabra “gorra”**

Luego de reproducir todo el contenido, se decide salir de la wikipedia y regresar al menú principal.

## Caso II

Se ingresa la palabra a consultar: “celular”

```
-- <SIP/mmolina-0000000c> Playing 'silence/5.ulaw' (language 'es')
fecodewiki.agi: este es el caracter7fin
-- <SIP/mmolina-0000000c> Playing 'silence/5.ulaw' (language 'es')
fecodewiki.agi: este es el caracter7fin
-- <SIP/mmolina-0000000c> Playing 'silence/5.ulaw' (language 'es')
fecodewiki.agi: este es el caracter*fin
fecodewiki.agi: Code: 0222033055508805550207770*
fecodewiki.agi: palabra a buscar CELULAR
-- Playing 'palabraing' (escape_digits=) (sample_offset 0)
-- AGI Script Executing Application: (Festival) Options: (celular)
== Parsing '/etc/asterisk/festival.conf': == Found
localhost*CLI> █
```

**Fig. 5.13 Ingreso de palabra “celular”**

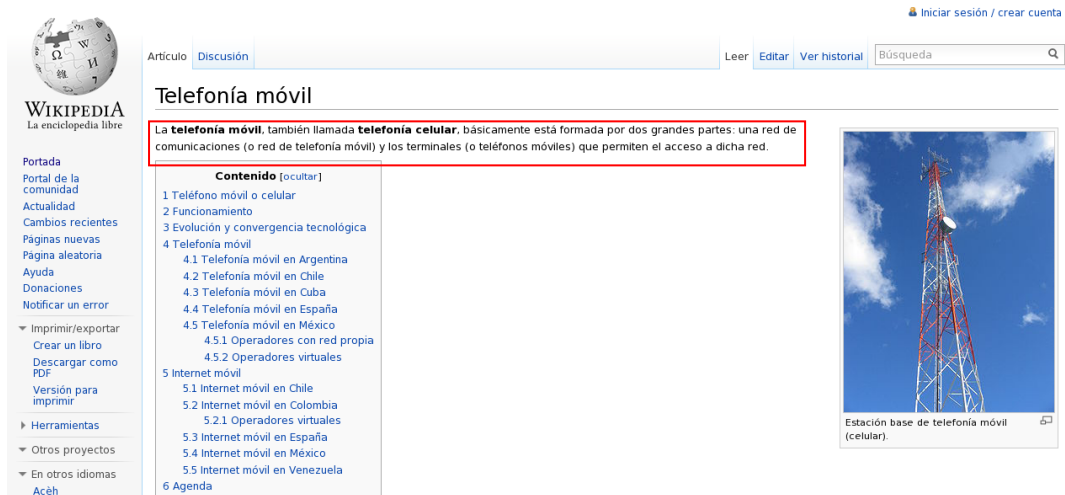
Se ejecuta el script wiki.agi, el cual genera un xml llamado wiki2.xml (anexo 12) donde se verificará que el número de definiciones encontradas han sido

tres, tal como se visualiza en la figura 5.14.



**Fig. 5.14 “Celular” Palabra ambigua**

El usuario puede decidir si leer el significado completo de algunas de las palabras halladas, en el siguiente ejemplo se ha decidido leer el segundo significado, para ello se ejecuta el script full3wiki.php, el cual consulta la página web del teléfono celular. Ver la figura 5.15.



**Fig. 5.15** Página web con significado completo.

Se decide salir de la wikipedia, por lo cual se retorna al script principal tal como se aprecia en la figura 5.16.

```

-- <SIP/mmolina-000000c>AGI Script wiki.agi completed, returning 0
-- <SIP/mmolina-000000c>AGI Script decodewiki.agi completed, returning 0
-- Executing [1@internos:1] Answer("SIP/mmolina-000000c", "") in new stack
-- Executing [1@internos:2] Playback("SIP/mmolina-000000c", "bienvenida") in
new stack
-- <SIP/mmolina-000000c> Playing 'bienvenida.slin' (language 'es')

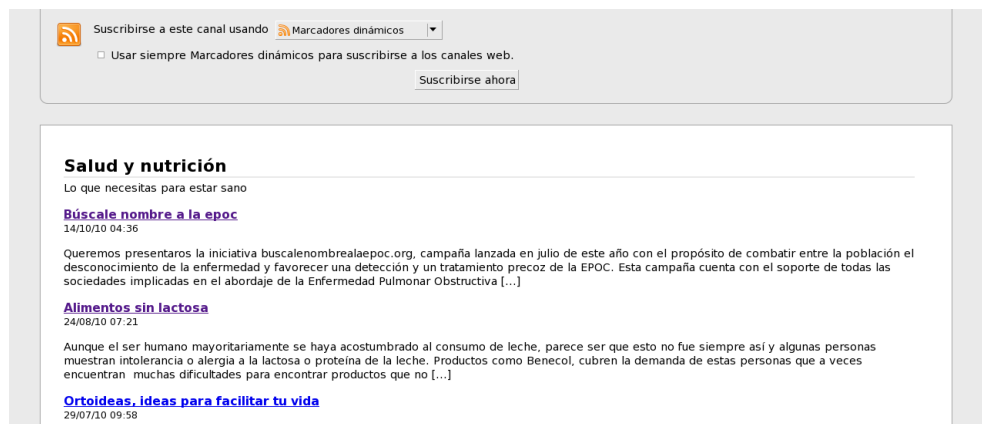
```

**Fig. 5.15** Finaliza wikipedia

### 5.2.4 BLOG

Se ha escogido el blog de salud y nutrición que se encuentra en la página <http://www.saludynutricion.es/feed/>.

Su manejo es similar a la lectura del periódico, ya que se reproducen los artículos que se encuentran publicados en la página rss del blog como puede ver en la figura 5.16.



**Fig. 5.16** Página rss del blog

Mediante el script `blog.agi` (anexo 13) como se aprecia en la figura 5.17, el cual generará el archivo `blog.xml` (anexo 14).

```

-- <SIP/mmolina-00000008>AGI Script diccionario.agi completed, returning 0
-- <SIP/mmolina-00000008>AGI Script decode.agi completed, returning 0
-- Executing [1@internos:1] Answer("SIP/mmolina-00000008", "") in new stack
-- Executing [1@internos:2] Playback("SIP/mmolina-00000008", "bienvenida") in new s
tack
-- <SIP/mmolina-00000008> Playing 'bienvenida.slin' (language 'es')
== Spawn extension (internos, 1, 2) exited non-zero on 'SIP/mmolina-00000008'
== Using SIP RTP CoS mark 5
-- Executing [3@internos:1] AGI("SIP/mmolina-00000009", "blogsaud.agi,http://www.s
aludynutricion.es/feed/") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/blogsaud.agi
-- AGI Script Executing Application: (Vxml) Options: (file:///tmp/1312674143.9rss.v
xml)
-- <SIP/mmolina-00000009> Playing '/tmp/cacheContent/default/en-GB/cache_Fe0UR9LeL
n3H6lqb4o1Ar==.slin' (language 'es')
-- <SIP/mmolina-00000009> Playing '/tmp/cacheContent/default/es-ES/cache_7z8dlizNKC
dK9aIVkk2pAx==.slin' (language 'es')
-- <SIP/mmolina-00000009> Playing '/tmp/cacheContent/default/es-ES/cache_7sjyfg6L9u
T9SFgu18iPCL==.slin' (language 'es')
-- <SIP/mmolina-00000009> Playing '/tmp/cacheContent/default/es-ES/cache_rHMAPfur90
zGUdu9G8NzC4==.slin' (language 'es')
localhost*CI T> █

```

**Fig. 5.17** Ejecución del script `blogsaud.agi`

Para el presente ejemplo se ha optado en leer el artículo número dos de manera completa, para lo cual ejecuta el script `fullblogsaud.php` (anexo 15),



el cual contiene el contenido de la página que se visualiza en la figura 5.18.



**Fig. 5.18 Contenido de script fullblogsalmud.php**

Luego, se elige la opción de regresar a los titulares y posteriormente salir del diccionario, por lo cual regresa al script principal, esto se puede apreciar en la figura 5.19

```
-- <SIP/mmolina-0000000e>AGI Script blogsalmud.agi completed, returning 0
-- Executing [1@internos:1] Answer("SIP/mmolina-0000000e", "") in new stack
-- Executing [1@internos:2] Playback("SIP/mmolina-0000000e", "bienvenida") in new s
tack
-- <SIP/mmolina-0000000e> Playing 'bienvenida.slin' (Language 'es')
localhost*CLI> █
```

**Fig. 5.19 Salida del diccionario**

### 5.2.5 LIBROS

Para la opción de lectura de libros digitales se hace uso del repositorio web <http://www.quieroleer.com.ar>, el cual contiene casi 50

libros y permite leer libros en línea.

En este caso, el usuario debe ingresar mediante el teclado numérico la letra inicial del apellido del autor del libro que desee leer. El script que recoge el ingreso de la letra es decodeLetra.agi (anexo 16) como se aprecia en la figura 5.20.

```
-- Executing [4@internos:1] Answer("SIP/mmolina-000000f", "") in new stack
-- Executing [4@internos:2] AGI("SIP/mmolina-000000f", "decodeLetra.agi") in new stack
tack
-- Launched AGI Script /var/lib/asterisk/agi-bin/decodeLetra.agi
-- Playing 'autorLetra' (escape_digits=) (sample_offset 0)
-- <SIP/mmolina-000000f> Playing 'silence/5.ulaw' (language 'es')
decodeLetra.agi: este es el caracter7fin
-- <SIP/mmolina-000000f> Playing 'silence/5.ulaw' (language 'es')
decodeLetra.agi: este es el caracter7fin
-- <SIP/mmolina-000000f> Playing 'silence/5.ulaw' (language 'es')
decodeLetra.agi: este es el caracter7fin
-- <SIP/mmolina-000000f> Playing 'silence/5.ulaw' (language 'es')
decodeLetra.agi: este es el caracter*fin
decodeLetra.agi: Code: 07770*
decodeLetra.agi: letra a buscar R
-- Playing 'letraingresada' (escape_digits=) (sample_offset 0)
-- Playing 'letters/r' (escape_digits=) (sample_offset 0)
-- Playing 'confirmacion' (escape_digits=) (sample_offset 0)
-- <SIP/mmolina-000000f> Playing 'silence/5.ulaw' (language 'es')
-- Playing 'esperexfavor' (escape_digits=) (sample_offset 0)
-- AGI Script Executing Application: (AGI) Options: (libro.agi,R)
-- Launched AGI Script /var/lib/asterisk/agi-bin/libro.agi
-- AGI Script Executing Application: (Vxml) Options: (file:///tmp/1312675136.15lib.vxml)
-- <SIP/mmolina-000000f> Playing '/tmp/cacheContent/default/en-GB/cache_3wIXnqj7HLfFxyT5hnnRDb==.slin' (language 'es')
```

**Fig. 5.20 Ejecución de script decodeLetra.agi**

Luego de confirmar la letra ingresada, en el presente ejemplo es ingresada la letra “R”, se ejecuta el script libro.agi (anexo 17) que genera el archivo libro.xml (anexo 18), el cual contiene los libros encontrados con la letra “R” como puede ver en la figura. 5.21.

## Libros de autores con: R

[Rowling, J.K. - 4.Harry Potter y el cáliz de fuego](#)  
Tras otro abominable verano con los Dursley, Harry se dispone a iniciar el cuarto curso en Hogwarts, la famosa escuela de magia y hechicería. A sus catorce años ser un joven m.....

[Rowling, J.K. - 5.Harry Potter y la Orden del Fenix](#)  
Las tediosas vacaciones de verano en casa de sus tios todavia no han acabado y Harry se encuentra más inquieto que nunca. Apenas ha tenido noticias de Ron y Hermione, y preste.....

[Rowling, J.K. - 6.Harry Potter y el misterio del principe](#)  
Con dieciseis años cumplidos, Harry inicia el sexto curso en Hogwarts en medio de terribles acontecimientos que asolan Inglaterra. Elegido capitán de quidditch, los entrenamie.....

[Rowling, J.K. - 3.Harry Potter y el prisionero de Azkaban](#)  
Por la cicatriz que lleva en la frente, sabemos que Harry Potter no es un niño como los demás, sino el héroe que venció a lord Voldemort, culpable de la muerte de sus padres. ....

[Rowling, J.K. - 2.Harry Potter y la Cámara Secreta](#)  
Tras derrotar una vez más a lord Voldemort, su siniestro enemigo en Harry Potter y la piedra filosofal, Harry espera impaciente en casa de sus insoportables tios el inicio del.....

[Rowling, J.K. - 1.Harry Potter y la piedra filosofal](#)  
Harry Potter se ha quedado huérfano y vive en casa de sus abominables tios y del insoportable primo Dudley. Harry se siente muy triste y solo, hasta que un buen día recibe una.....

Libros por Autor: [Ver todos](#) · [A](#) · [B](#) · [C](#) · [D](#) · [E](#) · [F](#) · [G](#) · [H](#) · [I](#) · [J](#) · [K](#) · [L](#) · [M](#) · [N/R](#) · [O](#) · [P](#) · [Q](#) · [R](#) · [S](#) · [T](#) · [U](#) · [V](#) · [W](#) · [X](#) · [Y](#) · [Z](#)

Fig. 5.21 Resultado de búsqueda de autor con letra “R”

Dentro de las opciones disponibles está leer el libro completo, en elejemplo se decide escoger el libro 1, para lo cual se ejecuta el script full3libro.php (anexo 19) como se aprecia en la figura 5.22.

```
decodeLetra.agi: este es el caracter7fin
-- <SIP/mmolina-0000000f> Playing 'silence/5.ulaw' (language 'es')
decodeLetra.agi: este es el caracter*fin
decodeLetra.agi: Code: 07770*
decodeLetra.agi: letra a buscar R
-- Playing 'letraingresada' (escape_digits=) (sample_offset 0)
-- Playing 'letters/r' (escape_digits=) (sample_offset 0)
-- Playing 'confirmation' (escape_digits=) (sample_offset 0)
-- <SIP/mmolina-0000000f> Playing 'silence/5.ulaw' (language 'es')
-- Playing 'esperexfavor' (escape_digits=) (sample_offset 0)
-- AGI Script Executing Application: (AGI) Options: (libro.agi,R)
-- Launched AGI Script /var/lib/asterisk/agi-bin/libro.agi
-- AGI Script Executing Application: (Vxml) Options: (file:///tmp/1312675136.15lib.vxml)
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/default/en-GB/cache_3wIXnqj7HLfFxyT5hhnRDb==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/default/en-GB/cache_hbG-4PW61KrGaIUf-wRIAg==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/default/es-ES/cache_fQeiW0KvZE1Xf82z-s-jCI==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/default/es-ES/cache_yJE84tWkDUw8LQZsKK8qCq==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/default/en-GB/cache_C3M4PW1Pj3LLL182B5G5Az==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_fflvYkT3G-aK0jy0nGZA4==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_Yym-N11U6YsGm9RmT1oqBN==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing 'descending-2tone.ulaw' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_Yym-N11U6YsGm9RmT1oqBN==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_+ZBYC2mC5FexAXTTGqOmCc==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_-rKJ+XUrlw1m3i2PXFjwCQ==.slin' (language 'es')
localhost*CLI> █
```

Fig. 5.22 Ejecución de script full3libro.php



La Mansión de los Ryddle

Los aldeanos de Pequeño Hangleton seguían llamándola «la Mansión de los Ryddle» aunque hacía ya muchos años que los Ryddle no vivían en ella. Erigida sobre una colina que dominaba la aldea, tenía cegadas con tablas algunas ventanas, al tejado le faltaban tejas y la hiedra se extendía a sus anchas por la fachada. En otro tiempo había sido una mansión hermosa y, con diferencia, el edificio más señorial y de mayor tamaño en un radio de varios kilómetros, pero ahora estaba abandonada y ruinoso, y nadie vivía en ella. En Pequeño Hangleton todos coincidían en que la vieja mansión era siniestra. Medio siglo antes había ocurrido en ella algo extraño y horrible, algo de lo que todavía gustaban hablar los habitantes de la aldea cuando los temas de chismorreo se agotaban. Habían relatado tantas veces la historia y le habían añadido tantas cosas, que nadie estaba ya muy seguro de cuál era la verdad. Todas las versiones, no obstante, comenzaban en el mismo punto: cincuenta años antes, en el amanecer de una soleada mañana de verano, cuando la Mansión de los Ryddle aún conservaba su imponente apariencia, la criada había entrado en la sala y había hallado muertos a los tres Ryddle. La mujer había bajado corriendo y gritando por la colina hasta llegar a la aldea, despertando a todos los que había podido. -Están allí echados con los ojos muy abiertos! Están fríos como el hielo! ¡Y llevan todavía la ropa de la cena! Llamaron a la policía, y toda la aldea se convirtió en un hervidero de curiosidad, de espanto y de emoción mal disimulada. Nadie hizo el menor esfuerzo en fingir que le apenaba la muerte de los Ryddle, porque nadie los quería. El señor y la señora Ryddle eran ricos, esnobes y groseros, aunque no tanto como Tom, su hijo ya crecido. Los aldeanos se preguntaban por la identidad del asesino, porque era evidente que tres personas que gozan, aparentemente, de buena salud no se mueren la misma noche de muerte natural. El Ahorcado, que era como se llamaba la taberna de la aldea, hizo su agosto aquella noche, ya que todo el mundo acudió para comentar el triple asesinato. Para ello habían dejado el calor de sus hogares, pero se vieron recompensados con la llegada de la cocinera de los Ryddle, que entró en la taberna con un golpe de efecto y anunció a la concurrencia, repentinamente callada, que acababan de arrestar a un hombre llamado Frank Bryce. -Frank! -gritaron algunos-. ¡No puede ser! Frank Bryce era el jardinero de los Ryddle y vivía solo en una humilde casita en la finca de sus amos. Había regresado de la guerra con la pierna rígida y una clara aversión a las multitudes y a los ruidos fuertes. Desde entonces, había trabajado para los Ryddle. Varios de los presentes se apresuraron a pedir una bebida para la cocinera, y todos se dispusieron a oír los detalles. -Siempre pensé que era un tipo raro -explicó la mujer a los lugareños, que la escuchaban expectantes, después de apurar la cuarta copa de jerez-. Era muy huraño. Debo de haberlo

**Fig. 5.24 Contenido que ejecuta full3pagina.php**

Durante la lectura del libro el usuario puede avanzar y retroceder entre páginas, además de salir del libro. Luego de leer el número de páginas deseadas se decide salir del libro, ver en la figura 5.25 la salida del script libro.agi y el ingreso al script principal welcome.agi.

```

-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_fflvYKs'
3G-ak0jyOnGZA4==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_Yym-N111
6YsGm9RmTloqBN==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing 'descending-2tone.ulaw' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_Yym-N111
6YsGm9RmTloqBN==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_+ZBYC2m
5FexAXTTGqOmCc==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_-rKJ+XUr
lw1m3i2PXFjwCQ==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/localhost/en-GB/cache_8bl5KNW
ths0TAuHvVpCpBN==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/default/en-GB/cache_3wIXnqj7HI
fFxyT5hhnRDb==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/default/en-GB/cache_hbG-4PW6U
rGaIUf-wRIAg==.slin' (language 'es')
-- <SIP/mmolina-0000000f> Playing '/tmp/cacheContent/default/en-GB/cache_STJoeq64i
q7MMJSDX-ZAj==.slin' (language 'es')
[Aug 6 19:05:06] ERROR[7362]: : : write() returned error: f
roken pipe
[Aug 6 19:05:06] ERROR[7362]: : : write() returned error: f
roken pipe
[Aug 6 19:05:06] ERROR[7362]: : : write() returned error: f
roken pipe
[Aug 6 19:05:06] ERROR[7362]: : : write() returned error: f
roken pipe
-- <SIP/mmolina-0000000f>AGI Script libro.agi completed, returning 0
-- <SIP/mmolina-0000000f>AGI Script decodeLetra.agi completed, returning 0
-- Executing [1@internos:1] Answer("SIP/mmolina-0000000f", "") in new stack
-- Executing [1@internos:2] Playback("SIP/mmolina-0000000f", "bienvenida") in new
tack
-- <SIP/mmolina-0000000f> Playing 'bienvenida.slin' (language 'es')
== Spawn extension (internos, 1, 2) exited non-zero on 'SIP/mmolina-0000000f'
localhost*CLI> █

```

Fig. 5.25 Finaliza el libro

# CONCLUSIONES Y RECOMENDACIONES

## CONCLUSIONES

1. Ha sido posible la implementación de un sistema IVR con acceso a repositorios web basado en tecnologías libres, con un bajo costo de inversión e implementación.
2. Mediante el estándar VoiceXML fue posible desarrollar varios diálogos de voz interactivos de manera automática y personalizada.
3. Mediante los scripts AGI escritos en Perl y PHP se ha logrado comprobar que con Asterisk es posible integrarlo con varias aplicaciones.
4. Las expresiones regulares son una herramienta fundamental para acceder a información de texto que se encuentra en cualquier archivo de origen, tal como html y xml.
5. La presencia de un servidor web local como Apache, dio soporte al interpretador de VXI, posibilitando la respuesta de voz y la interacción entre distintos vxml.

## RECOMENDACIONES

1. Se recomienda trabajar con páginas web estandarizadas porque de esta manera se facilita el acceso a los datos y se obtiene una menor probabilidad de errores, debido a que en el transcurso del desarrollo del proyecto se tuvo que hacer cambios de sitios web por este motivo.
2. Implementar una aplicación de reconocimiento de voz con lenguaje español en vez de uno en inglés para proveer del servicio en su totalidad y no de manera parcial como ocurre en el proyecto, debido a que no se encontró una aplicación de reconocimiento de voz en lenguaje español con licencia gratuita.
3. A partir de este proyecto es recomendable integrar una base de datos para almacenar los sitios más visitados por cada usuario para garantizar mayor rapidez en la ejecución del sistema.
4. Para mejorar el manejo de la memoria caché es recomendable almacenar los archivos de audio generados con mayor frecuencia.



# **GLOSARIO**

## **ASTERISK - CONTEXTO**

Son listas de extensiones distinguidas por un nombre, que permite la interacción de diferentes partes de un plan de marcado. Cada extensión definida en un contexto está aislada de otra similar definida en otro contexto, por ello, cada contexto es excluyente del resto. “El contexto es el punto en el plan de marcado donde las conexiones de un canal comenzarán.”

## **ASTERISK - EXTENSIONES**

Es la instrucción o grupo de ellas que Asterisk ejecutará a causa de una llamada entrante o por dígitos marcados en un canal activo (estímulo externo). El nombre de la extensión puede ser cualquier combinación alfanumérica, y puede tener uno o más prioridades.

## **ASTERISK - PRIORIDADES**

Son números que definen el orden de seguimiento de cada instrucción, siguen una secuencia continua que inicia siempre desde UNO y prosigue con las demás que se definen por mayor comodidad con la variable “n” que representa a cualquier número subsiguiente.

## **ASTERISK - APLICACIONES**

Realiza una acción específica en el canal activo, puede ser: reproducir un

sonido, aceptar una entrada desde el teclado, buscar algo en una base de datos, colgar una llamada, marcar a otro canal, entre otros. Cada aplicación puede llevar argumentos que sean necesarios para llevarse a cabo.

### **ALIEN SOFTWARE**

Alien hace conversiones entre los formatos rpm, deb, Stampede y tgz. Si se quiere usar un paquete pensado para otra distribución, este software lo podrá preparar para instalarlo en la versión deseada.

### **BERKELEY SOFTWARE DISTRIBUTION**

En español significa distribución de software berkeley es un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

### **CPAN**

Es un módulo en Perl que brinda la utilidad de descargar software escrito en Perl desde la línea de comandos. Es análogo a buscar directamente en los repositorios de archivos ubicados es *<http://search.cpan.org/>*

### **LENGUAJES DE PROGRAMACION**

Los lenguajes de programación son las herramientas que nos permiten crear programas.

### **PHP (Pre-Procesador de Hipertexto)**

En la actualidad es el lenguaje de programación más usado, ya que fue diseñado para la creación de páginas web dinámicas, para la interpretación de lado del servidor.

### **PERL**

Este lenguaje de programación es muy adoptado por su destreza en el procesador de texto sobre todo en los scripts; es similar al lenguaje C, Shell y AWK.

## **ANEXOS**

## Anexo 1

welcome.agi

```
#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1;
use Asterisk::AGI;
my $AGI = new Asterisk::AGI;
%input = $AGI->ReadParse();
sub asr {
    use IO::Socket;
    use FileHandle;
    use IPC::Open2;
    my $file = shift or return undef;
    my $host = shift || 'localhost';
    my $port = shift || '1069';
    my $fh;
    my $remote = IO::Socket::INET->new(
        Proto => "tcp",
        PeerAddr => "$host",
        PeerPort => "$port",
    ) or return undef;
    #Idea here being that you can pass a reference to an existing file handle... not yet
    #implemented, just pass a filename.
    if (ref $file) {
        my $fh = $file;
    } else {
        open (FH, $file) || return undef;
        $fh = *FH;
    }
    $file =~ /(gsm|wav)$/;
    my $type = $1;
    if ($type !~ /gsm|wav/) {
        warn "Unknown file type ($file)";
        return undef;
    }
    $pid = open2(*SOXIN, *SOXOUT, "sox -t $type - -s -r 8000 -w -t wav - 2>/dev/null")
    || warn ("Could not open2.\n");
    binmode $fh;
    binmode SOXIN;
    binmode SOXOUT;
    binmode $remote;
    while (defined(my $b = read $fh, my($buf), 4096)) {
        last if $b == 0;
        $count += $b;
        print SOXOUT $buf;
    }
}
```

```

close SOXOUT;
$count = 0;
my $sox = undef;
while (defined(my $b = read SOXIN, my($buf), 4096)) {
    last if $b == 0;
    $count += $b;
    $sox .= $buf;
}
print $remote length($sox) . "\n";
print $remote "$sox";
close SOXIN;
$count=0;
while (defined(my $b = read $remote, my($buf), 4096)) {
    last if $b == 0;
    $count += $b;
    $result .= $buf;
}
close $fh;
close $remote;
return "$result";
}
sub confirm {
@opciones=("periodico","diccionario","wikipedia","blog","libro,");
my $tries=0;
while ($tries <= 4) {
    my $vresponse = "";
    $AGI->stream_file($opciones[$tries],"");
    $AGI->stream_file("beep","");
    $AGI->record_file("/tmp/$$", 'wav','0','3000');
    $AGI->verbose("INTENTOS::: $tries");
    $vresponse = asr("/tmp/$$.wav");
    $AGI->verbose("CONFIRM: $vresponse");
    unlink( "/tmp/$$.wav" );
    if ($vresponse =~ /CANCEL|NO/i) {
        sleep 1;
        return undef;
    } elsif ($vresponse =~ /OK|SEE/i){
        $AGI->set_variable('RESPONSE', 'YES');
        $AGI->set_context($opciones[$tries]);
        $AGI->set_extension('start');
        $AGI->set_priority(1);
        return 1;
    }
    $tries++;
    if ($tries>4){
        $tries=0;
    }
} #fin del while
}#end sub

```

```
unless ( confirm() ) {  
  #They said no  
  #AGI->set_variable('RESPONSE', 'NO');  
  $AGI->verbose("Hasta Pronto");  
  $AGI->stream_file("hastapronto", "");  
  exit;  
}
```





```

';:;!,"'";(,');
$replace = array('a','e','i','o','u','A','E','I','O','U','Ni','ni','y');
#spliteamos =) RSS - EL UNIVERSO - CATEGORIA
$parts = explode("-", $listT[1][0]);
$diario=trim($parts[1]);
$categ=str_replace($search, $replace,trim($parts[2]));
if ($num>0)
{
    if
(preg_match_all("/<description>([^\<]+)</description>|<descriptionV>/", $infoxml, $list
D))
    {
        #recogemos el uniqueid para luego crear
        #un .vxml temporal con ese nombre ej: /tmp/09232rss.vxml
        $id = $agi[uniqueid];
        $filename="/tmp/" . $id . "rss.vxml";
        #SE CREA EL VXML
        preg_match_all("/<link>(.*?)</link>/", $infoxml, $listL); //<--recojo los
links
        $vxmlstr .= '<?xml version="1.0" encoding="iso-8859-1"?>
<vxml xml:lang="es-ES" version="2.0"
xmlns="http://www.w3.org/2001/vxml">
        <var name="ingreso" expr="0"/>
        <var name="total" expr=".( $num-1)."/>
        <var name="dir"/>
        <var name="fname" expr="\$.filename."/>
        <form id="menunews"><block cond="ingreso == 0">;
        $vxmlstr .= "" . $diario;
        $vxmlstr .= ". Titulares de ";
        $vxmlstr .= "" . $categ. ". ";
        $vxmlstr .= '<prompt>Escuche atentamente las instrucciones a
seguir. Si desea, avanzar a la siguiente noticia, presione uno. leer la Noticia
completa, presione dos. Repetir, presione 3. Retroceder, presione 4. Salir, presione
cero.</prompt><assign name="ingreso" expr="++ingreso"/></block>;
        $f = fopen($filename, "w");
        for ($i=1; $i <= $num; $i++)
        {
            //se agregan los titulos y descripciones
            $vxmlstr .= ' <block cond="ingreso == $.i.">
            <assign name="dir" expr="\$.listL[1][$.i]."/>
            <prompt>Noticia $.i.</prompt>
            <prompt>'.str_replace($search, $replace, $listT[1][$.i]).
            '. ' .str_replace($search,
$replace, $listD[1][$.i]).'</prompt></block>;
        }
        $vxmlstr .= '<field name="sel" type="digits?length=1">Si desea, Leer la
siguiente noticia, presione uno. leer Noticia completa, presione dos. Repetir,
presione 3. Retroceder, presione 4. Salir, presione cero.
<filled>

```

```

<if cond="ingreso >= total">
<goto next="#again"/>
<elseif cond="sel == 1"/>
<assign name="ingreso" expr="++ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 2"/>
<submit next="http://localhost/full3universo.php" namelist="dir fname"
enctype="multipart/form-data"/>
<goto nextitem="sel"/>
<elseif cond="sel == 3"/>
<goto next="#menunews"/>
<elseif cond="sel == 4 && ingreso > 1"/>
<assign name="ingreso" expr="--ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 0"/>
Saliendo del periodico.<exit/>
<else/>
No he entendido<goto nextitem="sel"/>
</if>
</filled>
</field>
</form>
<form id="again">
<field name="item" type="digits?length=1">
Si desea escuchar de nuevo los titulares , presione uno. Si desea salir del periodico,
presione dos.
</filled>
<if cond="item == 1">
<assign name="ingreso" expr="1"/><goto next="#menunews"/>
<elseif cond="item == 2"/>
Saliendo del periodico.<exit/>
<else/>
No he entendido<goto nextitem="item"/>
</if>
</filled>
</field>
</form>
</vxml>';
    fwrite($f,$vxmlstr);
    fflush($f);
    fclose($f);
    //se ejecuta el vxml creado
    write("EXEC Vxml file://".$filename);
    write("SET CONTEXT internos");
    write("SET EXTENSION 1");
    write("SET PRIORITY 1");
}
}
// clean up file handlers etc.

```

```
fflush($in);fclose($in);  
fflush($stdlog);fclose($stdlog);  
exit;  
?>
```

## Anexo 3

eluniverso.xml

```
<vxml xml:lang="es-ES" version="2.0">
<var name="ingreso" expr="0"/>
<var name="total" expr="55"/>
<var name="dir"/>
<var name="fname" expr="/tmp/1312672216.6rss.vxml"/>
<form id="menunews">
<block cond="ingreso == 0">
EL UNIVERSO. Titulares de Noticias del Ecuador y del mundo.
<prompt>
Escuche atentamente las instrucciones a seguir. Si desea, avanzar a la siguiente
noticia, presione uno. leer la Noticia completa, presione dos. Repetir, presione 3.
Retroceder, presione 4. Salir, presione cero.
</prompt>
<assign name="ingreso" expr="++ingreso"/>
</block>
<block cond="ingreso == 1">
<assign name="dir" expr="http://eluniverso.com/2011/08/06/1/1372/ecuador-costarica-pugnan-un-lugar-octavos.html"/>
<prompt>Noticia 1</prompt>
<prompt>
Ecuador se impone 20 ante Costa Rica en el Mundial Sub 20 en juego. La Tri
pasaria de ronda con una victoria ante los ticos y, con muchas menos opciones y
dependiendo de otros resultados, tambien con un empate como uno de las cuatro
mejores terceros.
</prompt>
</block>
<block cond="ingreso == 2">
<assign name="dir" expr="http://eluniverso.com/2011/08/06/1/1361/evo-morales-dice-dejaria-poder-momento-dependiera.html"/>
<prompt>Noticia 2</prompt>
<prompt>
Evo Morales dice que dejaria el poder en este momento si dependiera de el. El
mandatario de Bolivia dijo que la oposicion pierde su tiempo al tratar de
desgastarlo, porque el no tiene ambiciones personales, familiares, ni economicas.
</prompt>
</block>
<block cond="ingreso == 3">
<assign name="dir" expr="http://eluniverso.com/2011/08/06/1/1355/cynthia-viteri-presidente-correa-repite-solemnidad-cantinfladas-juridicas.html"/>
<prompt>Noticia 3</prompt>
<prompt>
Cynthia Viteri Presidente Correa repite con solemnidad cantinfladas juridicas .
Durante la cadena de este sabado, el presidente Rafael Correa hizo mofa de las
declaraciones de la asambleista por el movimiento Madera de Guerrero.
</prompt>
```

```

.....
</block>
<field name="sel" type="digits?length=1">
Si desea, Leer la siguiente noticia, presione uno. leer Noticia completa, presione
dos. Repetir, presione 3. Retroceder, presione 4. Salir, presione cero.
<filled>
<if cond="ingreso >= total">
<goto next="#again"/>
<elseif cond="sel == 1"/>
<assign name="ingreso" expr="++ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 2"/>
<submit next="http://localhost/full3universo.php" namelist="dir fname"
enctype="multipart/form-data"/>
<goto nextitem="sel"/>
<elseif cond="sel == 3"/>
<goto next="#menunews"/>
<elseif cond="sel == 4 && ingreso > 1"/>
<assign name="ingreso" expr="--ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 0"/>
Saliendo del periodico.
<exit/>
<else/>
No he entendido
<goto nextitem="sel"/>
</if>
</filled>
</field>
</form>
<form id="again">
<field name="item" type="digits?length=1">
Si desea escuchar de nuevo los titulares, presione uno. Si desea salir del periodico,
presione dos.
<filled>
<if cond="item == 1">
<assign name="ingreso" expr="1"/>
<goto next="#menunews"/>
<elseif cond="item == 2"/>
Saliendo del periodico.
<exit/>
<else/>
No he entendido
<goto nextitem="item"/>
</if>
</filled>
</field>
</form>
</vxml>

```



Retornando a los titulares.

```
<?php
  echo "<submit next=\"file://\".$fname.\"\" enctype=\"multipart/form-data\"/>";
?>
<else/>
No he entendido<goto nextitem="item"/>
</if>
</filled>
</field>
</form>
</vxml>
```

## Anexo 5

decode.agi

```
#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1;
use Asterisk::AGI;
my $AGI = new Asterisk::AGI;
%input = $AGI->ReadParse();
my $code;
my @splitted=();
my @buscar=();
my @replace=();
my $palabra;
my $i;
my $n=0;
my $tmp=0;
my $carac="";
my $tmp2=0;
my $m=0;
my $pleer;
my @leer=();
my @lectura=();
$AGI->stream_file("busqueda", "");
while ($carac ne '#' && $carac ne '*') {
    $carac="";
    $carac=$AGI->get_data('silence/5','5000','1');
    $AGI->verbose("este es el caracter".$carac."fin");
    if ($carac eq ""){
        $carac='0';
        $lastc=$carac;
        $AGI->stream_file("beep", "");
    }
    elsif ($carac ne $lastc && $lastc ne '0')
    {
        #si ingreso uno distinto al anterior ej: 77708
        $code=$code.'0';
        $lastc=$carac;
    }else{
        $lastc=$carac;
    }
    $code=$code.$carac;
}#end while

$AGI->verbose("Code: ".$code);
@splitted=split(/0+|#|\*/,$code);
```



```

        @buscar=("2","22","222",,"2222","3","33","333","3333","4","44","444","4444","
5","55","555","6","66","666","6666","66666","7","77","777","7777","8","88","888","888
8","9","99","999","9999");
        @replace=("A","B","C",,"%C1","D","E","F",,"%C9","G","H","I",,"%CD","J","K","L"
,"M","N","O",,"%D1",,"%D3","P","Q","R","S","T","U",,"%DA","V","W","X","Y","Z");
        @leer=(,"%C1",,"%C9",,"%CD",,"%D1",,"%D3",,"%DA");
        @lectura=("A","E","I","NI","O","U");
        for($i = 0 ; $i < @splitted ; $i++){
            $tmp=0;
            $n=0;
            while($n < @buscar && $tmp == 0 ){
                if($buscar[$n] eq $splitted[$i]){
                    $splitted[$i] =~ s/$buscar[$n]/$replace[$n]/;
                    $palabra = $palabra . $splitted[$i];
                    $m=0;
                    $tmp2=0;
                    while($m < @leer && $tmp2 == 0) {
                        if($splitted[$i] eq $leer[$m]) {
                            $splitted[$i] =~
s/$leer[$m]/$lectura[$m]/;
                            $pleer = $pleer . $splitted[$i];
                            $tmp2=1;
                        }
                        $m++;
                    }
                    if($tmp2 == 0){
                        $pleer = $pleer . $splitted[$i];
                    }
                    $tmp=1;
                }
                $n++;
            }
        }
    }
    $AGI->verbose("palabra a buscar ".$pleer);
    $AGI->stream_file("palabraing","");
    $AGI->exec('Festival', $pleer);
    $AGI->stream_file("confirmacion","");
    $conf=$AGI->get_data('silence/5','5000','1');
    if($conf eq '1'){
        $AGI->stream_file('esperexfavor','');
        $AGI->exec('AGI','diccionario.agi','.$palabra.','.$pleer);
    }
    else{
        $AGI->exec('AGI','decode.agi');
    }
}

```

## Anexo 6

diccionario.agi

```
#!/usr/bin/php -q
<?php
# don't let this script run for more than 60 seconds
set_time_limit(60);
# turn off output buffering
ob_implicit_flush(false);
# turn off error reporting, as it will most likely interfere with
# the AGI interface
#error_reporting(0);
$in = fopen("php://stdin","r");
$stdlog = fopen("/var/log/asterisk/acadi.log", "w");
# Definicion de funciones antes del programa principal
function read()
{
global $in, $debug;
$input = str_replace("\n", "", fgets($in, 4096));
return $input;
}
function errlog($line) {
global $err;
}
function write($line) {
global $debug;
echo $line."
";
}
while ($env=read())
{
$env = str_replace("\",\"", $env);
$s = split(" ", $env);
$agi[str_replace("agi_", "", $s[0])] = trim($s[1]);
if ($env == "")
{
break;
}
}
#PROGRAMA INICIAL
#variables desde dialplan
$palabra = $argv[1];
$pleer = $argv[2];
$url="http://buscon.rae.es/draeI/SrvltGUIBusUsual?origen=RAE&TIPO_BUS=3&LE
MA=";
$infxml=file_get_contents($url.$palabra);

#tomamos los datos ej: title, description.
```

```

$def=preg_match_all('/<span class="eAcep">([\^<>]+)</span>/', $infoxml, $listT);
$vxmlstr = '<?xml version="1.0" encoding="iso-8859-1"?>
    <vxml xml:lang="es-ES" version="2.0" xmlns="http://www.w3.org/2001/vxml">
        <var name="ingreso" expr="0"/>
        <var name="total" expr=".$def."/>
        <var name="dir"/>
        <form id="menunews">
            <block cond="ingreso == 0">La palabra a buscar es, '$pleer.'. Tiene, '$def.'
definiciones. Instrucciones a seguir. Presione uno, para avanzar a la siguiente
definicion. Repetir definicion, presione 3. Salir, presione cero.<assign
name="ingreso" expr="++ingreso"/></block>;
$search = array('á','é','í','ó','ú','Á','É','Í','Ó','Ú','Ñ','ñ',' ',' ',' ',' ',' ',' ',' ',' ');
$replace = array('a','e','i','o','u','A','E','I','O','U','Ni','ni');
if($def > 0){
    for($i=1; $i <= $def; $i++){
        $vxmlstr .= '<block cond="ingreso == '.$i.'">
            <prompt>Significado, '.$i.'</prompt>
            <prompt>'.str_replace($search, $replace, $listT[1][$i-1]).</prompt>
            </block>;
    }
}
else{
#no hay definiciones
$vxmlstr .= 'No existen definiciones.<goto next="#again"/>;
}
$vxmlstr .= '<field name="sel" type="digits?length=1">
Si desea continuar, presione uno. Repetir, presione 3. Salir, presione cero.
<filled>
<if cond="ingreso &gt;= total &amp;&amp; sel != 1">
<goto next="#again"/>
<elseif cond="sel == 1"/>
<assign name="ingreso" expr="++ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 0"/>
<goto next="#again"/>
<elseif cond="sel == 3"/>
<goto next="#menunews"/>
<else/>
No he entendido<goto nextitem="sel"/>
</if>
</filled>
</field>
</form>
<form id="again">
<field name="item" type="digits?length=1">
Fin de las definiciones. Si desea escuchar todo de nuevo, presione uno. Si desea
salir presione cero.
<filled>
<if cond="item == 1">

```

```
<assign name="ingreso" expr="1"/><goto next="#menunews"/>
<elseif cond="item == 0"/>
Saliendo del diccionario.<exit/>
<else/>
No he entendido<goto nextitem="item"/>
</if></filled></field></form></vxml>';
$fl = fopen("/tmp/dic.vxml","w");
fwrite($fl,$vxmlstr);
fclose($fl);
write("EXEC Vxml file:///tmp/dic.vxml");
write("SET CONTEXT internos");
write("SET EXTENSION 1");
write("SET PRIORITY 1");
// clean up file handlers etc.
fflush($in);fclose($in);
fflush($stdlog);fclose($stdlog);
exit;
?>
```

## Anexo 7

Diccionario.xml

```
<vxml xml:lang="es-ES" version="2.0">
<var name="ingreso" expr="0"/>
<var name="total" expr="6"/>
<var name="dir"/>
<form id="menunews">
<block cond="ingreso == 0">
La palabra a buscar es, TIERNO. Tiene, 6 definiciones. Instrucciones a seguir.
Presione uno, para avanzar a la siguiente definicion. Repetir definicion, presione 3.
Salir, presione cero.
<assign name="ingreso" expr="++ingreso"/>
</block>
<block cond="ingreso == 1">
<prompt>Significado, 1</prompt>
<prompt>
Que se deforma facilmente por la presion y es facil de romper o partir
</prompt>
</block>
<block cond="ingreso == 2">
<prompt>Significado, 2</prompt>
<prompt> Reciente, de poco tiempo</prompt>
</block>
<block cond="ingreso == 3">
<prompt>Significado, 3</prompt>
<prompt>
Se dice de la edad de la niniez, para explicar su delicadeza y docilidad
</prompt>
</block>
<block cond="ingreso == 4">
<prompt>Significado, 4</prompt>
<prompt> Propenso al llanto</prompt>
</block>
<block cond="ingreso == 5">
<prompt>Significado, 5</prompt>
<prompt> Afectuoso, carinoso y amable</prompt>
</block>
<block cond="ingreso == 6">
<prompt>Significado, 6</prompt>
<prompt>
Dicho de los ojos: Que tienen una fluxion ligera continua
</prompt>
</block>
<field name="sel" type="digits?length=1">
Si desea continuar, presione uno. Repetir, presione 3. Salir, presione cero.
</filled>
```

```
<if cond="ingreso >= total && sel != 1">  
<goto next="#again"/>  
<elseif cond="sel == 1"/>  
<assign name="ingreso" expr="++ingreso"/>  
<goto next="#menunews"/>  
<elseif cond="sel == 0"/>  
<goto next="#again"/>  
<elseif cond="sel == 3"/>  
<goto next="#menunews"/>  
<else/>
```

No he entendido

```
<goto nextitem="sel"/>
```

```
</if>
```

```
</filled>
```

```
</field>
```

```
</form>
```

```
<form id="again">
```

```
<field name="item" type="digits?length=1">
```

Fin de las definiciones. Si desea escuchar todo de nuevo, presione uno. Si desea salir presione cero.

```
</filled>
```

```
<if cond="item == 1">
```

```
<assign name="ingreso" expr="1"/>
```

```
<goto next="#menunews"/>
```

```
<elseif cond="item == 0"/>
```

Saliendo del diccionario.

```
<exit/>
```

```
<else/>
```

No he entendido

```
<goto nextitem="item"/>
```

```
</if>
```

```
</filled>
```

```
</field>
```

```
</form>
```

```
</vxml>
```

## Anexo 8

decodewiki.agi

```
#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1;
use Asterisk::AGI;
my $AGI = new Asterisk::AGI;
%input = $AGI->ReadParse();
my $code;
my @splitted=();
my @buscar=();
my @replace=();
my $palabra;
my $i;
my $n=0;
my $tmp=0;
my $carac="";
$AGI->stream_file("busquedawiki", "");
while ($carac ne '#' && $carac ne '*') {
    $carac="";
    #$AGI->stream_file("beep", "");
    $carac=$AGI->get_data('silence/5','5000','1');
    $AGI->verbose("este es el caracter".$carac."fin");
    if ($carac eq ""){
        $carac='0';
        $lastc=$carac;
        $AGI->stream_file("beep", "");
    }
    elsif ($carac ne $lastc && $lastc ne '0'){
        #si mete uno distinto al anterior ej: 77708
        $code=$code.'0';
        $lastc=$carac;
    }else{
        $lastc=$carac;
    }
    $code=$code.$carac;
}#end while
$AGI->verbose("Code: ".$code);
if ($code ne '0*'){
    @splitted=split(/0+|#|\/|$/, $code);
    @buscar=("2", "22", "222", "2222", "3", "33", "333", "3333", "4", "44", "444", "4444", "5", "55", "555", "6", "66", "666", "6666", "66666", "7", "77", "777", "7777", "8", "88", "888", "8888", "9", "99", "999", "9999");
    @replace=("A", "B", "C", "Á", "D", "E", "F", "É", "G", "H", "I", "Í", "J", "K", "L", "M", "N", "O", "Ñ", "Ó", "P", "Q", "R", "S", "T", "U", "V", "Ú", "W", "X", "Y", "Z");
    for($i = 0 ; $i < @splitted ; $i++){
```

```

        $tmp=0;
        $n=0;
        while($n < @buscar && $tmp == 0 ){
            if($buscar[$n] eq $splitted[$i]){
                $splitted[$i] =~
s/$buscar[$n]/$replace[$n]/;
                $palabra = $palabra . $splitted[$i];
                $tmp=1;
            }
            $n++;
        }
    }
}
else{
    $AGI->exec('AGI','decodewiki.agi');
}
$AGI->verbose("palabra a buscar ".$palabra);
$AGI->stream_file("palabraing","");
$AGI->exec('Festival',lc($palabra));
$AGI->stream_file("confirmacion","");
$conf=$AGI->get_data('silence/5','5000','1');
if($conf eq '1'){
    $AGI->stream_file('esperexfavor','');
    $AGI->exec('AGI','wiki.agi','.ucfirst(lc($palabra)));
}
else{
    $AGI->exec('AGI','decodewiki.agi');
}

```



## Anexo 9

wiki.agi

```
#!/usr/bin/php -q
<?php
# don't let this script run for more than 60 seconds
set_time_limit(60);
# turn off output buffering
ob_implicit_flush(false);
# turn off error reporting, as it will most likely interfere with
# the AGI interface
#error_reporting(0);
$in = fopen("php://stdin","r");
$stdlog = fopen("/var/log/asterisk/acadi.log", "w");
# Definicion de funciones antes del programa principal
function read(){
global $in, $debug;
$input = str_replace("\n", "", fgets($in, 4096));
return $input;
}
function errlog($line) {
global $err;
}
function write($line) {
global $debug;
echo $line."\n";
}
while ($env=read())
{
$env = str_replace("\",\"", $env);
$s = split(" ", $env);
$agi[str_replace("agi_", "", $s[0])] = trim($s[1]);
if ($env == "")
{
break;
}
}
$filename = "/tmp/wikiagi.vxml";
$f1 = fopen($filename, "w");
$palabra = $argv[1];
#$palabra = "Polo";
#$dir = $_GET[dir];
```



Si desea, leer el significado <value expr="ingreso"/>, presione uno. Siguiente significado, presione 2. Salir, presione cero.

```
<filled>
<if cond="ingreso == total">
<goto next="#again"/>
<elseif cond="sel == 1"/>
<block>si desea salir presione cero. espere por favor. </block><submit
next="http://localhost/full3wiki.php" namelist="dir fname" enctype="multipart/form-
data"/>
<goto nextitem="sel"/>
<elseif cond="sel == 2"/>
<assign name="ingreso" expr="++ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 0"/>
Saliendo de la wikipedia.<exit/>
<else/>
No he entendido<goto nextitem="sel"/>
</if>
</filled>
</field>
</form>';
}
elseif($nofound > 0 && $desambi == 0){
    $vxmlstr .= 'No se han encontrado referencias de la palabra <goto
next="#again"/></block></form>';
}
else{
    $vxmlstr .= '<prompt>Instrucciones. Si desea continuar presione uno. caso
contrario, presione cero.</prompt></block>
<field name="resp" type="digits?length=1">
<filled>
<if cond="resp == 0">
saliendo de la wikipedia.<exit/>
<elseif cond ="resp == 1"/>
<submit next="http://localhost/full3wiki.php" namelist="dir fname"
enctype="multipart/form-data"/>
<else/>
no he entendido.<goto nextitem="resp"/>
</if>
</filled>
</field>
</form>';
```

```

}
$vxmlstr .= '<form id="again">
<field name="item" type="digits?length=1">
Fin de los significados. Si desea escuchar de nuevo los significados, presione uno.
Si desea salir presione cero.
<filled>
<if cond="item == 1">
<assign name="ingreso" expr="1"/><goto next="#menunews"/>
<elseif cond="item == 0"/>
Saliendo de la wikipedia.<exit/>
<else/>
No he entendido<goto nextitem="item"/>
</if>
</filled>
</field>
</form>
</vxml>';
}
else{
    $vxmlstr = '<?xml version="1.0" encoding="iso-8859-1"?>
    <vxml xml:lang="es-ES" version="2.0"
xmlns="http://www.w3.org/2001/vxml">
<form><block><prompt>Definicion no disponible. Saliendo de la wikipedia.
<exit/></prompt></block></form></vxml>';
}
fclose($f);
fwrite($fl,$vxmlstr);
fflush($fl);
fclose($fl);
//se ejecuta el vxml creado
write("EXEC Vxml file://". $filename);
write("SET CONTEXT internos");
write("SET EXTENSION 1");
write("SET PRIORITY 1");
// clean up file handlers etc.
fflush($in);fclose($in);
fflush($stdlog);fclose($stdlog);
exit;
?>

```

## Anexo 10

Wiki1.xml

```
<vxml xml:lang="es-ES" version="2.0">
<var name="fname" expr="/tmp/wikiagi.vxml"/>
<var name="ingreso" expr="1"/>
<var name="dir" expr="http://es.wikipedia.org/wiki/Gorra"/>
<var name="total" expr="0"/>
<form id="menunews">
<block cond="ingreso == 1">
    Wikipedia. La enciclopedia libre.
</block>
<prompt>
Instrucciones. Si desea continuar presione uno. caso contrario, presione cero.
</prompt>
</block>
<field name="resp" type="digits?length=1">
<filled>
<if cond="resp == 0">
saliendo de la wikipedia.
</if>
<elseif cond="resp == 1">
<submit next="http://localhost/full3wiki.php" namelist="dir fname"
enctype="multipart/form-data"/>
</elseif>
no he entendido.
<goto nextitem="resp"/>
</if></filled></field></form>
<form id="again">
<field name="item" type="digits?length=1">
Fin de los significados. Si desea escuchar de nuevo los significados, presione uno.
Si desea salir presione cero.
<filled>
<if cond="item == 1">
<assign name="ingreso" expr="1"/>
<goto next="#menunews"/>
</if>
<elseif cond="item == 0">
Saliendo de la wikipedia.
</elseif>
No he entendido
<goto nextitem="item"/>
</if>
</filled>
</field>
</form>
</vxml>
```

## Anexo 11

full3wiki.php

```
<?php
header("Content-type: text/xml");
echo("<?xml version='1.0'?>\n");
?>
<vxml version="2.0">
<form id="news">
<block>
<?php
$dir = $_GET[dir];
$fname = $_GET[fname];
$f=fopen("/tmp/wkitmp.txt","w");
$infxml = file_get_contents($dir);
$num=preg_match_all('/<div id="bodyContent">([^\@]+[^\^]+)<!-- \tbodyContent --
>/', $infxml, $listT);
fwrite($f, "num= ". $num);
$text=preg_match_all('/(<p>(?:.+)</p>|<h.>(?:.+)</h.>|<li>(?:.+)</li>)/', $listT[1][0], $
bodyContent);
fwrite($f, "text= ".$text."\n");
//arreglos para reemplazar tildes y comillas
$search = array('á','é','í','ó','ú','Á','É','Í','Ó','Ú','Ñ','ñ','<br>',"&nbsp;","'",'"','!','-
',';','"',"'','(')');
$replace = array('a','e','i','o','u','A','E','I','O','U','Ni','ni',"");
if($infxml != ""){
    for($i=0; $i<$text ; $i++){
        if(substr($bodyContent[1][$i],0,2)!="<h"){
            $noti = str_replace($search, $replace,$bodyContent[1][$i]);
            $sintags=
preg_replace('/<sup[^\>]+>.+</sup>|<[^\>]*>|\\[[^\]]*\\|\\[^\A-Za-z0-
9.\\s]|cita.*requerida/', "", $noti);
            $resultado = preg_replace("/
((XC|XL|L?X{0,3})(IX|IV|V?I{0,3})|C|XL|V|I)([.\\s])/e", ""
'.romtoint("\\2', '\\3'). '\\4"', $sintags);
            echo ($resultado . " ");
        }
        else{
            $i=$text;
        }
    }
}
else{
echo "Definicion no encontrada.";
}
function romtoint($dec,$uni){
    $lookup = array('XC','XL','IX','IV','X','L','V','I','');

```

```

$rep = array('9 ','4 ','9 ','4 ','1 ','5 ','5 ','1 ','0 ');
if ($dec==" && $uni==" ) return "";
if ($dec != ""){
    $decrep = str_replace($lookup,$rep,$dec,$numrep);
    $expdecena = explode(" ",$decrep);
    $totaldec = 0;
    for ($i=0; $i<$numrep; $i++)
        $totaldec += (int)$expdecena[$i];
}else{
    $totaldec = 0;
}
if ($uni != ""){
    $decrep = str_replace($lookup,$rep,$uni,$numrep);
    $expdecena = explode(" ",$decrep);
    $totaluni = 0;
    for ($i=0; $i<$numrep; $i++)
        $totaluni += (int)$expdecena[$i];
}else
    $totaluni = 0;
return ($totaldec*10) + $totaluni;
}
fclose($fl);
?>

```

</block>

<field name="item" type="digits?length=1">

Fin del significado. Si desea escuchar de nuevo, presione uno. Si desea retornar a la wikipedia, presione cero.

<filled>

<if cond="item == 1">

<goto next="#news"/>

<elseif cond="item == 0"/>

retornando a la wikipedia.

<?php

echo "<submit next=\"file:///\".\$fname.\"\" enctype=\"multipart/form-data\"/>";

?>

<else/>

No he entendido<goto nextitem="item"/>

</if>

</filled>

</field>

</form>

</vxml>

## Anexo 12

Wiki2.xml

```
<vxml xml:lang="es-ES" version="2.0">
<var name="fname" expr="/tmp/wikiagi.vxml"/>
<var name="ingreso" expr="1"/>
<var name="dir" expr="http://es.wikipedia.org/wiki/Celular"/>
<var name="total" expr="0"/>
<form id="menunews">
<block cond="ingreso == 1">
    Wikipedia. La enciclopedia libre.
<assign name="total" expr="3"/>
La palabra Celular tiene 3 significados.
</block>
<block cond="ingreso == 1">
<assign name="dir" expr="http://es.wikipedia.org/wiki/C%C3%A9lula"/>
<prompt>Significado, 1</prompt>
<prompt>
celular, lo relativo a la célula, unidad estructural y funcional principal de los seres
vivos.
</prompt>
</block>
<block cond="ingreso == 2">
<assign name="dir"
expr="http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil"/>
<prompt>Significado, 2</prompt>
<prompt>
teléfono celular, un dispositivo electrónico para telecomunicaciones personales a
distancia con red inalámbrica.
</prompt>
</block>
<block cond="ingreso == 3">
<assign name="dir" expr="http://es.wikipedia.org/wiki/Cellular"/>
<prompt>Significado, 3</prompt>
<prompt>Celular, película estadounidense de 2004.</prompt>
</block>
<field name="sel" type="digits?length=1">
    Si desea, leer el significado
<value expr="ingreso"/>
, presione uno. Siguiente significado, presione 2. Salir, presione cero.
<filled>
<if cond="ingreso == total">
<goto next="#again"/>
<elseif cond="sel == 1"/>
<block>si desea salir presione cero. espere por favor. </block>
<submit next="http://localhost/full3wiki.php" namelist="dir fname"
enctype="multipart/form-data"/>
<goto nextitem="sel"/>
```



```

<elseif cond="sel == 2"/>
<assign name="ingreso" expr="++ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 0"/>
Saliendo de la wikipedia.
<exit/>
<else/>
No he entendido
<goto nextitem="sel"/>
</if>
</filled>
</field>
</form>
<form id="again">
<field name="item" type="digits?length=1">
Fin de los significados. Si desea escuchar de nuevo los significados, presione uno.
Si desea salir presione cero.
<filled>
<if cond="item == 1">
<assign name="ingreso" expr="1"/>
<goto next="#menunews"/>
<elseif cond="item == 0"/>
Saliendo de la wikipedia.
<exit/>
<else/>
No he entendido
<goto nextitem="item"/>
</if>
</filled>
</field>
</form>
</vxml>

```



```

(preg_match_all("/<description><![CDATA\[([.+]*)\]></description>|<description>(.+)</description>/",$infoxml,$listD)){
    #recogemos el uniqueid para luego crear
    #un .vxml temporal con ese nombre ej: /tmp/09232rss.vxml
    $id = $agi[uniqueid];
    $filename="/tmp/".$id.".rss.vxml";
    #SE CREA EL VXML
    preg_match_all("/<link>(.+)</link>/",$infoxml,$listL);
```

```

<goto next="#menunews"/>
<elseif cond="sel == 4 & & ingreso > 1"/>
<assign name="ingreso" expr="--ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 0"/>
Saliendo del blog.<exit/>
</if></filled></field></form>
<form id="again">
<field name="item" type="digits?length=1">
Fin de los articulos. Si desea escuchar todos los articulos de nuevo, presione uno.
Si desea salir presione dos.
</filled>
<if cond="item == 1">
<assign name="ingreso" expr="1"/><goto next="#menunews"/>
<elseif cond="item == 2"/>
Hasta Pronto<exit/>
</else/>
No he entendido<goto nextitem="item"/>
</if>
</filled>
</field>
</form>
</vxml>;
    fwrite($f,$vxmlstr);
    fflush($f);
    fclose($f);
    //se ejecuta el vxml creado
    write("EXEC Vxml file://".$filename);
    write("SET CONTEXT internos");
    write("SET EXTENSION 1");
    write("SET PRIORITY 1");
}
}
// clean up file handlers etc.
fflush($in);fclose($in);
fflush($stdlog);fclose($stdlog);
exit;
?>

```

## Anexo 14

Blog.xml

```
<vxml xml:lang="es-ES" version="2.0">
<var name="ingreso" expr="0"/>
<var name="total" expr="10"/>
<var name="dir"/>
<var name="fname" expr=""/tmp/1312859682.0rss.vxml"/>
<form id="menunews">
<block cond="ingreso == 0">
Bienvenido al Blog de Salud y nutricion. Lo que necesitas para estar sano.
<prompt>
Escuche atentamente las instrucciones a seguir. Leer siguiente articulo, presione
uno. Articulo completo, presione dos. Repetir el articulo, presione 3. Anterior
articulo, presione cuatro. Salir, presione cero.
</prompt>
<assign name="ingreso" expr="++ingreso"/>
</block>
<block cond="ingreso == 1">
<assign name="dir" expr="http://www.saludynutricion.es/2010/10/14/buscale-
nombre-a-la-epoc"/>
<prompt>Articulo 1</prompt>
<prompt>
Buscale nombre a la epoc. Queremos presentaros la iniciativa
buscalenombreaepoc.org campania lanzada en julio de este anio con el proposito
de combatir entre la poblacion el desconocimiento de la enfermedad y favorecer
una deteccion y un tratamiento precoz de la EPOC. Esta campania cuenta con el
soporte de todas las sociedades implicadas en el abordaje de la Enfermedad
Pulmonar Obstructiva .
</prompt>
</block>
<block cond="ingreso == 2">
<assign name="dir" expr="http://www.saludynutricion.es/2010/08/24/alimentos-sin-
lactosa"/>
<prompt>Articulo 2</prompt>
<prompt>
Alimentos sin lactosa. Aunque el ser humano mayoritariamente se haya
acostumbrado al consumo de leche parece ser que esto no fue siempre asi y
algunas personas muestran intolerancia o alergia a la lactosa o proteina de la leche.
Productos como Benecol cubren la demanda de estas personas que a veces
encuentran muchas dificultades para encontrar productos que no .
</prompt>
</block>
<block cond="ingreso == 3">
<assign name="dir" expr="http://www.saludynutricion.es/2010/07/29/ortoideas-
ideas-para-facilitar-tu-vida"/>
<prompt>Articulo 3</prompt>
```

```

<prompt>
Ortoideas ideas para facilitar tu vida. Desde tiendy nos avisan del estreno de
Ortoideas. Se trata de una tienda online de accesorios para personas con distinto
tipo de problemas de movilidad y otras minusvalias. Para los que no sufrimos
ninguno de estos problemas seguro que muchos de los articulos que nos proponen
nos sorprenderan por su originalidad y sencillez y a .
</prompt>
</block>
....
<field name="sel" type="digits?length=1">
Si desea leer siguiente articulo, presione uno. Articulo
<value expr="ingreso"/>
completo, presione dos. Repetir el articulo, presione 3. Anterior articulo, presione
cuatro. Salir, presione cero.
<filled>
<if cond="sel == 1">
<assign name="ingreso" expr="++ingreso"/>
<goto next="#menunews"/>
<elseif cond="ingreso == total"/>
<goto next="#again"/>
<elseif cond="sel == 2"/>
espere por favor.
<submit next="http://localhost/fullblogsahud.php" namelist="dir fname"
enctype="multipart/form-data"/>
<goto nextitem="sel"/>
<elseif cond="sel == 3"/>
<goto next="#menunews"/>
<elseif cond="sel == 4 && ingreso > 1"/>
<assign name="ingreso" expr="--ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 0"/>
Saliendo del blog.
<exit/>
</if>
</filled>
</field>
</form>
<form id="again">
<field name="item" type="digits?length=1">
Fin de los articulos. Si desea escuchar todos los articulos de nuevo, presione uno.
Si desea salir presione dos.
<filled>
<if cond="item == 1">
<assign name="ingreso" expr="1"/>
<goto next="#menunews"/>
<elseif cond="item == 2"/>
Hasta Pronto
<exit/>
<else/>

```

No he entendido  
<goto nextitem="item"/>  
</if>  
</filled>  
</field>  
</form>  
</vxml>





## Anexo 16

decodeLetra.agi

```
#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1;
use Asterisk::AGI;
my $AGI = new Asterisk::AGI;
%input = $AGI->ReadParse();
my $code;
my @splitted=();
my @buscar=();
my @replace=();
my $letra;
my $i;
my $n=0;
my $tmp=0;
my $carac="";
$AGI->stream_file("autorLetra", "");
while ($carac ne '#' && $carac ne '*') {
    $carac="";
    #$AGI->stream_file("beep", "");
    $carac=$AGI->get_data('silence/5','5000','1');
    $AGI->verbose("este es el caracter".$carac."fin");
    if ($carac eq ""){
        $carac='0';
        $lastc=$carac;
        $AGI->stream_file("beep", "");
    }
    elsif ($carac ne $lastc && $lastc ne '0')
    {
        $code=$code.'0';
        $lastc=$carac;
    }else{
        $lastc=$carac;
    }
    $code=$code.$carac;
}#end while
$AGI->verbose("Code: ".$code);
if ($code ne '0*')
{
    @splitted=split(/0+|#|\/|$/, $code);
    @buscar=("2","22","222","3","33","333","4","44","444","5","55","555","6","66","
666","7","77","777","7777","8","88","888","9","99","999","9999");
    @replace=("A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","
R","S","T","U","V","W","X","Y","Z");
    for($i = 0 ; $i < @splitted ; $i++){
```

```

$tmp=0;
$n=0;
while($n < @buscar && $tmp == 0 ){
    if($buscar[$n] eq $splitted[$i]){
        $splitted[$i] =~ s/$buscar[$n]/$replace[$n]/;
        $letra = $letra . $splitted[$i];
        $tmp=1;
    }
    $n++;
}
}
}
else{
    $AGI->exec('AGI','decodeLetra.agi');
}
$AGI->verbose("letra a buscar ".$letra);
$AGI->stream_file("letraingresada","");
$AGI->stream_file("letters/.lc($letra));
$AGI->stream_file("confirmacion","");
$conf=$AGI->get_data('silence/5','5000','1');
if($conf eq '1'){
    $AGI->stream_file('esperexfavor','');
    $AGI->exec('AGI','libro.agi',$letra);
}
else{
    $AGI->exec('AGI','decodeLetra.agi');
}
}

```

## Anexo 17

libro.agi

```
#!/usr/bin/php -q
<?php
# don't let this script run for more than 60 seconds
set_time_limit(60);
# turn off output buffering
ob_implicit_flush(false);
# turn off error reporting, as it will most likely interfere with
# the AGI interface
#error_reporting(0);
$in = fopen("php://stdin","r");
$stderr = fopen("/var/log/asterisk/acadi.log", "w");
# Definicion de funciones antes del programa principal
function read(){
    global $in, $stderr;
    $input = str_replace("\n", "", fgets($in, 4096));
    return $input;
}
function errlog($line) {
    global $stderr;
}
function write($line) {
    global $stderr;
    echo $line."n";
}
while ($env=read()){
    $env = str_replace("\n", "", $env);
    $s = split(":", $env);
    $agi[str_replace("agi_", "", $s[0])] = trim($s[1]);
    if ($env == ""){
        break;
    }
}
#PROGRAMA INICIAL
#variables desde dialplan
$f = fopen('/tmp/temporal2.txt','w');
$letra = $argv[1];
fwrite($f,$letra);
fclose($f);
$url="http://www.quieroleer.com.ar/ql.index.phtml?autores=";
$xml=file_get_contents($url.$letra);
$xml=utf8_encode($xml);
#tomamos los datos ej: title, description.
$def=preg_match_all('/<DIV><P><A HREF="([^\"]*)">([^\<]*)</A>/', $xml,$list);
$id = $agi[uniqueid];
$filename="/tmp/" . $id . ".lib.vxml";
```

```

$vxmlstr = '<?xml version="1.0" encoding="iso-8859-1"?>
  <vxml xml:lang="es-ES" version="2.0" xmlns="http://www.w3.org/2001/vxml">
    <var name="ingreso" expr="0"/>
    <var name="total" expr=".$def."/>
    <var name="dir"/>
    <var name="fname" expr="\.$filename."/>
    <form id="menunews">
      <block cond="ingreso == 0">Se encontraron, '.$def.' libros.</block>;
$search = array('á','é','í','ó','ú','Á','É','Í','Ó','Ú','Ñ','ñ','-','"','"','"','"','"','"','"','"');
$replace = array('a','e','i','o','u','A','E','I','O','U','Ni','ni','');
if($def > 0){
  $vxmlstr .= '<block cond="ingreso == 0">Instrucciones a seguir. Si desea,
leer el Libro, presione uno. avanzar el siguiente libro, presione 2. Repetir, presione
3. Retroceder, presione 4. Salir, presione cero.<assign name="ingreso"
expr="++ingreso"/></block>;
  for($i=1; $i <= $def; $i++){
    $vxmlstr .= '<block cond="ingreso == '.$(i)."'>
    <assign name="dir" expr="\.$listT[1][$(i-1)]."/>
    <prompt>Libro, '.$(i).'/</prompt>
    <prompt>'.str_replace($search, $replace,$listT[2][$(i-1)].'/</prompt>
    </block>;
    fwrite($f,"dentro de for valor de i=".$i);
  }
  $f=fopen("/tmp/lili.txt","w");
  fwrite($f,"valor de i=".$i);
  fwrite($f,"total=".$def);
}
else{
#no hay libros
  $vxmlstr .= '<block><goto next="#again"/></block>;
}
$vxmlstr .= '<field name="sel" type="digits?length=1">
Si desea, leer el Libro, presione uno. Avanzar el siguiente libro, presione 2. Repetir,
presione 3. Retroceder, presione 4. Salir, presione cero.
<filled>
<if cond="ingreso &gt;= total &amp;&amp; sel != 1">
<goto next="#again"/>
<elseif cond="sel == 1"/>
<submit next="http://localhost/full3libro.php" namelist="dir fname"
enctype="multipart/form-data"/>
<exit/>
<goto nextitem="sel"/>
<elseif cond="sel == 2"/>
<assign name="ingreso" expr="++ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 3"/>
<goto next="#menunews"/>
<elseif cond="sel == 4 &amp;&amp; ingreso &gt; 1"/>
<assign name="ingreso" expr="--ingreso"/>

```

```

<goto next="#menunews"/>
<elseif cond="sel == 0"/>
Saliendo de la biblioteca virtual.<exit/>
<else/>
No he entendido<goto nextitem="sel"/>
</if>
</filled>
</field>
</form>
<form id="again">
<field name="item" type="digits?length=1">
Fin de la lista de libros. Si desea escuchar de nuevo, presione uno. Si desea salir
presione cero.
</filled>
<if cond="item == 1">
<assign name="ingreso" expr="1"/><goto next="#menunews"/>
<elseif cond="item == 0"/>
Saliendo de la biblioteca virtual.<exit/>
<else/>
No he entendido<goto nextitem="item"/>
</if>
</filled>
</field>
</form></vxml>;
$fl = fopen($filename,"w");
fwrite($fl,$vxmlstr);
fclose($fl);
write("EXEC Vxml file://". $filename);
write("SET CONTEXT internos");
write("SET EXTENSION 1");
write("SET PRIORITY 1");
// clean up file handlers etc.
fflush($in);fclose($in);
fflush($stdlog);fclose($stdlog);
exit;
?>

```

## Anexo 18

Libro.xml

```
<vxml xml:lang="es-ES" version="2.0">
<var name="ingreso" expr="0"/>
<var name="total" expr="6"/>
<var name="dir"/>
<var name="fname" expr="/tmp/1312675651.16lib.vxml"/>
<form id="menunews">
<block cond="ingreso == 0">Se encontraron, 6 libros.</block>
<block cond="ingreso == 0">
Instrucciones a seguir. Si desea, leer el Libro, presione uno. avanzar el siguiente
libro, presione 2. Repetir, presione 3. Retroceder, presione 4. Salir, presione cero.
<assign name="ingreso" expr="++ingreso"/>
</block>
<block cond="ingreso == 1">
<assign name="dir" expr="libros/harry4"/>
<prompt>Libro, 1</prompt>
<prompt>Rowling, JK , 4Harry Potter y el caliz de fuego</prompt>
</block>
<block cond="ingreso == 2">
<assign name="dir" expr="libros/harry5"/>
<prompt>Libro, 2</prompt>
<prompt>Rowling, JK , 5Harry Potter y la Orden del Fenix</prompt>
</block>
<block cond="ingreso == 3">
<assign name="dir" expr="libros/harry6"/>
<prompt>Libro, 3</prompt>
<prompt>
Rowling, JK , 6Harry Potter y el misterio del principe
</prompt>
</block>
<block cond="ingreso == 4">
<assign name="dir" expr="libros/harry3"/>
<prompt>Libro, 4</prompt>
<prompt>
Rowling, JK , 3Harry Potter y el prisionero de Azkaban
</prompt>
</block>
<block cond="ingreso == 5">
<assign name="dir" expr="libros/harry2"/>
<prompt>Libro, 5</prompt>
<prompt>Rowling, JK , 2Harry Potter y la Camara Secreta</prompt>
</block>
<block cond="ingreso == 6">
<assign name="dir" expr="libros/harry1"/>
<prompt>Libro, 6</prompt>
<prompt>Rowling, JK , 1Harry Potter y la piedra filosofal</prompt>
```

```

</block>
<field name="sel" type="digits?length=1">
Si desea, leer el Libro, presione uno. Avanzar el siguiente libro, presione 2. Repetir,
presione 3. Retroceder, presione 4. Salir, presione cero.
<filled>
<if cond="ingreso >= total && sel != 1">
<goto next="#again"/>
<elseif cond="sel == 1"/>
<submit next="http://localhost/full3libro.php" namelist="dir fname"
enctype="multipart/form-data"/>
<exit/>
<goto nextitem="sel"/>
<elseif cond="sel == 2"/>
<assign name="ingreso" expr="++ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 3"/>
<goto next="#menunews"/>
<elseif cond="sel == 4 && ingreso > 1"/>
<assign name="ingreso" expr="--ingreso"/>
<goto next="#menunews"/>
<elseif cond="sel == 0"/>
Saliendo de la biblioteca virtual.
<exit/>
</else/>
No he entendido
<goto nextitem="sel"/>
</if>
</filled>
</field>
</form>
<form id="again">
<field name="item" type="digits?length=1">
Fin de la lista de libros. Si desea escuchar de nuevo, presione uno. Si desea salir
presione cero.
<filled>
<if cond="item == 1">
<assign name="ingreso" expr="1"/>
<goto next="#menunews"/>
<elseif cond="item == 0"/>
Saliendo de la biblioteca virtual.
<exit/>
</else/>
No he entendido
<goto nextitem="item"/>
</if></filled></field></form></vxml>

```

## Anexo 19

full3libro.php

```
<?php
header("Content-type: text/xml");
echo("<?xml version=\`1.0\`?>\n");
?>
<vxml version="2.0">
<var name="npag"/>
<var name="inicio" expr="http://localhost/full3libro.php"/>
<var name="raiz"/>
<var name="answt"/>
<form id="news">
<block>
<?php
$dir = $_GET[dir];
$fname = $_GET[fname];
$raiz = "http://www.quieroleer.com.ar/" . $dir;
$infoxml = file_get_contents($raiz);
$infoxml=utf8_encode($infoxml);
$f=fopen("/tmp/paglib1.txt","w");
#tomamos los datos
$num=preg_match_all('/<H.>(.*)</H.><p>([^\r]*)<BR>/', $infoxml, $listT);
$npag=preg_match_all('/<A [^\r]*>/', $infoxml, $listP)-1;
#arreglos para reemplazar tildes y comillas
$search =
array('á', 'é', 'í', 'ó', 'ú', 'Á', 'É', 'Í', 'Ó', 'Ú', 'Ñ', 'ñ', '<BR>', '\n', '","', '!', ':', ';', '()', '&nbsp;');
$replace = array('a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U', 'Ni', 'ni', ' ', ' ');
$titulo = str_replace($search, $replace, $listT[1][0]);
$breve = str_replace($search, $replace, $listT[2][0]);
$todo = $titulo . " . " . $breve;
fwrite($f, $dir);
echo $todo;
echo "<assign name=\`npag\` expr=\`\". $npag. "\`"/>";
echo "<assign name=\`raiz\` expr=\`\". $raiz. "\`"/>";
?>
</block>
<field name="answ" type="number">
Este libro tiene <value expr="npag"/> paginas. Por favor, digite el numero de pagina
que desea leer, o presione cero para volver a los libros.
<audio src="uri:descending-2tone.wav">
accediendo, por favor espere
</audio>
<filled>
<!-- vuelve a listar los libros-->
<if cond="answ == 0">
<?php
echo "<submit next=\`file://\". $fname. "\` enctype=\`multipart/form-data\`"/>";
```



```

?>
<elseif cond="answ &gt; npag"/>
No existe esa pagina<goto nextitem="answ"/>
<elseif cond="answ &lt;= npag"/>
<assign name="answt" expr="answ"/>
<goto next="#subd"/>
<else/>
No he entendido<goto nextitem="answ"/>
</if>
</filled>
</field>
</form>
<form id="subd">
<subdialog name="sub" src="http://localhost/full3pagina.php" namelist="npag raiz
inicio answt" enctype="multipart/form-data" method="post">
<filled>
<if cond="sub.sgt == 0">
<goto next="#news"/>
<elseif cond="sub.sgt == 100"/>
Saliendo del libro
<?php
    echo "<submit next=\"file:///\".$fname.\"\" enctype=\"multipart/form-data\"/>";
?>
<elseif cond="answt != sub.sgt"/>
<assign name="answt" expr="sub.sgt"/>
<goto next="#subd"/>
<else/>
<prompt>accediendo a la pagina</prompt>
</if>
</filled>
</subdialog>
</form>
</vxml>

```

## Anexo 20

full3pagina.php

```
<?php
header("Content-type: text/xml");
echo("<?xml version=\\"1.0\\"?>\n");
?>
<vxml version="2.0">
<var name="bandera"/>
<var name="npag"/>
<var name="raiz"/>
<var name="inicio"/>
<var name="cont"/>
<var name="sgt" expr="99"/>
<form id="news">
<block>
<?php
set_time_limit(15);
ob_implicit_flush(false);
$npag = $_POST[npag];
$inicio = $_POST[inicio];//vxml predecesor
$raiz = $_POST[raiz];//pagina web
$actual = $_POST[answt];
$fl=fopen("/tmp/pag.txt","w");
/*$npag = 300;
$inicio = 'full3libro.php';//vxml predecesor
$raiz = 'http://www.quieroleer.com.ar/libros/divina-comedia';//pagina web
$actual = 1;*/
$infxml = file_get_contents($raiz.$actual.".html");
$infxml = utf8_encode($infxml);
#tomamos los datos
$num = preg_match_all('/<TD
align="left">[<BR>]*([^\s]*)<VTR><VTABLE><VDIV>/', $infxml, $listT);
#arreglos para reemplazar tildes y comillas
$search = array('á','é','í','ó','ú','Á','É','Í','Ó','Ú','Ñ','ñ','&nbsp;');
$replace = array('a','e','i','o','u','A','E','I','O','U','Ni','ni');
$texto = str_replace($search, $replace, $listT[1][0]);
$sintags = preg_replace(array('/[.][.]+\s{0-9}<BR>/', '/\s\s+[0-9]+<BR>/', '/[\s\n]+\./', '/<[^<>]*>/', '/[^A-Za-z0-9,\n ]/,', '/\./', '/[ ]{2,}/'), array('1.;;;;;;', ';;;;', ' '), $texto);
$resultado = preg_replace("/ ((XC|XL|L?X{0,3})(IX|IV|V?I{0,3})|CX|LVI|I)([. \s])/e", "'romtoint('\2', '\3'). '\4'", $sintags);
function romtoint($dec, $uni){
    $lookup = array('XC','XL','IX','IV','X','L','V','I','');
    $rep = array('9','4','9','4','1','5','5','1','0');
    if ($dec==" " && $uni==" ") return " ";
    if ($dec!=" "){
        $decrep = str_replace($lookup, $rep, $dec, $numrep);
```

```

        $expdecena = explode(" ",$decrep);
        $totaldec = 0;
        for ($i=0; $i<$numrep; $i++)
            $totaldec += (int)$expdecena[$i];
    }else{
        $totaldec = 0;
    }
    if ($uni != ""){
        $decrep = str_replace($lookup,$rep,$uni,$numrep);
        $expdecena = explode(" ",$decrep);
        $totaluni = 0;
        for ($i=0; $i<$numrep; $i++)
            $totaluni += (int)$expdecena[$i];
    }else
        $totaluni = 0;
    return ($totaldec*10) + $totaluni;
}
fwrite($fl,$resultado);
fclose($fl);
//bandera es la pagina actual
echo "<assign name='bandera' expr='\"".$actual."\"/>";
echo "<assign name='raiz' expr='\"'".$raiz."\"/>";
echo "<assign name='inicio' expr='\"'".$inicio."\"/>";
echo "<assign name='npag' expr='\"'".$npag."\"/>";
echo " Pagina ".$actual." ";
echo "Siga las siguientes instrucciones.";
if ($actual>=$npag) {
    echo "Para ir a la Pagina anterior, presione 1. Inicio, presione 2.";
}elseif ($actual==1){
    echo "Para ir a la Pagina siguiente, presione 3. Inicio, presione 2.";
}else{
    echo "Para ir a la Pagina anterior, presione 1. Inicio, presione 2. Pagina siguiente,
presione 3.";
}
echo " Salir, presione cero. ";
if($resultado != "TD"){
    echo $resultado;
}
else{
    echo " Esta pagina no tiene contenido.";
}
?>
</block>
<field name="item" type="number">
<?php
if ($actual>=$npag) {
    echo "Fin del libro. Pagina anterior, presione 1. Inicio, presione 2.";
}elseif ($actual==1){
    echo "Para ir a la Pagina siguiente, presione 3. Inicio, presione 2.";
}

```

```

}else{
    echo "Para ir a la Pagina anterior, presione 1. Inicio, presione 2. Pagina siguiente,
presione 3.";
}
echo " Salir, presione cero. ";
?>
<filled>
<if cond="item == 2">
<assign name="sgt" expr="0"/>
<return namelist="sgt"/>
<elseif cond="item == 1 && bandera != 1"/>
<assign name="sgt" expr="bandera-1"/>
<return namelist="sgt"/>
<elseif cond="item == 3 && bandera != npag"/>
<assign name="sgt" expr="bandera+1"/>
<return namelist="sgt"/>
<elseif cond="item == 0"/>
<assign name="sgt" expr="100"/>
<return namelist="sgt"/>
<else/>
No he entendido<goto nextitem="item"/>
</if>
</filled>
</field>
</form>
</vxml>

```

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Freedom Scientific, Fabricante mundial de productos tecnológicos para personas ciegas o con baja visión, <http://www.freedomscientific.com/about/about.asp>, fecha de consulta julio 2011
- [2] Blog Aguja de Marear, Encuesta sobre discapacidad en Chile en Abril 28 del 2005, <http://usuarios.discapnet.es/ajimenez/aguja/2005/04/encuesta-sobre-discapacidad-en-chile.html>, fecha de consulta julio 2011
- [3] Espacio Logopédico, El Braille: mucho más que un sistema de lectura para ciegos (parte II), [http://www.espaciologopedico.com/articulos/articulos2.php?Id\\_articulo=436](http://www.espaciologopedico.com/articulos/articulos2.php?Id_articulo=436), fecha de consulta julio 2011
- [4] Van Meggelen Jim, Madsen Leif, Smith Jared , Asterisk: The Future of Telephony, O'Reilly, fecha de consulta junio 2011
- [5] 3CX, Significado de los términos FXS y FXO, <http://www.3cx.com/PBX/FXS-FXO.html>, fecha de consulta julio 2011
- [6] I6net\*, VXi\* VoiceXML Browser, <http://www.i6net.com/products/vxi/>, fecha de consulta julio 2011
- [7] I6net\*, Manual de Navegador VXi\*, <http://www.i6net.com/support/documents/>, fecha de consulta julio 2011

- [8] W3C, Manual sobre Voice Extensible Markup Language Version 2.0, <http://www.w3.org/TR/voicexml20/#dml5.2.4>, fecha de consulta Julio 2011
- [9] Wikipedia, Festival TTS, [http://es.wikipedia.org/wiki/Festival\\_\(TTS\)](http://es.wikipedia.org/wiki/Festival_(TTS)), fecha de consulta julio 2011
- [10] Voip-info.org, Descripción de Sphinx, <http://www.voip-info.org/wiki/view/Sphinx>, fecha de consulta julio 2011
- [11] Asterisk, Descargas, <http://www.asterisk.org/downloads>, fecha de consulta julio 2011
- [12] Hess Joey, Alien (conversor de paquetes Linux), <http://kitenet.net/~joey/code/alien/>, fecha de consulta julio 2011
- [13] VozToVoice, Mejorando las voces de Festival para Asterisk en CentOS, <http://voztovoice.org/?q=node/213>, fecha de consulta julio 2011
- [14] Voip-info.org, Instalación de Asterisk-Flite, <http://www.voip-info.org/wiki/view/Asterisk+Flite>, fecha de consulta julio 2011
- [15] Voip-info.org, Integración Asterisk con Festival, <http://www.voip-info.org/wiki/view/Asterisk+festival+installation>, fecha de consulta julio 2011

[16] Speech Software at CMU, Flite (Festival-light),  
<http://www.speech.cs.cmu.edu/flite/>, fecha de consulta julio 2011