

**UNIVERSIDAD FRANCISCO GAVIDIA**



**“DESARROLLO DE UN SISTEMA DE INFORMACION BASADO EN VOICEXML  
PARA LA UNIVERSIDAD FRANCISCO GAVIDIA”**

**TRABAJO DE GRADUACIÓN PRESENTADO POR**

CARLOS ALFREDO MORALES GOMEZ  
JOSE RAUL PINEDA LEMUS  
MARIO RAFAEL RUIZ VARGAS

**PREVIA OPCIÓN AL TITULO**

MAESTRIA EN INFORMATICA APLICADA EN REDES

JUNIO DE 2007

SAN SALVADOR, EL SALVADOR, CENTRO AMERICA.

## INDICE

CAPITULO I .....	1
A. GENERALIDADES .....	1
B. JUSTIFICACIÓN .....	4
C. OBJETIVOS .....	5
1. GENERAL .....	5
2. ESPECIFICOS .....	5
D. ALCANCES .....	6
E. LIMITACIONES .....	7
F. PLANTEAMIENTO DEL PROBLEMA.....	8
RESULTADOS ESPERADOS AL TERMINAR EL PROYECTO.....	8
CAPITULO II .....	9
A. MARCO TEORICO: VoiceXML (VXML).....	9
VOICE EXTENSIBLE MARKUP LANGUAGE (VOICEXML) VERSION 2.0 .....	9
B. MODELO ARQUITECTONICO .....	9
1. Metas del VoiceXML.....	10
2. Alcance de VoiceXML.....	10
3. Principios de diseño de VoiceXML .....	11
4. Requerimientos para la Implementación de la Plataforma.....	12
C. CONCEPTOS .....	13
1. Diálogos y Subdiálogos .....	13
2. Sesiones .....	13
3. Aplicaciones.....	14
4. Gramática .....	14
5. Eventos.....	14
6. Enlaces.....	15
7. Elementos del VOICEXML.....	15
8. Estructura y ejecución del documento .....	17
D. CONSTRUCCIONES DE DIÁLOGO .....	24
1. Formas o Formularios.....	24
2. Elementos de los Formularios .....	25
3. Menús.....	28
CAPITULO III .....	34
A. PROPUESTA DE SOLUCION. ....	34
B. ARQUITECTURA DEL SISTEMA PROPUESTO.....	34
1. Arquitectura del servicio.....	37
C. ESQUEMA DE SOLUCION DE LA PROPUESTA.....	41
1. Definición de requerimientos de la aplicación.....	41
2. Diseño del producto .....	43
3. Construcción .....	70
4. Instalación.....	73
BIBLIOGRAFÍA .....	79
ANEXOS .....	80
ANEXO: A1. Manual del Administrador .....	81
ANEXO: A2. Manual del Usuario .....	93

# CAPITULO I

## A. GENERALIDADES

Para nadie es una novedad que la creación de las aplicaciones Web ha cambiado la forma de hacer negocios en todo el mundo, pero para lograr este gran avance ha sido necesario (y aún lo es) crear nuevas tecnologías, pero al inicio de esta tendencia la creación de nuevas tecnologías surgían en desorden, pues cada empresa creaba una tecnología que era incompatible con las de otras empresas, por esta razón surgieron varias organizaciones que se encargan de crear especificaciones estándares (guías de desarrollo) que las tecnologías deben acatar.

Entre este tipo de organizaciones destaca la World Wide Web Consortium (W3C por sus siglas en inglés). Esta organización tiene entre otros meritos la creación de las siguientes especificaciones:

- HTTP
- HTML
- XHTML
- CSS
- DOM y
- Otras más

Todas las especificaciones mencionadas están relacionadas directamente con el desarrollo Web.

Pero desde varios años atrás existe una tendencia que cada vez esta atrayendo más la atención de empresas, esta es el desarrollo de aplicaciones "voice enabled", es decir aplicaciones que en lugar de recibir y entregar información al usuario a través de interfaces graficas las envía y recibe a través interfaces basadas en voz, normalmente usando un teléfono cualquiera como dispositivo de entrada y salida. Esta tendencia tiene cada vez más fuerza debido al gran auge de los teléfonos móviles.

Para desarrollar este tipo de aplicaciones la W3C esta creando los siguientes estándares:

- VoiceXML
- Speech Recognition Grammar Specification (SRGS)
- Speech Synthesis Markup Language (SSML)
- Semantic Interpretation
- Call Control XML (CCXML)
- Entre otros.

Todas las especificaciones mencionadas forman el Speech Interfase Framework el cual que permite que estas especificaciones interactúen entre si y puedan ser utilizadas en aplicaciones Web mediante las cuales se generan dinámicamente documentos "voice

enabled", esto es posible debido a que todas las especificaciones están basadas en XML.

Mediante el uso de las tecnologías de voz se pueden desarrollar un sin número de aplicaciones que tendrán una característica en común: Poder acceder a información desde cualquier lugar. Esto se basa en la premisa de que en el mundo hay mucho más gente con teléfonos (incluyendo celulares) que gente con computadoras con acceso a Internet.

Entre los tipos de aplicaciones más conocidos se encuentran:

➤ Interactive Voice Response (IVR)

IVR es cualquier sistema integrado mediante el cual el usuario puede realizar transacciones a través de un teléfono. Por ejemplo:

- Un banco puede implementar un sistema mediante el cual el usuario puede transferir dinero entre sus cuentas, entre cuentas de diversos bancos, etc.
- Un usuario puede registrar la información diaria de su trabajo en un lugar alejado de un país, de esa forma en la oficina central de la empresa en la que trabaja pueden hacer un seguimiento del trabajo del empleado.
- Un Call Center puede ser automatizado en casi todas las opciones repetitivas (como transferir la llamada a un anexo, dejar un mensaje, etc.) de tal forma que el usuario pueda contactarse con un operador para operaciones personalizadas.

➤ Voice Mails

- Mediante este tipo de aplicaciones es posible que un usuario pueda escuchar, los mensajes de correo electrónico que ha recibido mientras esta conduciendo a su trabajo.

Una de las principales ventajas de usar esta tecnología es que permite realizar aplicaciones con un enfoque de apoyo social, pues en los lugares donde no hay Internet es seguro que al menos hay un teléfono cerca y mediante ese teléfono se puede registrar información de enfermedades, desastres, robos, cualquier otro tipo de eventos que sucedan por esos lugares y con esa información en tiempo real se pueden tomar muchas decisiones para el bien de la gente que reside en esos lugares alejados.

Estos son solo dos simples ejemplos de lo que se puede desarrollar con este tipo de tecnologías, pero cada día se abren un abanico de posibilidades que permitirán acceder de una manera fácil y natural a la información desde virtualmente cualquier lugar  
VoiceXML es una especificación propuesta por la W3C<sup>1</sup> que tiene como objetivo crear archivos XML, llamados documentos, que puedan reproducir sonido digitalizado, sonido

---

<sup>1</sup> World Wide Web Consortium.

sintetizado usando la tecnología TTS<sup>2</sup>, reconocer información ingresada por el usuario (tonos DTMF<sup>3</sup>) y reconocer palabra y/o frases pronunciadas por una persona, todo esto usando un dispositivo telefónico (teléfono clásico, celular o cualquier otra variante).

VoiceXML esta basado completamente en XML, es decir necesita que el documento VoiceXML sea "bien formado" para que pueda ser reconocido como correcto. Esto no ocurre con HTML, pero sí con XHTML. Actualmente esta especificación se encuentra en la versión 2.

Antes de entrar en discutir VoiceXML tenemos que entender como funciona VoiceXML, para ello revisemos la figura 1.

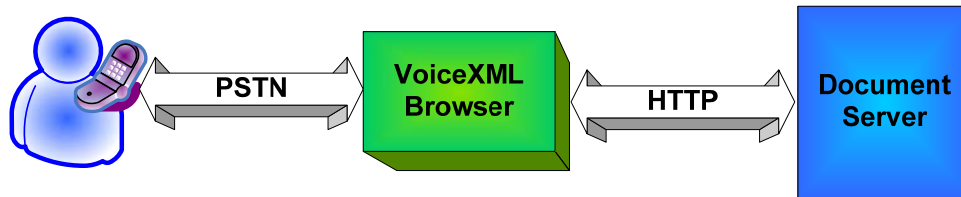


Figura 1. Estructura de un sistema VXML

De esta figura podemos obtener los siguientes enunciados.

- Mientras que HTML permite crear interfaces "gráficas" para que el usuario ingrese y reciba información, VoiceXML genera interfaces orales. Es decir, el usuario no "ve" la información, la escucha.
- Como VoiceXML no genera interfaces gráficas, el usuario no necesita una computadora (ni una PDA) solo le basta con un teléfono.
- El usuario se conecta al Browser a través la Red Publica de Telefonía (PSTN) o a través de una Red Telefónica IP (Voice over IP)
- VoiceXML, al igual que HTML, necesita de un browser para procesar la información, cada browser debe ser capaz de reconocer y procesar las etiquetas de cada lenguaje.
- Luego este Voice Browser solicita a un servidor Web (vía protocolo http) los documentos en formatos en vxml.

---

<sup>2</sup> Text to Speech

<sup>3</sup> Dual-tone multi-frequency.

## **B. JUSTIFICACIÓN**

Actualmente nos encontramos en la era de la Información, lo que implica que para desarrollar nuestras actividades cotidianas, debemos estar informados acerca de los sucesos que nos interesan, en tal sentido, con la utilización de tecnología vanguardista en este proyecto se plantea la utilización de VoiceXML para ofrecer alternativas de consulta de información.

En esa vía se desarrolla un prototipo para la utilización de VoiceXML en algunos servicios de la Universidad Francisco Gavidia, como: Consulta de calificaciones vía teléfono, Consulta de correo electrónico vía Teléfono, Envío de VoiceMail vía correo electrónicos y Notificación de nuevos correos.

En la actualidad los alumnos para obtener dicha información debe utilizar una computadora con acceso a Internet e ingresar al sitio de la universidad (<http://www.ufg.edu.sv>) sin embargo; si este quiere consultarlo desde algún lugar que no tiene acceso Internet, se ve imposibilitado a la obtención de la información y como una alternativa de solución a esta situación, se plantea la utilización de una aplicación de VoiceXML, con el cual el estudiante a través de la utilización de un teléfono ya sea este fijo o móvil y su número de carné, puede acceder a un menú que lo guía en los diferentes servicios que se encuentren activos conectándose así a las bases de datos de información de la Universidad satisfaciendo así sus necesidades de información.

El prototipo a presentar será una puerta para la implementación de nuevos servicios que serán de beneficio para toda la comunidad universitaria, es decir para sus estudiantes de pre-grado, post-grado, educación continua, personal docente y no docente de la institución; así mismo, esta solución se basa en software de bajo costo y cuya instalación no requiere de gran inversión en hardware, pues la Universidad Francisco Gavidia cuenta con una infraestructura tecnológica bien cimentada como para poder emprender proyectos similares.

Adicionalmente, este proyecto servirá de base para desarrollar futuros trabajos de investigación relacionados a la utilización de aplicaciones que involucren tecnología VoiceXML combinadas con herramientas de reconocimiento de voz, aportando así las líneas o referencias necesarias para echar a andar futuros trabajos de graduación tanto para estudiantes de las carreras de pre-grado como también para estudiantes de post-gradados.

## **C. OBJETIVOS**

### **1. GENERAL**

Desarrollar un sistema de información basada en VoiceXML para explotar el potencial tecnológico con el que cuenta la Universidad Francisco Gavidia.

### **2. ESPECIFICOS**

- Explotar la infraestructura tecnológica con la que cuenta la universidad para ofrecer novedosos servicios.
- Desarrollar una aplicación para brindar los servicios de: Consulta de calificaciones vía teléfono, Consulta de correo electrónico vía Teléfono, Envío de VoiceMail vía correo electrónicos y Notificación de nuevos correos.

## D. ALCANCES

- Tomar como modelo la Universidad Francisco Gavidia para el desarrollo de servicios a través de la tecnología VoiceXML
- Los servicios que se desarrollarán a través de la tecnología VoiceXML serán:
  - **Consulta de calificaciones vía teléfono**  
A través del servicio de VoiceXML todo usuario ingresará al sistema a través de cualquier teléfono (móvil o fijo) y podrá consultar su record académico del ciclo que esta cursando
  - **Consulta de correo electrónico vía Teléfono**  
El sistema podrá conectarse y leer el contenido de su buzón de correo electrónico, así mismo podrá responder, borrar y escuchar hora y fecha de un determinado mensaje.
  - **Envío de Voicemail vía correo electrónico.**  
Este servicio permitirá enviar a grupos de personas, correos electrónicos con archivos de audio adjunto.
  - **Notificación de nuevos correos**  
Este servicio de notificación de nuevos correos electrónicos el cual tiene dos opciones: a) Activar el envío de mensajes SMS y b) notificaciones de nuevos correos a su teléfono.



## E. LIMITACIONES

- No se dispone de un fondo financiero para compra de equipos de marca para el desarrollo de un sistema para producción.
- Los servicios serán desarrollados a manera de prototipo y no serán implementados, quedará a discreción de la institución si adopta esta tecnología.
- La universidad cuenta con diferentes estructuras de bases de datos (MySQL, SQL Server, PostgreSQL), lo que dificulta el desarrollo de la aplicación.
- Cuando el número DTMF (tono) correspondiente al nombre de usuario (en texto) se repita, se adicionará un carácter numérico al final de este valor. Dicho carácter numérico iniciará en 1, y si existen muchas coincidencias este valor de 1 se incrementará como tantas coincidencias existan.
- Para ser usuario registrado del sistema E-Voice, el usuario deberá tener una cuenta activa en el servidor de correo de la Universidad.
- Para el servicio de notificaciones de nuevos correos vía teléfono, si posee dos cuenta en el sistema E-Voice como docente y alumno, deberá colocar un números telefónico diferente para cada usuario.
- Debido al costo de una Tarjeta Telefónica y al poco tiempo para presentación de la Tesis, no se trabajará con OpenVXI 3.0 como interprete del Lenguaje VoiceXML, si no que se trabajará con VXIAsterisk.

## F. PLANTEAMIENTO DEL PROBLEMA

Para solventar los problemas de información de nuestros usuarios, se plantea la utilización de una nueva alternativa para acceder a algunos servicios que actualmente están siendo ofrecidos por la Universidad Francisco Gavidia a través de la página Web, para ello se plantea utilizar tecnología vanguardista como son las aplicaciones de reconocimiento de texto (text to speech) ya que la tendencia actual en el desarrollo de soluciones es la integración de servicios de INTERNET con la telefonía móvil o fija.

Es a través de la utilización de aplicaciones VoiceXML que los usuarios pueden acceder a la información sin la necesidad de contar con la conexión a INTERNET, ya que con la telefonía podrán resolver sus necesidades básicas de información.

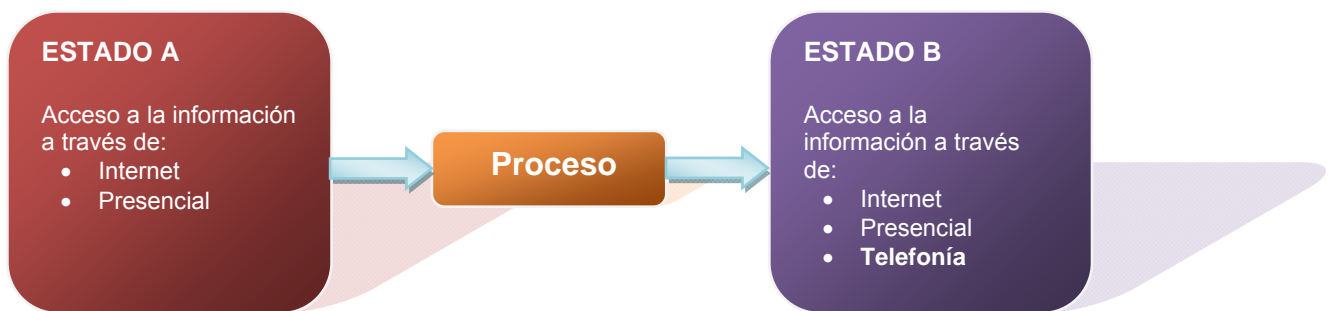


Figura 2. Método de la Caja Negra.

## RESULTADOS ESPERADOS AL TERMINAR EL PROYECTO

El desarrollo del prototipo dará paso a una nueva opción para que la universidad ofrezca los servicios de información a sus usuarios, permitiendo que la institución se posicione a nivel nacional con el uso de aplicaciones tecnológicas orientadas a los servicios educativos.

El prototipo será capaz de:

- Integrase con dos extensiones de la planta telefónica institucional.
- Contar con una interfaz Web para facilitar la administración de cuentas de usuario y asignación de privilegios, valiéndose de una adecuación de la base de datos que actualmente posee el Registro Académico de la Universidad.
- Autenticar al usuario con la base de datos de correo electrónico.
- Permitirle al usuario de la aplicación seleccionar cualquiera de los servicios:
  - Consulta de calificaciones vía teléfono
  - Consulta de correo electrónico vía teléfono
  - Envío de Voicemail vía correo electrónico.
  - Notificación de nuevos correos

## CAPITULO II

### A. MARCO TEORICO: VoiceXML (VXML)

#### VOICE EXTENSIBLE MARKUP LANGUAGE (VOICEXML) VERSION 2.0

### B. MODELO ARQUITECTONICO

El modelo arquitectónico asumido en este documento presenta los siguientes componentes:

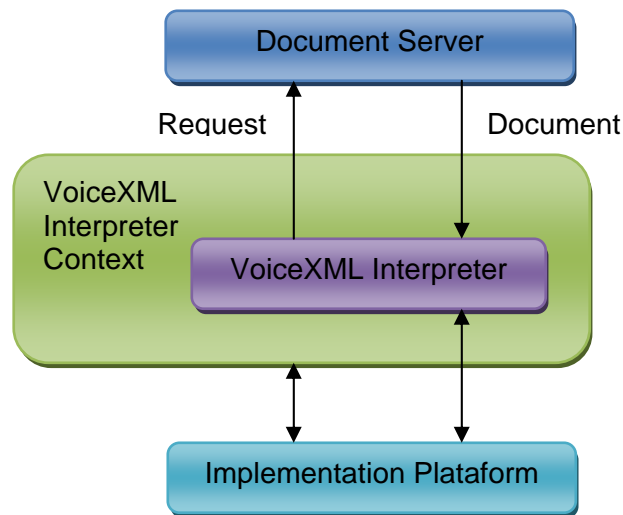


Figura 1: Componentes del VXML

Un Web Server procesa peticiones de una aplicación cliente, el Servidor produce documentos VoiceXML, los cuales son procesados por el intérprete VoiceXML. El contexto del intérprete de VoiceXML puede supervisar entradas del usuario en paralelo al intérprete de VoiceXML. Por ejemplo, un contexto del intérprete de VoiceXML puede esperar a escuchar siempre una frase especial de escape que lleve al usuario a una ayuda personal de alto nivel, y otra, puede esperar a escuchar las frases de escape que alteran preferencias del usuario como volumen o características text-to-speech.

La implementación de la plataforma es controlada por el intérprete de contextos VoiceXML. Por ejemplo, en una aplicación interactiva de respuesta por voz, el intérprete de VoiceXML puede ser responsable de detectar una llamada entrante, de adquirir el documento inicial de VoiceXML, y de contestar la llamada, mientras que el intérprete de VoiceXML conduciendo el diálogo después de su respuesta.

La implementación de la plataforma en práctica genera acontecimientos en respuesta a acciones del usuario y a acontecimientos del sistema (expiración del contador de tiempo). Algunos de estos acontecimientos son efectuados por el intérprete de VoiceXML en sí mismo, según lo especificado por el documento de VoiceXML, mientras que otros son efectuados sobre el contexto del intérprete de VoiceXML.

## 1. Metas del VoiceXML

La meta principal de VoiceXML consiste en traer de la Web y entregar su contenido a aplicaciones de respuesta por voz, y liberar a los autores de tales aplicaciones de la administración de los recursos de la programación de bajo nivel.

Además, permite la integración de los servicios de voz con servicios de datos usando el paradigma familiar de cliente-servidor. Un servicio de voz se ve como una secuencia de interacción de diálogos entre un usuario y la implementación de una plataforma.

Los diálogos son proporcionados por los servidores del documento (servidores Web), que pueden ser externos a la implementación de la plataforma. Los servidores de documentos mantienen la lógica total del servicio, realizan operaciones en las bases de datos, y producen diálogos.

Un documento de VoiceXML especifica cada interacción de diálogo que se conducirá por un intérprete de VoiceXML. La entrada del usuario afecta la interpretación del diálogo y se recoge en las peticiones realizadas a un servidor de documentos.

El servidor de documentos (Servidor Web) contesta con otro documento de VoiceXML para continuar la sesión del usuario con otros diálogos.

### **VoiceXML es un lenguaje que:**

- Reduce al mínimo las interacciones cliente-servidor especificando interacciones múltiples por documento.
- Separa código de la interacción del usuario (en VoiceXML) de la lógica del servicio (ej. escrituras del cgi).
- Promueve portabilidad del servicio a través de plataformas puestas en prácticas.
- Es fácil de utilizar para las interacciones simples, pero proporciona características de lenguaje para apoyar diálogos complejos.

Mientras que VoiceXML se esfuerza en acomodar los requisitos de una mayoría de servicios de respuesta por voz, los servicios con requisitos rigurosos también se pueden implementar realizando un nivel de control más fino.

## 2. Alcance de VoiceXML

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

- Salida de voz sintetizado (text-to-speech).
- Salida de archivos audio.
- Reconocimiento de voz.
- Reconocimiento de entrada DTMF.

- Grabación de entrada de voz.
- Control del flujo de diálogo.
- Características de la telefonía tales como transferencia y desconexión de la llamada.

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

### **3. Principios de diseño de VoiceXML**

- El lenguaje promueve la portabilidad de servicios con la abstracción de los recursos de la plataforma.
- El lenguaje acomoda diversidad de plataformas en formatos de audio, formatos de gramática, y esquemas de URI. Mientras que los fabricantes de plataformas pueden apoyarse a través de varios formatos gramaticales. El lenguaje requiere un formato común gramatical llamado XML sobre Especificaciones gramaticales de reconocimiento de voz (Speech Recognition Grammar Specification –SRGS) para facilitar la interoperabilidad. Similarmente, varios formatos de audio para lectura y la grabación pueden ser soportados.
- El lenguaje tiene una semántica bien definida. No se requiere La heurística del cliente para determinar la interpretación del elemento del documento.
- El lenguaje reconoce interpretaciones semánticas de gramáticas y pone esta información a disposición de la aplicación.
- El lenguaje tiene un mecanismo de control de flujo.
- No es pensado para uso intensivo de cómputo, operaciones con bases de datos, u operaciones de sistema. Éstos se asumen que serán manejados por los recursos fuera del intérprete del documento, ej. un servidor del documento.
- La lógica general del servicio, la administración del estado, la generación de diálogo, y la secuencia del diálogo se asume que residen fuera del intérprete del documento.
- El lenguaje proporciona maneras de enlazar documentos usando URIs, y también de enviar datos a los scripts del servidor utilizando URIs.
- VoiceXML proporciona las maneras de identificar exactamente que datos enviará al servidor, y que método del HTTP (GET o POST) a utilizar en el envío.

## 4. Requerimientos para la Implementación de la Plataforma

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

- **Adquisición de documento.** Se espera que el contexto del intérprete recupere los documentos para que el intérprete de VoiceXML actúe sobre él. El esquema de URI del "HTTP" debe ser soportado.  
En algunos casos, la petición del documento es generada por la interpretación de un documento de VoiceXML, mientras que otras peticiones son generadas por el contexto del intérprete, en respuesta a acontecimientos fuera del alcance del lenguaje, por ejemplo una llamada telefónica entrante. Al publicar peticiones del documento vía HTTP, el contexto del intérprete se identifica usando el "**User-Agent**" como variable "<name>/<version>", por ejemplo, "acme-browser/1.2".
- **Salida audio.** Una implementación de plataforma debe soportar salida de audio usando archivos audio y text-to-speech (TTS).  
La plataforma debe poder ordenar libremente TTS y la salida audio. Si un recurso de salida de audio no está disponible, un evento de **error.noresource** debe ser presentado. Los archivos de audio son referidos por un URI. El lenguaje especifica que un sistema de archivos de formatos audio es requerido
- **Entrada audio.** Una implementación de la plataforma es requerida para detectar y reportar el carácter y/o la entrada de voz simultáneamente, y para controlar la duración del intervalo de la detección de la entrada con un contador de tiempo cuya longitud es especificada por un documento de VoiceXML.  
Si un recurso de audio de la entrada no está disponible, un acontecimiento de **error.noresource** debe ser presentado.
  - Debe reflejar los caracteres (por ejemplo, DTMF) incorporados por un usuario. Las plataformas deben soportar la gramática de XML del DTMF descritas en la especificación del reconocimiento de voz de W3C [ SRGS ]. Deben también apoyar la forma aumentada de BNF (ABNF) de gramáticas de DTMF descritas en la especificación de la gramática del reconocimiento de voz de W3C [ SRGS ].
  - Debe ser capaz de recibir datos dinámicos de reconocimiento de voz. Debe poder utilizar datos de gramática de reconocimiento de voz en la forma de XML. Debe poder recibir datos de la gramática de reconocimiento de voz en la forma de ABNF, y debe soportar otros formatos tales como el formato de la gramática de JSpeech [ JSGF ]<sup>4</sup>o formatos propietarios. Algunos elementos de VoiceXML contienen datos de la gramática de voz; otros refieren a datos de la gramática de voz con un URI. El reconocedor de voz debe ser capaz de acomodar

---

<sup>4</sup> JSpeech Grammar Format

dinámicamente la entrada hablada la cual está escuchando con cualquier método de especificación de datos de gramática de voz .

- Debe poder grabar el audio recibido del usuario. La implementación de la plataforma debe poder colocar la grabación a disposición de la variable a solicitud de esta. El lenguaje especifica un conjunto de requerimientos de formatos de audio los que deben ser soportados por la plataforma.
- **Transferencia:** La plataforma deberá poder hacer una conexión a terceros a través de una red de comunicaciones, tal como el teléfono.

## C. CONCEPTOS

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

### 1. Diálogos y Subdiálogos

Hay dos clases de diálogos: formas (o formularios) y menús. Los formularios definen una interacción que recoge los valores para un sistema de variables. Cada campo puede especificar una sintaxis que define las entradas permisibles para ese campo. Si una sintaxis a nivel de forma está presente, puede ser utilizada para llenar varios campos a partir de una elocución.

Un menú se presenta al usuario con una selección de opciones y es entonces cuando las transiciones se dan a otro diálogo basado en dichas opciones.

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

### 2. Sesiones

Una sesión inicia cuando el usuario comienza a interactuar recíprocamente con un contexto del intérprete de VoiceXML. Continúa hasta que se cargan y se procesan los

documentos, y finaliza cuando lo es requerido por el usuario, un documento, o el contexto del intérprete.

### 3. Aplicaciones

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

Mientras que es cargado, las variables de la aplicación del documento raíz están disponibles para los otros documentos.

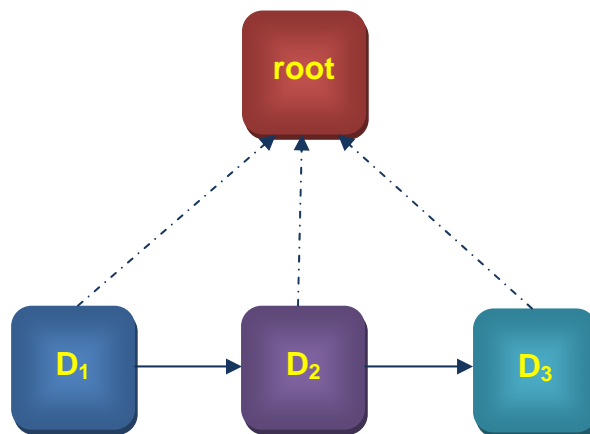


Figura 2: Transiciones entre documentos en una aplicación

### 4. Gramática

Cada diálogo tiene una o más gramáticas de voz y/o de DTMF asociados a él. En las aplicaciones dirigidas por la máquina, las gramáticas de cada diálogo están activas solamente cuando el usuario está en ese diálogo.

En aplicaciones mezcladas, donde el usuario y la máquina se alternan, los diálogos se señalan por medio de una bandera para hacer sus gramáticas activas (es decir, escuchado) incluso cuando el usuario están en otro diálogo dentro del mismo documento, o en otro documento cargado de la misma aplicación. En esta situación, si el usuario dice algo que enlaza las gramáticas activas de otro diálogo, se ejecutan transiciones a ese otro diálogo, con la elocución del usuario tratada como si fuera dicha en ese diálogo. Mezclando dicha iniciativa agrega flexibilidad y energía para aplicaciones de voz.

### 5. Eventos

VoiceXML proporciona un mecanismo de form-fill para manejar la entrada "normal" del usuario. Además, VoiceXML define un mecanismo para manejar los acontecimientos no cubiertos por el mecanismo form. Los acontecimientos son lanzados por la plataforma bajo una variedad de circunstancias, por ejemplo cuando no responde el usuario, no



responden de manera inteligible, peticiones de ayuda, etc. El intérprete también lanza acontecimientos si encuentra un error semántico en un documento de VoiceXML. Los acontecimientos son tomados por los elementos catch o por su taquigrafía sintáctica.

Cada elemento en el cual un acontecimiento puede ocurrir puede especificar elementos de captura (catch). Además, los elementos de captura también se heredan al incluir elementos "como si fueran por copia". En este sentido, el acontecimiento común que maneja el comportamiento de la variable de entorno se puede especificar en cualquier nivel, y se aplica en los niveles más bajos.

## 6. Enlaces

Un enlace (link), soporta iniciativas mezcladas. Este especifica una gramática activa siempre que el usuario esté en el alcance del enlace (link). Si el usuario ingresa coincidencias en el enlace de gramática, el control lo transfiere al URI destino. Un enlace o vínculo se puede utilizar para lanzar un evento ó para ir a un URI destino.

## 7. Elementos del VOICEXML

Elemento	Propósito.
<assign>	Asigna un valor a una variable
<audio>	Ejecuta un clip de audio
<block>	Un contenedor de código ejecutable (no interactivo).
<catch>	Captura un evento.
<choice>	Define un elemento de menú.
<clear>	Limpia una o más variables en el formulario.
<disconnect>	Desconecta una sesión.
<else>	Usado dentro de una sentencia <if>.
<elseif>	Usado dentro de la sentencia <if>.
<enumerate>	Término para enumerar la selección de opciones en un menú.
<error>	Captura un error de evento
<exit>	Salir de una sesión.
<field>	Declara una entrada de campo en un formulario.
<filled>	Una acción es ejecutada cuando los campos son llenados.
<form>	Un dialogo para presentar información y recoger datos.
<goto>	Ir a otro diálogo en el mismo o en otro documento.
<grammar>	Especifica un reconocimiento de voz o gramática DTMF.

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

Tabla 1: Elementos del VoiceXML.

## 8. Estructura y ejecución del documento

Un documento de VoiceXML se compone sobre todo de los elementos a nivel superior llamados *diálogos*. Hay dos tipos de diálogos: *formularios* y *menús*. Un documento puede también tener elementos <meta> y <metadata>, elementos <var> y <script>, elementos <property>, elementos <match> y elementos <link>.

### Ejecución dentro de un documento

La ejecución del documento comienza en el primer diálogo por defecto. Cada diálogo ejecutado, determina el siguiente diálogo. Cuando un diálogo no especifica un diálogo sucesor, la ejecución del documento finaliza.

A continuación se muestra un ejemplo: "Hello word!", para ilustrar algo de lo mencionado anteriormente.

Se hace que una variable de documento llamada "hi" lleve a cabo el saludo. Su valor se utiliza como prompt en primer formulario. Una vez que el primer formulario realiza el saludo, va al formulario nombrada "say\_goodbye", con el cual se despide del usuario con un "Goodbye!".

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd"
  version="2.0">
<meta name="author" content="John Doe"/>
<meta name="maintainer" content="hello-support@hi.example.com"/>
<var name="hi" expr="Hello World!"/>
<form>
  <block>
    <value expr="hi"/>
    <goto next="#say_goodbye"/>
  </block>
</form>
<form id="say_goodbye">
  <block>
    Goodbye!
  </block>
</form>
</vxml>
```

Alternativamente los formularios pueden ser combinados:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd"
  version="2.0">
<meta name="author" content="John Doe"/>
<meta name="maintainer" content="hello-support@hi.example.com"/>
<var name="hi" expr="Hello World!"/>
<form>
  <block>
    <value expr="hi"/> Goodbye!
  </block>
</form>
</vxml>
```

Los atributos del < vxml > incluidos:

<b>version</b>	La versión de VoiceXML de este documento (requerido). El número de versión actual es 2.0.
<b>xmlns</b>	El namespace designado para VoiceXML (requerido). El namespace para VoiceXML se define por <a href="http://www.w3.org/2001/vxml">http://www.w3.org/2001/vxml</a> .
<b>xml:base</b>	La base URI para este documento esta definido segun en [ <a href="#">Xml-base</a> ]. Como en [ <a href="#">HTML</a> ]
<b>xml:lang</b>	Identificador del lenguaje para este documento.
<b>Aplicación</b>	El URI de la aplicación raíz del documento, si lo hay.

Tabla 2: atributos del < vxml >

La información del lenguaje es heredada bajo de la jerarquía del documento: el valor de "xml:lang" se hereda por los elementos que también definen las propiedades de "xml:lang", tales como <grammar> y <prompt>, a menos que estos elementos especifiquen un valor alternativo.

## Ejecutar una aplicación Multi-Documento

Normalmente, cada documento funciona como una aplicación aislada. En los casos donde usted quisiera que los documentos múltiples trabajaran conjuntamente como una aplicación, usted selecciona un documento para que este sea *el documento raíz de la aplicación*, y el resto para ser los *documentos secundarios de la aplicación*. Cada documento de la hoja nombra el documento de la raíz en sus elementos <vxml>.

Cuando se hace esto, cada vez que se llama al intérprete para cargar y para ejecutar un documento de la hoja en esta aplicación, primero carga el documento raíz. El resto del documento de la aplicación es cargado hasta que el intérprete es llamado para cargar

un documento que pertenezca a las diferentes aplicaciones. Así una de las dos condiciones siguientes se realizan siempre durante la interpretación:

- Se carga el documento raíz de la aplicación y el usuario lo ejecuta: no hay hojas en el documento.
- Se carga el documento raíz de la aplicación y una solo hoja del documento y el usuario lo ejecuta en la hoja del documento.

Si hay una cadena de subdiálogos definidos en documentos separados, entonces puede haber más de una hoja de documento cargado aunque la ejecución estará solamente en uno de estos documentos.

Cuando una carga de la hoja del documento causa una carga del documento raíz, no se ejecuta ninguno de los diálogos en el documento raíz. La ejecución comienza en la hoja del documento.

Hay muchos beneficios para aplicaciones multi-documento.

- Las variables del documento raíz están disponibles para ser utilizadas por las hojas del documento, tal información puede ser compartida y conservada.
- Los elementos <property> del documento Raíz, especifican valores por defecto para las propiedades utilizadas en las hojas del documento.
- El código común de ECMAScript se puede definir en elementos <script> del documento raíz y utilizarlo en las hojas de los documentos.
- Los elementos <catch> del documento Raíz definen eventos por defecto manejados por las hojas de los documentos.
- El alcance de la sintaxis del documento raíz están activas cuando el usuario está en una hoja del documento, de modo que este pueda interactuar con los formularios, los links y menús del documento raíz.

Aquí está una aplicación de un doble-documento que ilustra esto:

### **Documento raíz de la aplicación** (app-root.vxml)

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd"
  version="2.0">
<var name="bye" expr="Ciao"/>
<link next="operator_xfer.vxml">
  <grammar type="application/srgs+xml" root="root" version="1.0">
    <rule id="root" scope="public">operator</rule>
  </grammar>
</link>
</vxml>
```

## Hoja del Documento (leaf.vxml)

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd"
  version="2.0" application="app-root.vxml">
<form id="say_goodbye">
<field name="answer">
  <grammar type="application/srgs+xml" src="/grammars/boolean.grxml"/>
  <prompt>Shall we say <value expr="application.bye"/>?</prompt>
  <filled>
    <if cond="answer">
      <exit/>
    </if>
    <clear namelist="answer"/>
  </filled>
</field>
</form>
</vxml>
```

En este ejemplo, la aplicación está diseñada para que leaf.vxml deba ser cargada primero. El atributo de la aplicación especifica que app-root.vxml debe ser utilizada como la aplicación raíz del documento. De tal forma que app-root.vxml es cargada, creando la variable de la aplicación bye y también define un vínculo (link) que navega sobre operador-xfer.vxml siempre que el usuario diga "operador". El usuario comienza en el formulario say\_goodbye:

C (computadora): ¿decir Ciao?

H (humano): Si.

C: No entendía lo que usted dijo. (Un mensaje plataforma-especifico)

C: ¿decimos Ciao?

H: Ciao.

C: No entendía lo que usted dijo.

H: Operator.

C: (va a operador\_xfer.vxml, que transfiere a un operador humano).

Observe que cuando el usuario está en una aplicación multi-documento, al menos dos documentos están cargados al mismo tiempo.

El elemento <vxml> de un documento raíz no tiene un atributo específico en la aplicación. El elemento <vxml> de una hoja del documento tiene un atributo específico en la aplicación. Un intérprete siempre tiene un documento raíz cargado; no siempre tiene cargada una hoja del documento.

El nombre actual del intérprete de la aplicación es la aplicación raíz del documento absoluto URI. El URI absoluto incluye una cadena de consulta, si está presente, sino ella no incluye un identificador del fragmento. El intérprete permanece en la misma aplicación mientras el nombre siga siendo el mismo.

## Subdiálogos

Un subdiálogo es un mecanismo para descomponer secuencias complejas de diálogos para mejorar las estructuras, o para crear componentes reutilizables. Por ejemplo, la solicitud de la información de la cuenta puede implicar el recopilar de varios pedazos de información, tales como número de cuenta, número de teléfono, etc. Un servicio de información de cuenta del cliente se pudo estructurar con varias aplicaciones independientes que podrían compartir este bloque básico, así sería razonable construirlo como subdiálogo. Esto se ilustra en el siguiente ejemplo. El primer documento, app.vxml, busca ajustar una cuenta de cliente, y al hacerlo debe conseguir la información de la cuenta. La información de la cuenta es obtenida usando un elemento del subdiálogo que invoque otro documento de VoiceXML para solicitar la entrada del usuario.

Mientras que se está ejecutando el segundo documento, el diálogo que llama al primero se suspende, esperando devolver la información. El segundo documento proporciona los resultados de las interacciones del usuario utilizando el elemento <return >, y los valores que resultantes se alcanzan con la variable definida por la propiedad conocida en el elemento del subdiálogo <subdialog>.

## Aplicación de Servicio al Cliente (app.vxml)

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd"
  version="2.0">
  <form id="billing_adjustment">
    <var name="account_number"/>
    <var name="home_phone"/>
    <subdialog name="accountinfo" src="acct_info.vxml#basic">
      <filled>
        <!-- Note the variable defined by "accountinfo" is
          returned as an ECMAScript object and it contains two
          properties defined by the variables specified in the
          "return" element of the subdialog. -->
```

```

    <assign name="account_number" expr="accountinfo.acctnum"/>
    <assign name="home_phone" expr="accountinfo.acctphone"/>
  </filled>
</subdialog>

<field name="adjustment_amount">
  <grammar type="application/srgs+xml" src="/grammars/currency.grxml"/>
  <prompt>
    What is the value of your account adjustment?
  </prompt>
  <filled>
    <submit next="/cgi-bin/updateaccount"/>
  </filled>
</field>
</form>
</vxml>

```

**Documento que contiene la información subdiálogo de la cuenta** (acct\_info.vxml)

```

<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd"
  version="2.0">
  <form id="basic">
    <field name="acctnum">
      <grammar type="application/srgs+xml" src="/grammars/digits.grxml"/>
      <prompt> What is your account number? </prompt>
    </field>
    <field name="acctphone">
      <grammar type="application/srgs+xml" src="/grammars/phone_numbers.grxml"/>
      <prompt> What is your home telephone number? </prompt>
      <filled>
        <!-- The values obtained by the two fields are supplied
          to the calling dialog by the "return" element. -->
        <return namelist="acctnum acctphone"/>
      </filled>
    </field>
  </form>
</vxml>

```

Subdiálogos agrega un nuevo contexto en la ejecución cuando ellos son invocados. El subdiálogo puede ser un nuevo diálogo dentro del documento existente, o un nuevo diálogo dentro de un nuevo documento.



Subdiálogos se puede componer de varios documentos. La figura 3 demuestra el flujo de la ejecución donde una secuencia de documentos (D) transición a un subdiálogo (SD).

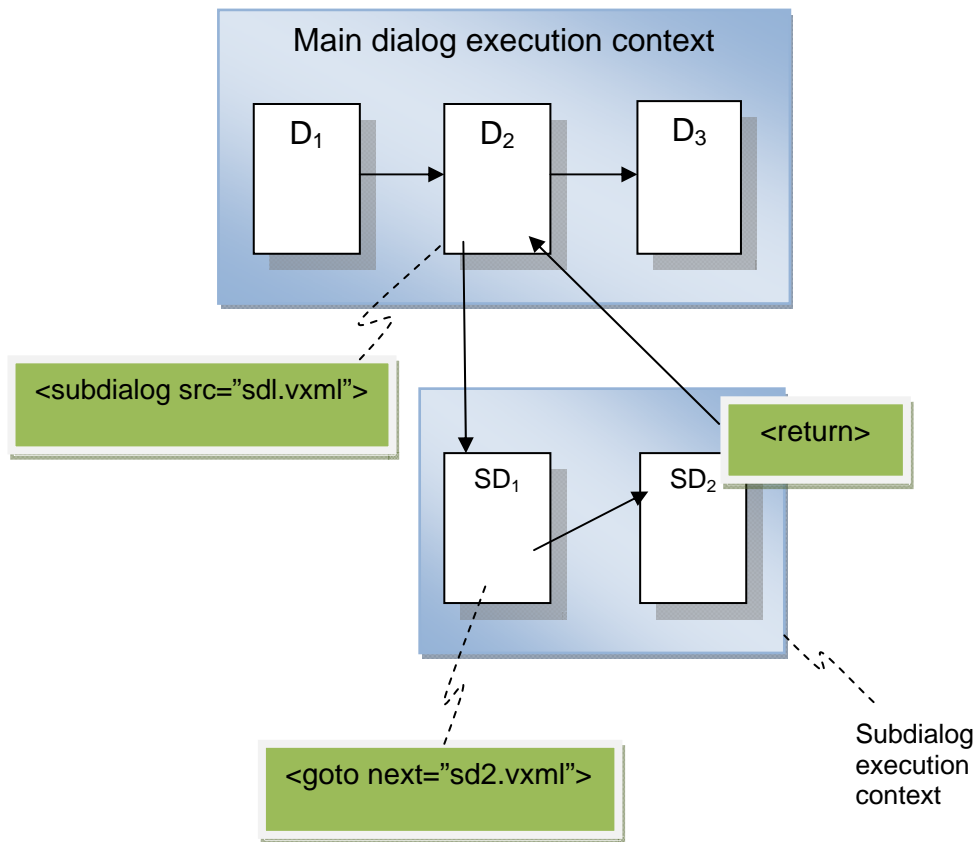


Figura 3: Subdiálogo integrado por varios documentos

El contexto de la ejecución en el diálogo D2 se suspende cuando invoca el subdialog SD1 en el documento sd1.vxml.

Este subdiálogo especifica la ejecución para ser transferido al diálogo en sd2.vxml (el usar `< goto >`). Por lo tanto, cuando el diálogo en sd2.vxml se ejecuta, el control de este es devuelto directamente al diálogo D2 (al usar `<return>`).

El cuadro 4 demuestra un ejemplo de un subdiálogo del multi-documento donde el control se transfiere a partir de un subdiálogo a otro.

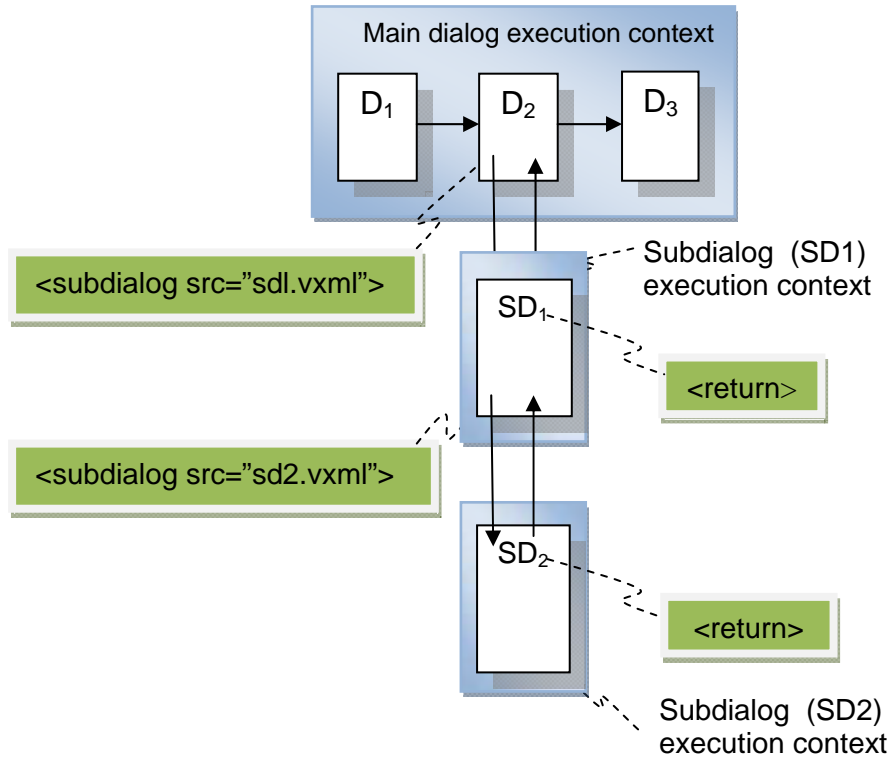


Figura 4: Subdiálogo integrado por varios documentos al volver del primer documento del subdiálogo.

## D. CONSTRUCCIONES DE DIÁLOGO

### 1. Formas o Formularios

Las formas son el componente dominante de los documentos de VoiceXML. Una forma contiene:

1. Un sistema de form items, elementos que se visitan en el lazo principal del algoritmo de la forma. Los items de la forma se subdividen en ítems de entrada que pueden ser llenados por la entrada del usuario y controlar los artículos que no puede.
2. Declaraciones de variables no-form .
3. Manejadores de Eventos.
4. Acciones "filled", bloques de procedimiento lógicos que se ejecutan cuando ciertas combinaciones de variables de entrada son asignadas.

Los atributos del los formularios son:

ID	El nombre del formulario. Si está especificada, la forma se puede referir dentro del documento o de otro documento. Para la instancia: <form id="weather">, <goto next="#weather">.
Scope	El alcance por defecto del formulario. Si es diálogo entonces las gramáticas del formulario son activas

	<p>solamente en el formulario. Si el alcance es el documento, entonces las gramáticas del formulario son activas durante cualquier diálogo en el mismo documento. Si el alcance es el documento y el documento es un documento raíz de la aplicación, entonces las gramáticas del formulario son activas durante cualquier diálogo en cualquier documento de esta aplicación. Observe que el alcance de las gramáticas individuales del formulario toma precedencia sobre el alcance por defecto; por ejemplo, en documentos no-raíz en un formulario con el alcance "dialog" por defecto, y una gramática del formulario con el alcance "document", entonces esa gramática es activa en cualquier diálogo en el documento.</p>
--	---

Tabla 3. Atributos de los formularios.

Esta sección describe algunos de los conceptos detrás de los formularios, y se brindan algunos ejemplos detallados de su operación.

## 2. Elementos de los Formularios

Son los elementos que se pueden visitar en el lazo principal del algoritmo de la interpretación del formulario.

Cuando el FIA (form interpretation algorithm) selecciona un artículo de control, el artículo de control puede contener un bloque del código procedimental para ejecutarse, o puede decir al FIA configurar el prompt y recogerla de un formulario mezclado.

### Elementos de entrada

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

Los artículos de la entrada consisten en:

<b>&lt;FIELD&gt;</b>	Un elemento de entrada cuyo valor se obtiene vía gramáticas de asr o de dtmf.
<b>&lt;record&gt;</b>	Un elemento de entrada cuyo valor es un clip de audio grabado por el usuario. Un elemento <record> puede recoger un mensaje del correo de voz, por ejemplo.
<b>&lt;transfer&gt;</b>	Un elemento de entrada que transfiere a un usuario a otro

	número de teléfono. Si la transferencia devuelve el control, la variable de campo será fijada al estado del resultado.
<b>&lt;object&gt;</b>	Este elemento de entrada invoca un “objeto” con varios parámetros. El resultado del objeto de la plataforma es un objeto ECMAScript. Un objeto de la plataforma podría ser un diálogo que recopila la información de la tarjeta de crédito. Otros podrían recolectar un mensaje de texto usando un cierto método propietario de entrada de texto (DTMF). No hay requisito para las puestas en práctica de objetos para plataforma específicas, aunque las implementaciones deben manejar el elemento <object> con un error “error.unsupported.objectname”, si el objeto en particular no es soportado.
<b>&lt;subdialog&gt;</b>	Un elemento de entrada <subdialog> es como una llamada de función. Invoca otro diálogo en la página actual, o invoca otro documento de VoiceXML. Vuelve un objeto de ECMAScript como su resultado

### Elementos de control

Hay dos tipos de elementos de control:

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

Tabla 5: Elementos de control.

### Variables y Condiciones de los Formularios

Cada elemento del formulario tiene una variable asociada, que por defecto es fijada en indefinido cuando se incorpora al formulario. Esta variable del formulario contendrá el resultado de la interpretación del elemento del formulario.

Típicamente a los elementos de entrada se le asignan nombres., pero a los de control **no**. Generalmente a las variables de los formularios no le son asignados valores iniciales y las condiciones de protección no son especificadas.

Pero a veces hay una necesidad de un control más detallado. Un formulario puede tener una variable fijada inicialmente para ocultar un campo, y más adelante liberarla (usando <clear>).

Otro campo puede tener una condición de protección que lo active solamente cuando no se ha recogido datos, y cuando se han llenado otros campos. Un elemento "block" podría ejecutarse solamente cuando cierta condición sea verdadera. Así, el control se puede realizar sobre la orden en la cual los electos del formulario son seleccionados y ejecutados por el FIA, no obstante en general, muchos diálogos pueden ser construidos sin el recurso a este nivel de la complejidad.

En resumen, todos los elementos de los formularios tienen los siguientes atributos:

NAME	El nombre del variable del formulario dentro de ámbito del dialogo, que mantendrá el valor del elemento del formulario.
Exp	Valor inicial del elemento variable en un formulario.
Cond	Es una expresión para evaluar conjuntamente con la prueba del elemento variable del formulario. Si está ausente, el valor por defecto es verdadero, o en el caso de <initial>, una evaluación determina si ha sido llenado cualquier variable de entrada.

### Formularios dirigidos

El tipo más simple y más común de formularios es en la que los elementos del formulario se ejecutan exactamente una vez en orden secuencial.

Aquí está un servicio informativo del tiempo que utiliza tal formulario.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd">
<form id="weather_info">
<block>Welcome to the weather information service.</block>
<field name="state">
  <prompt>What state?</prompt>
  <grammar src="state.grxml" type="application/srgs+xml"/>
  <catch event="help">
    Please speak the state for which you want the weather.
  </catch>
</field>
<field name="city">
  <prompt>What city?</prompt>
```

```

<grammar src="city.grxml" type="application/srgs+xml"/>
<catch event="help">
  Please speak the city for which you want the weather.
</catch>
</field>
<block>
  <submit next="/servlet/weather" namelist="city state"/>
</block>
</form>
</vxml>

```

Este diálogo se procesa secuencialmente:

C (computadora): Bienvenido al servicio informativo del tiempo. ¿Cuál es el nombre del estado?

H (humano): Ayuda

C: Por favor hable, el estado para el cual usted desea conocer el tiempo.

H: Georgia

C: ¿Qué ciudad?

H: Tblisi

C: No comprendo. ¿Qué ciudad?

H: Macon

C: Las condiciones en Macon Georgia están asoleadas y claras a las 11 a.m

### 3. Menús.

Un menú es un dialogo con opciones donde se debe seleccionar una obligatoriamente. El menú siguiente ofrece al usuario tres opciones:

```

<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd">
<menu>
  <prompt>
    Welcome home. Say one of: <enumerate/>
  </prompt>
  <choice next="http://www.sports.example.com/vxml/start.vxml">
    Sports
  </choice>
  <choice next="http://www.weather.example.com/intro.vxml">
    Weather
  </choice>
  <choice next="http://www.stargazer.example.com/voice/astronews.vxml">
    Stargazer astrophysics news

```

```
</choice>
<noinput>Please say one of <enumerate/></noinput>
</menu>
</vxml>
```

Este diálogo podría seguir de la forma siguiente:

C (computadora): Bienvenido a su hogar. Diga uno de: deportes; tiempo; Noticias de astrofísica de Stargazer.

H (humano): Astrología.

C: No entiendo lo que usted dijo. (un mensaje plataforma-específico.)

C: Bienvenido a su hogar. Diga uno de: deportes; tiempo; Noticias de astrofísica de Stargazer.

H: deportes.

C: (procesa <http://www.sports.example.com/vxml/start.vxml>)

## Elementos de menús

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

<b>id</b>	El identificador del menú. Permite que el menú sea el blanco de un <code>&lt; goto &gt;</code> o un <code>&lt; submit &gt;</code> .
<b>scope</b>	El alcance de la gramática de menú. Si es diálogo (por defecto), las gramáticas del menú son solamente activas cuando el usuario navega en él. Si el alcance es un documento, sus gramáticas son activas sobre el documento entero.
<b>dtmf</b>	Cuando se configura como verdadero, las primeras nueve opciones que no han especificado explícitamente un valor para el atributo del dtmf son dadas implícitas "1", "2", etc. Las opciones restantes no tienen especificaciones explícitas para los atributos del dtmf, es decir no serán asignadas valores de DTMF. Si hay opciones que han especificado su propio DTMF como la utilización de secuencias como "*", "#", o "0", un error.badfetch será lanzado. El valor por defecto es falso.
<b>accept</b>	Cuando se configura como "exact" (por defecto), el texto de los elementos escogidos en el menú define la frase exacta que se reconocerá. Cuando se configura como

	"approximate", el texto de los elementos escogidos define una frase aproximada del reconocimiento. Cada < opción > puede eliminar este ajuste.
--	--

Tabla 6: Atributos de < menú >

### Elemento <choice>

El elemento < choice> responde a varios propósitos:

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

Los atributos del elemento <choice> son:

<b>dtmf</b>	La secuencia de DTMF para esta opción. Es equivalente a un simple DTMF <grammar> y las características de DTMF se aplica al reconocimiento de la secuencia. A diferencia de gramáticas de DTMF, los espacios en blanco son opcionales: dtmf="123#" es equivalente a dtmf="1 2 3 #".
<b>accept</b>	Cuando se configura como "exact" (por defecto), el texto de los elementos escogidos en el menú define la frase exacta que se reconocerá. Cuando se configura como "approximate", el texto de los elementos escogidos definen una frase aproximada del reconocimiento
<b>next</b>	El URI del diálogo o del documento siguiente.
<b>expr</b>	Especifica una expresión para evaluar como se envía a un URI.
<b>event</b>	Especifica el lanzamiento de un acontecimiento en vez de especificar el siguiente.
<b>eventexpr</b>	Una expresión de ECMAScript que evalúa al nombre del acontecimiento que se lanzará.
<b>message</b>	Una secuencia del mensaje que proporciona contexto adicional sobre el acontecimiento que es lanzado. El mensaje está disponible como el valor de una variable dentro del alcance del elemento.
<b>messageexpr</b>	Una expresión de ECMAScript que evalúa la secuencia del mensaje.
<b>fetchaudio</b>	Esto omite la característica del fetchaudio.
<b>fetchhint</b>	Esto omite la característica del documentfetchhint.
<b>fetchtimeout</b>	Esto omite la característica del fetchtimeout.



<b>maxage</b>	Esto omite la característica del documentmaxage.
<b>maxstale</b>	Esto omite la característica del documentmaxstale.

Tabla 7: Atributos < opción >

Exactamente uno de "next", "expr", "event" o "eventexpr" debe ser especificado; si no, un acontecimiento de error.badfetch es lanzado. Exactamente uno de "message" o de "messageexpr" puede ser especificado; si no, se lanza un acontecimiento de error.badfetch.

Si un elemento gramatical < grammar > es especificado en <choice>, entonces la gramática externa se utiliza en vez de una gramática generada automáticamente. Esto permite que el desarrollador controle precisamente la gramática de <choice>; por ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd">
<menu>
  <choice next="http://www.sports.example.com/vxml/start.vxml">
    <grammar src="sports.grxml" type="application/srgs+xml"/>
    Sports
  </choice>
  <choice next="http://www.weather.example.com/intro.vxml">
    <grammar src="weather.grxml" type="application/srgs+xml"/>
    Weather
  </choice>
  <choice next="http://www.stargazer.example.com/voice/astronews.vxml">
    <grammar src="astronews.grxml" type="application/srgs+xml"/>
    Stargazer astrophysics
  </choice>
</menu>
</vxml>
```

## DTMF en menús

Los menús pueden confiar en Speech, en DTMF, o en ambas combinaciones incluyendo el elemento <property> en el elemento menú <menú>. A continuación se muestra un menú DTMF con secuencias explícitas de DTMF dadas a cada opción, usando los atributos dtmf del elemento <choice>:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/vxml
http://www.w3.org/TR/voicexml20/vxml.xsd">
<menu>
<property name="inputmodes" value="dtmf"/>
<prompt>
For sports press 1, For weather press 2, For Stargazer
astrophysics press 3.
</prompt>
<choice dtmf="1" next="http://www.sports.example.com/vxml/start.vxml"/>
<choice dtmf="2" next="http://www.weather.example.com/intro.vxml"/>
<choice dtmf="3" next="http://www.stargazer.example.com/astronews.vxml"/>
</menu>
</vxml>

```

Alternativamente, se puede configurar los atributos en <menu> para asignar una secuencia de dígitos DTMF para cada uno de los primeros nueve opciones que no han especificado sus propias secuencias de DTMF: la primera opción tiene DTMF "1", "2", etc:

```

<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/vxml
http://www.w3.org/TR/voicexml20/vxml.xsd">
<menu dtmf="true">
<property name="inputmodes" value="dtmf"/>
<prompt>
For sports press 1, For weather
press 2, For Stargazer astrophysics press 3.
</prompt>
<choice next="http://www.sports.example.com/vxml/start.vxml"/>
<choice next="http://www.weather.example.com/intro.vxml"/>
<choice dtmf="0" next="#operator"/>
<choice next="http://www.stargazer.example.com/voice/astronews.vxml"/>
</menu>
</vxml>

```

## Enumerar elementos

El elemento <enumerate> es una descripción generada automáticamente de las opciones disponibles para el usuario. Especifica una plantilla que se aplica a cada opción en el orden que aparece en el menú. Si se utiliza sin contenido, una plantilla por defecto que enumera todas las opciones es utilizada por el contexto del intérprete.

Si tiene contenido, este es especificado por la plantilla y puede referirse a dos variables especiales: `_prompt` es el aviso de la opción, y `_dtmf` es una representación

normalizada (es decir un solo un espacio en blanco entre el símbolo DTMF) de la secuencia DTMF asignada en la opción (note que si no se asigna ninguna secuencia de DTMF al elemento <choice>, o un elemento de la <grammar> es especificado en <choice>, entonces la variable \_dtmf asigna el valor undefined ECMAScript). Por ejemplo, si el menú visto en la sección 2.2.3 se reescribe como:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd">
<menu dtmf="true">
  <prompt>
    Welcome home.
  <enumerate>
    For <value expr="_prompt"/>, press <value
      expr="_dtmf"/>.
  </enumerate>
</prompt>
<choice next="http://www.sports.example.com/vxml/start.vxml">
  sports </choice>
<choice next="http://www.weather.example.com/intro.vxml">
  weather </choice>
<choice next="http://www.stargazer.example.com/voice/astronews.vxml">
  Stargazer astrophysics news
</choice>
</menu>
</vxml>
```

Entonces el aviso será como:

C: Bienvenido a su hogar. Para los deportes, presione 1. Para el tiempo, presione 2. Para las noticias de la astrofísica de Stargazer, presione 3.

El elemento <enumerate> puede ser utilizado dentro de prompts y elementos catch asociados con elementos del < menú > y elementos < field > que contienen elementos <opción>, según lo descrito en secciones anteriores. Se lanza un acontecimiento de error.semantic si <enumerate> se utiliza dentro de otro (por ejemplo, < enumerate > dentro de < enumerate >).

## CAPITULO III

### A. PROPUESTA DE SOLUCION.

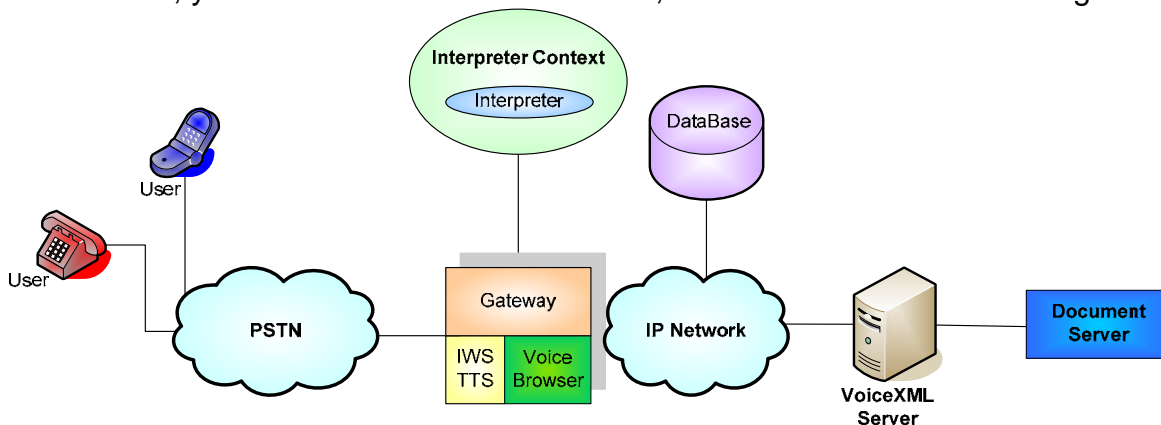
En este capítulo se detalla en primer lugar todos los elementos involucrados en el Sistema. Se comienza describiendo cada uno de los servicios que se pondrán en funcionamiento, así también la arquitectura del sistema propuesto.

Es importante mencionar que todos los servicios implementados fueron tomados en base a los requisitos que el sistema debería de cumplir. Se presenta la factibilidad técnica, económica y operativa del proyecto.

Una de las partes mas importantes en la implementación de un sistema es el diseño de la base de datos por lo que se da una breve explicación de la entidad relación de esta, así también se presenta un diagrama HIPO. Finalmente se menciona los pasos a seguir para la instalación de los elementos mas principales del sistema.

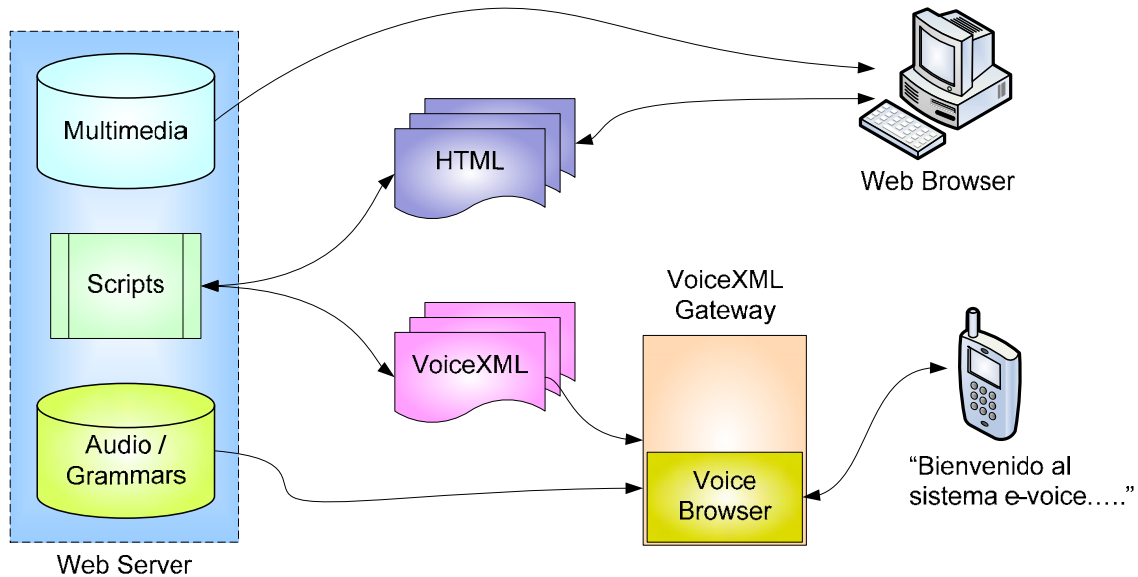
### B. ARQUITECTURA DEL SISTEMA PROPUESTO

VoiceXML permite el acceso a los servicios apoyándose en una arquitectura cliente/servidor, y de la infraestructura Telefónica, tal como se muestra en la figura 1.



**Figura 1. Arquitectura del sistema propuesto.**

La arquitectura de VoiceXML se parece tanto a la arquitectura de tecnologías Web tradicionales como a la arquitectura de WAP. El usuario puede acceder por líneas fijas o móviles. El navegador VoiceXML típico corre sobre un Gateway de Voz especializado que se conecta tanto a la red pública de telefonía conmutada como a Internet. Estos Gateways de voz extienden el poder de la Web a teléfonos tradicionales.



**Figura 2. Interacción entre los lenguajes de programación**

Los elementos que forman parte de la arquitectura del sistema son:

- **Servidor:** realiza la función de un servidor Web. Está compuesto(entre otros elementos) por: El Servidor de documentos cuya función es contener todos los documentos Vxml estáticos y dinámico y facilitárselos al interprete Vxml (Vxml Browser) a medida que vayan siendo solicitados. En nuestro caso se esta utilizando un Servidor Web Apache versión 2.0.54 montado sobre plataforma Linux.
- **Gateway VoiceXML:** Elemento encargado de: responder a las llamadas de los usuarios y ejecutar los documentos Vxml, para ello se comunica con el servidor WEB utilizando el protocolo http que es el encargado de obtener los documentos adecuados.

Los Gateway VoiceXML a nuestro criterio lo hemos dividido en dos partes:

- **Intérprete VoiceXML:** Este Modulo esta compuesto por VXIAsterisk en nuestro Proyecto (pero se puede utilizar OpenVXI 3.0 ó Sipvxml que son Opensource), este interpreta los documentos VXML que son recibidos del Servidor Web , y además dirige y controla el diálogo entre el usuario-máquina.
- **El interprete de contexto VoiceXML:** se considera un proceso ejecutado en "back-ground", Recoge entradas del usuario del usuario en paralelo con el intérprete de VoiceXML. Se encarga de las siguientes funciones:
  - Permitir el diálogo entre el Intérprete y le Servidor de Documentos

- Procesar determinados eventos no relacionados directamente con el estado actual del diálogo en curso ( nuevas llamadas entrantes...

Para interactuar con el usuario y para realizar todas las acciones programadas en el documento Vxml, el Gateway debe disponer de:

- Mecanismo de interacción entre el servidor de documentos y el intérprete para solicitar y recoger los documentos
- Mecanismos para la reproducción de ficheros de audio
- Mecanismos para la generación de audio a partir de texto (Text-To-Speech)
- Mecanismos para la recogida de datos de usuario ya sea mediante mecanismos de reconocimiento de voz (IVR) como de reconocimiento de tonos DTMF.
- Mecanismos para la grabación de voz.

El Interprete para poder llevar algunas de las tareas que anteriormente se mencionaron, necesita una forma para poder comunicarse con otros módulos de apoyo, como: TTS e IVR.

En el caso de la comunicación con el modulo de TTS, el Interprete VXML tiene implementada una interfaz la cual sirve para ejecutar la aplicación de reconocimiento de Texto (TTS). Esta interfaz es un archivo programado en PHP cuyo nombre es "tts.php", este nombre puede ser cualquiera y se define en el archivo de configuración del Vxml Browser que para el caso de VXiAsterisk se llama: "client.cfg".

La aplicación de reconocimiento de Texto utilizada para el lenguaje en español es un motor TTS conocido como "swift", y es desarrollado por la empresa Cepstral. Este motor tiene la ventaja que trabaja a nivel de comando y tiene una gran variedad de voces, aunque estas son licenciadas. Una alternativa OpenSource puede ser Festival. En el caso del idioma ingles se esta utilizando una version reducida de Festival conocida como "flite".

- **Interfaces de I/O (Voice Proccesing Board o SIP Proccesing interface):** Para poder conectarse a los sistemas telefónicos llámese fijo, móvil o IP, es necesario que la plataforma implementada en nuestro proyecto tenga una forma para conectarse a estos sistemas. Esta conexión se puede hacer de diferentes maneras bien utilizando interfaces de telefonía analógicas o digitales y en el caso de telefonía IP, se puede hacer utilizando el protocolo SIP.

Nuestro proyecto utiliza interfaces de telefonía analógicas con puertos FX0 ya que podemos tener la opción de conectarlo a una Línea FXS (Foreign Exchange Station) de algún proveedor telefónico o a una PBX con puertos análogos FXS (Foreign Exchange Office).



**Figura 3. Interfaces Telefónicas.**

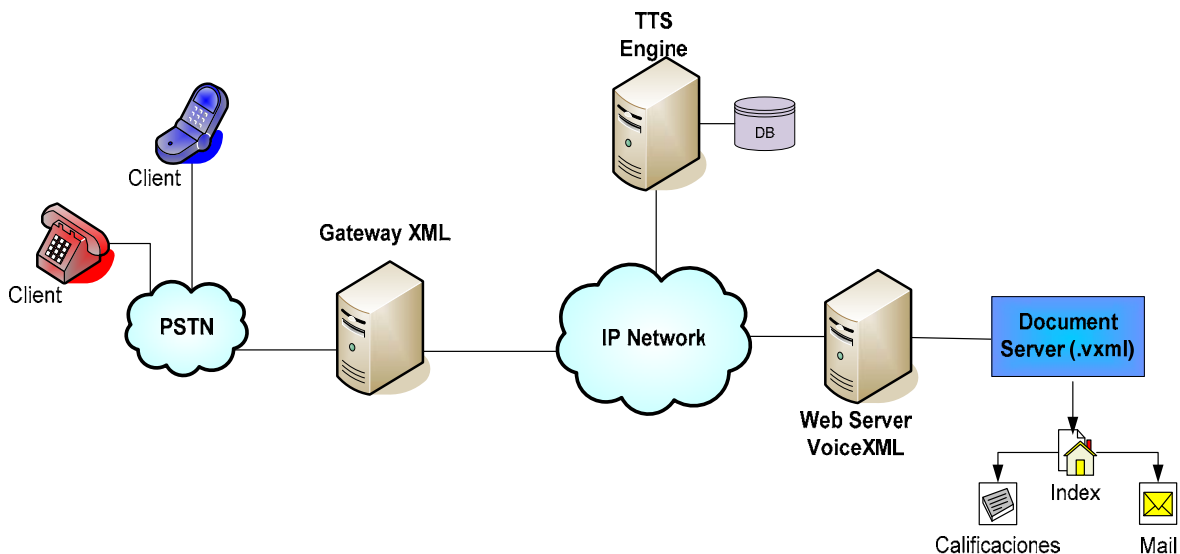
Si se piensa conectar este sistema a la PSTN<sup>5</sup> utilizando interfaces digitales, esta conexión se puede hacer utilizando tarjetas **E1 (TDM)**, esto puede permitir recibir hasta un máximo de 31 llamadas simultáneas.

Otra forma, como se mencionó anteriormente es hacerlo vía Redes de Datos (IP), utilizando el protocolo SIP<sup>6</sup>, que es ampliamente utilizado para aplicaciones de Telefonía IP, esta forma de conexión también se utilizará en la aplicación a desarrollar.

Para poder integrar el Interprete VXML (VXML Browser) con estas formas de interfaces conectadas con los sistemas Telefónicos se usó un programa Intermediario llamado Asterisk, el cual es un programa para implementación de Sistemas PBX.

## 1. Arquitectura del servicio

La arquitectura del servicio que hemos generado se puede resumir en la figura 4, que se presenta a continuación:



**Figura 4. Arquitectura del Sistema Implementado**

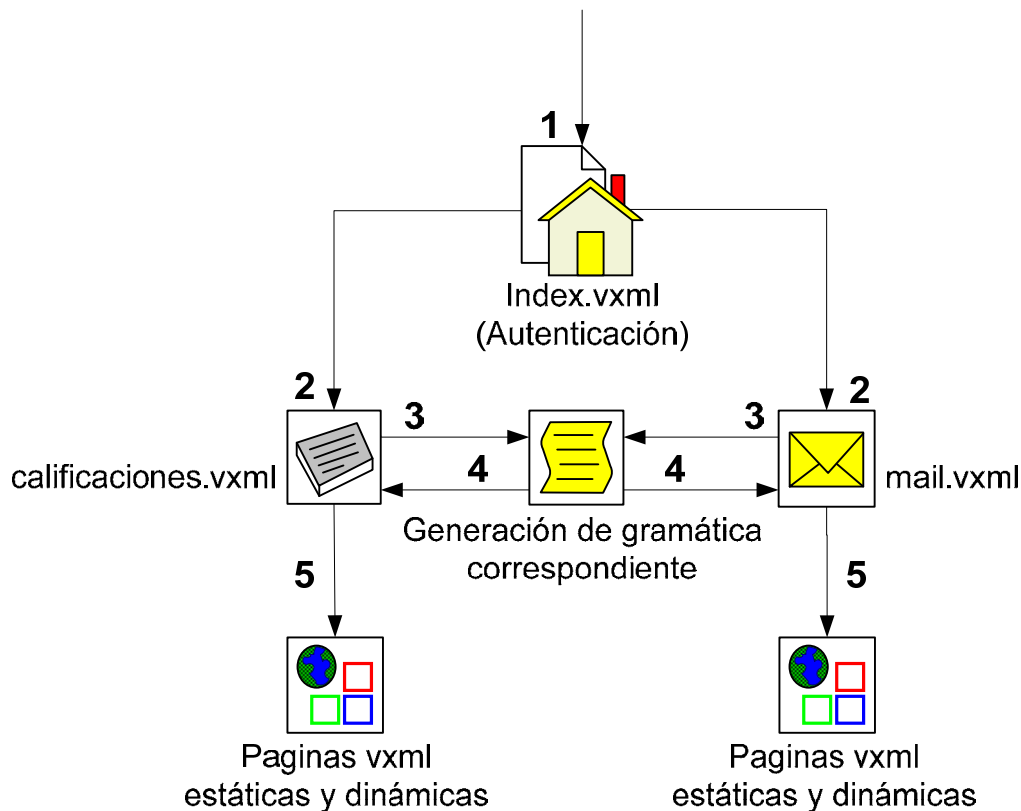
<sup>5</sup> Public switched telephone network

<sup>6</sup> Session initiation protocol

El proceso es el siguiente:

El cliente realiza una llamada telefónica al servicio solicitado y contacta con el Gateway VXML que es el encargado de responder a las llamadas de los usuarios. Éste según vaya siendo necesario dialoga con el Servidor de Documentos para obtener los documentos adecuados a medida que vayan siendo solicitados y posteriormente se los pasa al intérprete para que el cliente pueda operar en función de las diferentes posibilidades. El intérprete es el encargado de llevar el hilo de la llamada una vez que esta ya se ha establecido.

Los documentos que se encuentran en el Servidor de Documentos son ficheros .vxml estáticos y dinámicos. Y como se mencionó anteriormente se encuentran en un servidor Web.



**Figura 5. Diagrama general de navegación entre documentos.**

A continuación describimos el procedimiento de unos de los servicios:



## **Envío de VoiceMail vía Correo Electrónico a grupos de Usuarios**

1. El usuario ingresa a un portal, en este el usuario crea grupos de usuarios, por ejemplo: Trabajo, Oficina, Proyecto, etc.; Luego el usuario, dentro de estos grupos agrega cuentas de correo electrónico y el nombre completo del usuario. (por ejemplo: juan.perez@ufg.edu.sv, Juan Pérez). El usuario puede crear tantos grupos y usuarios dentro estos grupos quiera.
2. Si el usuario quiere enviar un correo a cualquiera de los grupos de correos creados vía portal, los pasos son los siguientes:
  - El usuario marca un número telefónico. Le solicitará autenticarse y posteriormente le saldrá un menú, en donde le pedirá que para enviar un correo a un grupo de usuarios marque "x" numero.
  - El sistema le listará los grupos que tiene disponibles (El sistema le dirá por ejemplo: "Tiene 3 grupos habilitados, para oficina marque 1, para Universidad marque 2, para trabajo marque 3).

Al marcar cualquiera de los números que lista el sistema, le solicitará que deje su mensaje de voz, después de un tono (el tiempo de duración de cada mensaje es de 10seg), luego de que el usuario haya grabado su mensaje, el sistema se lo reproducirá y le preguntará al usuario si lo quiere salvar y enviar (presionando 1) o si lo quiere grabar nuevamente (presionando 2). Si el usuario no presiona cualquiera de estos números, el sistema le dirá que presione un número válido. Se lo repetirá tres veces hasta que el usuario digite un número correcto.

Como complemento al servicio de poder escuchar los correos electrónicos vía teléfono, se han desarrollado una aplicación que permite hacer notificaciones de nuevos correos, si estos son de sus contactos. Las dos diferentes formas de notificaciones se describen a continuación:

### ***Notificaciones vía SMS***

Es importante mencionar que el servicio del escucha de correos electrónicos tiene una opción adicional que consiste en que el usuario puede activar el envío de mensajes SMS a su teléfono celular, cuando tenga en un nuevo mensaje en su buzón de e-mail.

Es importante mencionar que el usuario simplemente ingresa su número de celular y activa el servicio en el Portal Web, y la aplicación (e-sms) identifica de que compañía pertenece el número que el usuario ingreso.

Es hacer notar que por el momento este servicio no funcionará para las empresas Telefónica (Movistar) e Intelfon (Red), ya que tienen desactivado dicho servicio. Para el caso de la compañía Telemovil (Tigo) el usuario deberá activar este servicio en la página Web de dicha compañía.

Este servicio de Notificaciones de entrada de correos nuevos vía SMS tiene un archivo de configuración en donde se ingresan los valores de las variables que se utilizan en este programa. Cabe mencionar que este programa corre como demonio en Linux.

El archivo de configuración antes mencionado se llama “e-sms.conf” y esta ubicado en la carpeta “etc”.

Para el caso de que un usuario ingrese un número de celular de las compañías Telefónica e Intelfon, se le enviará un correo a su cuenta haciéndole saber que los números telefónicos de dichas compañías no tienen activado el envío de mensajes SMS.

El remitente de dichos correos al igual que el de notificaciones será la cuenta:  
e-voice@ufg.edu.sv.

### ***Notificación vía Teléfono***

Además de las notificaciones vía SMS, el sistema tendrá la posibilidad de notificar al usuario vía teléfono, en que momento tiene un su buzón, un nuevo correo enviado por alguno de los contactos.

Para la activación de este servicio y de notificaciones vía SMS, el usuario tendrá que llenar una solicitud vía Web.

## C. ESQUEMA DE SOLUCION DE LA PROPUESTA



**Figura 6. Esquema de solución.**

### 1. Definición de requerimientos de la aplicación

- Consulta de correos electrónicos para ello se debe:
  - Contar con una aplicación Web que pueda administrar a los usuarios a los que se les quiere escuchar correos, vía teléfono.
  - Listar los nombres antes de escuchar vía teléfono los correos de un determinado usuario.
  - Enumerar los correos por asunto del usuario seleccionado.
  - Responder correos del usuario seleccionado.
  - Borrar correos.
  - Escuchar la hora y día que se recibió un correo del usuario seleccionado.
- Poder enviar correos de voz vía e-mail a un grupo de usuarios para ello se debe:
  - Contar con una aplicación Web, que pueda agregar diferentes grupos de usuarios.
  - Enumerar los diferentes grupos que el usuario ha creado.
  - Permitir regrabar el mensaje si el usuario no está satisfecho con el que grabo.
  - Informar si el correo fue enviado satisfactoriamente a los usuarios del grupo.
- Poder redireccionar llamadas a un Call Center, para ello se debe:
  - Tener la posibilidad de que su llamada sea direccionada a un Call Center.

- Notificación de nuevos correos vía SMS, para ello se debe:
  - Notificar vía mensajes SMS que tiene un correo reciente de sus contactos.
- Notificación de nuevos correos vía Teléfono para ello se debe:
  - Tener otra forma de notificación al usuario, indicándole que tiene mensajes de correo recientes de sus contactos. Esta forma deberá de ser por medio de un mensaje de voz, enviado a su teléfono fijo, celular o IP.

## Factibilidad técnica y económica

La Universidad Francisco Gavidia, cuenta con los recursos necesarios tanto de hardware y software para la implementación y diseño del proyecto, con esto es posible llevar a buen termino el desarrollo de la solución, basándose en software libre.

- **Software:** El desarrollo del proyecto lleva consigo la utilización de software libre y software comercial.

<i>Descripción</i>	<i>Software</i>	<i>Precio</i>
Sistema operativo	Linux SUSE 10.2	
Software de Diseño	Macromedia DreamWeaver 8.0	\$383.99 <sup>7</sup>
Bases de Datos	MySQL.	
Software para Central Telefónica	Asterisk	
Intérprete de lenguaje VXML	VXIasterisk	
Text to speech.	Cepstral (Engine TTS para Idioma Español). Flite (Engine TTS para Idioma Ingles)	\$ 30.00 <sup>8</sup>
Software identificador de idioma	Dbacl.	
Servidor web	Apache	
Lenguaje de programación	Perl PHP	
<b>Total</b>		<b>\$ 413.99</b>

- **Hardware:** En cuanto a hardware, el equipo necesario para la implementación del proyecto es crítico ya que sin esto el software no es factible y no se podría realizar, así mismo, el número de peticiones satisfechas dependerá de la cantidad de canales de reconocimiento de texto y la cantidad de puertos FXO que pueda administrar la tarjeta.:

<sup>7</sup> <http://www.amazon.com>

<sup>8</sup> <http://www.cepstral.com>

<b>Dispositivo</b>	<b>Tipo</b>	<b>Precio</b>
Server	Pentium III, 1.0 Ghz, 512 MB RAM, 20 GB. Tarjeta de Red.	\$210.00
	Pentium III, 533 Ghz, 512 MB RAM, 20 GB. Tarjeta de Red.	\$150.00
Tarjeta para conexión de líneas análogas.	Tarjeta Digium TDM400P	\$381.00 <sup>9</sup>

En conclusión se dispone de todo el hardware y software necesario para la realización del proyecto.

## 2. Diseño del producto

A continuación se detallan los lineamientos a considerar para la realización del proyecto, esto contempla: Diseño de Base de Datos, Diseño de interfaz web, flujo de datos de los scripts a utilizar, y la descripción de las herramientas a utilizar.

### **Estándares para el desarrollo de Aplicaciones**

Se tiene como objetivo definir el estándar para el desarrollo de las diferentes aplicaciones que se elaboran con el sistema voice XML; esto con el fin de agilizar el diseño y programación de los mismos.

La nomenclatura a utilizar es la siguiente:

A. **Tablas:** el nombre de cada tabla estará formada máximo por 14 caracteres, ejemplo:

t	e	v	_	e	m	a	i	l	g	r	o	u	p
1	2	3	4	5	6	7	8	9	10	11	12	13	14

1. Siempre tendrá una “t” para identificar que es una tabla.
2. Primeros dos carácter según la abreviatura que se defina para la aplicación.
3. Carácter “\_” obligatorio en la posición 4.
4. De esta posición a la 14, se definirá el nombre de la tabla.

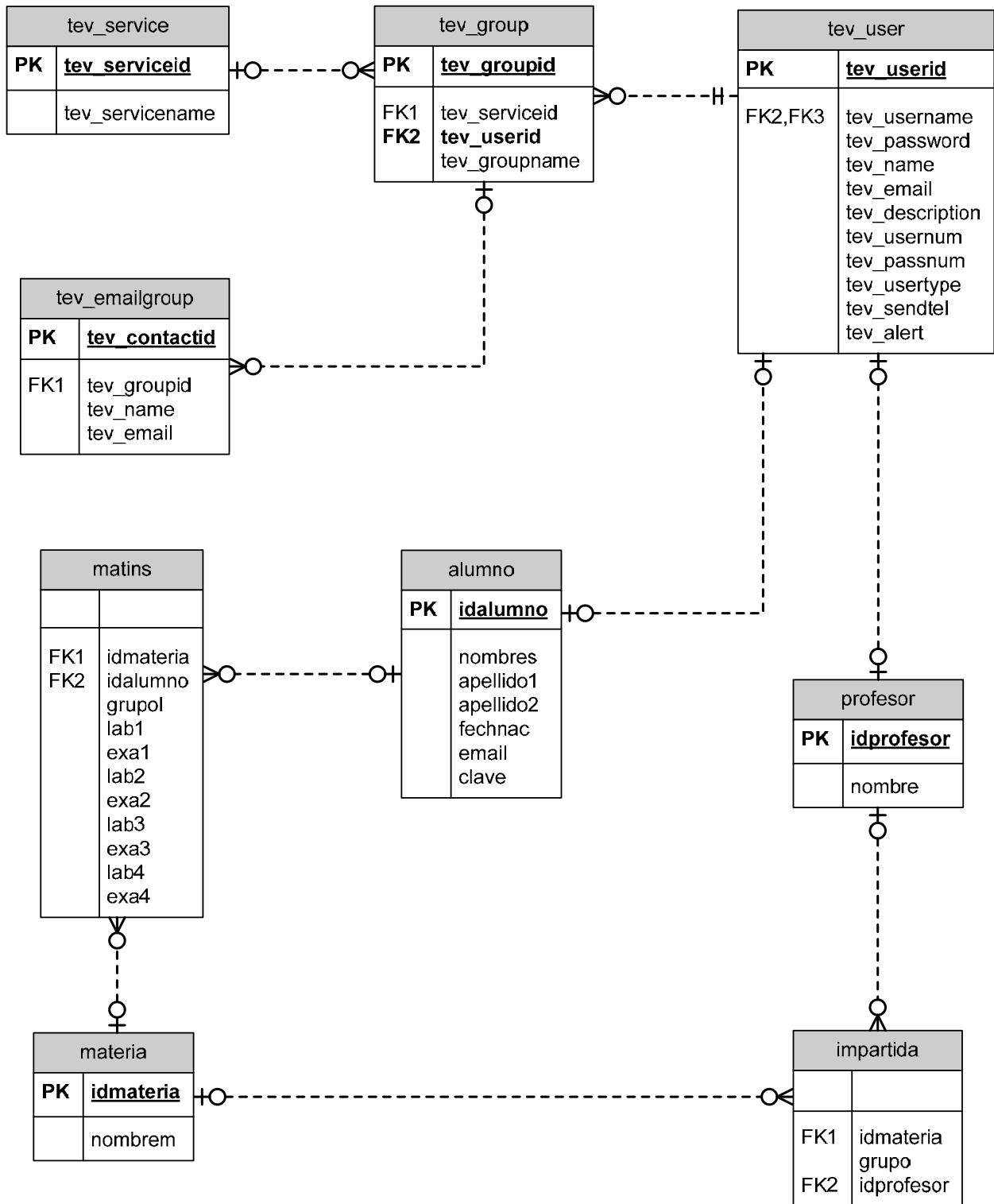
B. **Campos:** el nombre de cada campo estará formado por 10 caracteres,

t	e	v	_	s	e	r	v	i	c	e	i	d	
1	2	3	4	5	6	7	8	9	10	11	12	13	14

1. Primeros dos carácter según la abreviatura que se defina para la aplicación.
2. Carácter “\_” obligatorio en la posición 4.
3. De esta posición a la 14, se definirá el nombre del campo.

<sup>9</sup> Digium.com

## Diagrama entidad relación de aplicativo E-VOICE



## Diccionario de datos

### tev\_user

Tabla que aloja los datos de información personal relacionados con los usuarios del Sistema eVoice como: el usuario y clave de acceso, nombre, dirección de e-mail, tipo de usuario, etc. Esta tabla esta relacionada con las tablas tev\_group, alumno y profesor.

Nombre físico	Tipo	Longitud	Tipo de llave/opción	Comentario
tev_userid	int	11	PK, no null	Identificador del usuario
tev_username	varchar	10	FK2,FK3, no null	Usuario de la persona
tev_password	varchar	15	no null	Clave de acceso
tev_name	varchar	50	Null	Nombre del usuario
tev_email	varchar	50	Null	E-mail del usuario
tev_usernum	varchar	10	Null	Almacena la representación DTMF del usuario
tev_passnum	varchar	15	Null	Almacena la representación DTMF del password
tev_description	longtext	-	Null	Descripción sobre el usuario
tev_usertype	varchar	5	Null	Tipo de usuario
tev_sendtel	varchar	8	Null	Almacena el número de celular de un usuario del sistema.
tev_alert	integer	11	Null	Tipo de notificación ó servicio que solicita un usuario.

### tev\_group

Tabla que almacena los datos relacionados con los grupos de individuos que son creados dentro del sistema eVoice por cualquier usuario del mismo. Tiene relaciones con las tablas tev\_service, tev\_user y tev\_emailgroup.

Nombre físico	Tipo	Longitud	Tipo de llave/opción	Comentario
tev_groupid	int	11	PK, no null	Número identificador del grupo
tev_serviceid	int	11	FK1, no null	Número identificador del servicio
tev_userid	int	11	FK2, no null	Número identificador del usuario
tev_groupname	longtext	-	Null	Almacena nombre del grupo

### tev\_emailgroup

Esta tabla aloja datos de las personas de contacto que posee un usuario del Sistema eVoice como el nombre, e-mail y a que grupo pertenece el contacto. Esta relacionada con la tabla tev\_group.

Nombre físico	Tipo	Longitud	Tipo de llave/opción	Comentario
tev_contactid	int	11	PK, no null	Número identificador del contacto
tev_serviceid	int	11	FK1, null	Número identificador del servicio
tev_groupid	int	11	Fk2, null	Número identificador del grupo
tev_name	longtext	-	Null	Nombre del usuario
tev_email	longtext	-	Null	E-mail del usuario

### tev\_service

Tabla que almacena datos de los tipos de servicio incluidos en el Sistema eVoice, estos servicios son: Envío de voicemail y lectura de e-mail. Esta relacionada con la tabla tev\_group.

Nombre físico	Tipo	Longitud	Tipo de llave/opción	Comentario
tev_serviceid	int	11	PK, no null	Número identificador del servicio
tev_servicename	longtext	-	null	Nombre del servicio

### alumno

Esta tabla aloja todos los datos de información personal de los alumnos de la UFG, los datos de esta tabla son llamados desde la base de datos de Registro Académico de la UFG. Esta relacionada con las tablas tev\_user y matins.

Nombre físico	Tipo	Longitud	Tipo de llave/opción	Comentario
idalumno	char	10	PK,null	Número identificador del alumno
nombres	char	10	Null	Nombres del alumno
apellido1	char	10	Null	Primer apellido del alumno
apellido2	char	10	Null	Segundo apellido del alumno
fechnac	datetime	-	Null	Fecha de nacimiento del alumno
email	char	10	Null	E-mail del alumno
clave	char	10	Null	Clave de acceso del alumno



## matins

Tabla que almacena datos de las materias que son inscritas por cada alumno de la UFG. Los datos de esta tabla son llamados desde la base de datos de Registro Académico de la UFG. Esta relacionada con las tablas alumno y matins.

Nombre físico	Tipo	Longitud	Tipo de llave/opción	Comentario
idmateria	char	4	FK1, null	Número identificador de materia
idalumno	char	10	FK2, null	Número identificador del alumno
grupol	char	4	Null	Grupo de la materia
lab1	decimal	10,2	Null	Almacena calificación de laboratorio 1
exa1	decimal	10,2	Null	Almacena calificación de examen parcial 1
lab2	decimal	10,2	Null	Almacena calificación de laboratorio 2
exa2	decimal	10,2	Null	Almacena calificación de examen parcial 2
lab3	decimal	10,2	Null	Almacena calificación de laboratorio 3
exa3	decimal	10,2	Null	Almacena calificación de examen parcial 3
lab4	decimal	10,2	Null	Almacena calificación de laboratorio 4
exa4	decimal	10,2	Null	Almacena calificación de examen parcial 4

## materia

Tabla que almacena el número identificador y el nombre de todas las materias que son impartidas en las distintas carreras que ofrece la UFG. Los datos de esta tabla son llamados desde la base de datos de Registro Académico de la UFG. Esta relacionada con las tablas impartida y matins.

Nombre físico	Tipo	Longitud	Tipo de llave/opción	Comentario
idmateria	char	4	PK, no null	Número identificador de la materia
nombrem	longtext	-	Null	Almacena el nombre de la materia

### profesor

Tabla que aloja un número identificador y nombre de los docentes que laboran para la UFG. Esta tabla tiene relación con las tablas impartida y tev\_user.

Nombre físico	Tipo	Longitud	Tipo de llave/opción	Comentario
idprofesor	char	10	PK, no null	Número identificador del profesor
nombre	longtext	-	Null	Nombre del profesor

### impartida

Tabla que almacena datos relacionados con el nombre del docente a quien se asigna impartir una materia en un determinado grupo. Los datos de esta tabla son llamados desde la base de datos de Registro Académico de la UFG y esta relacionada con las tablas: profesor y materia.

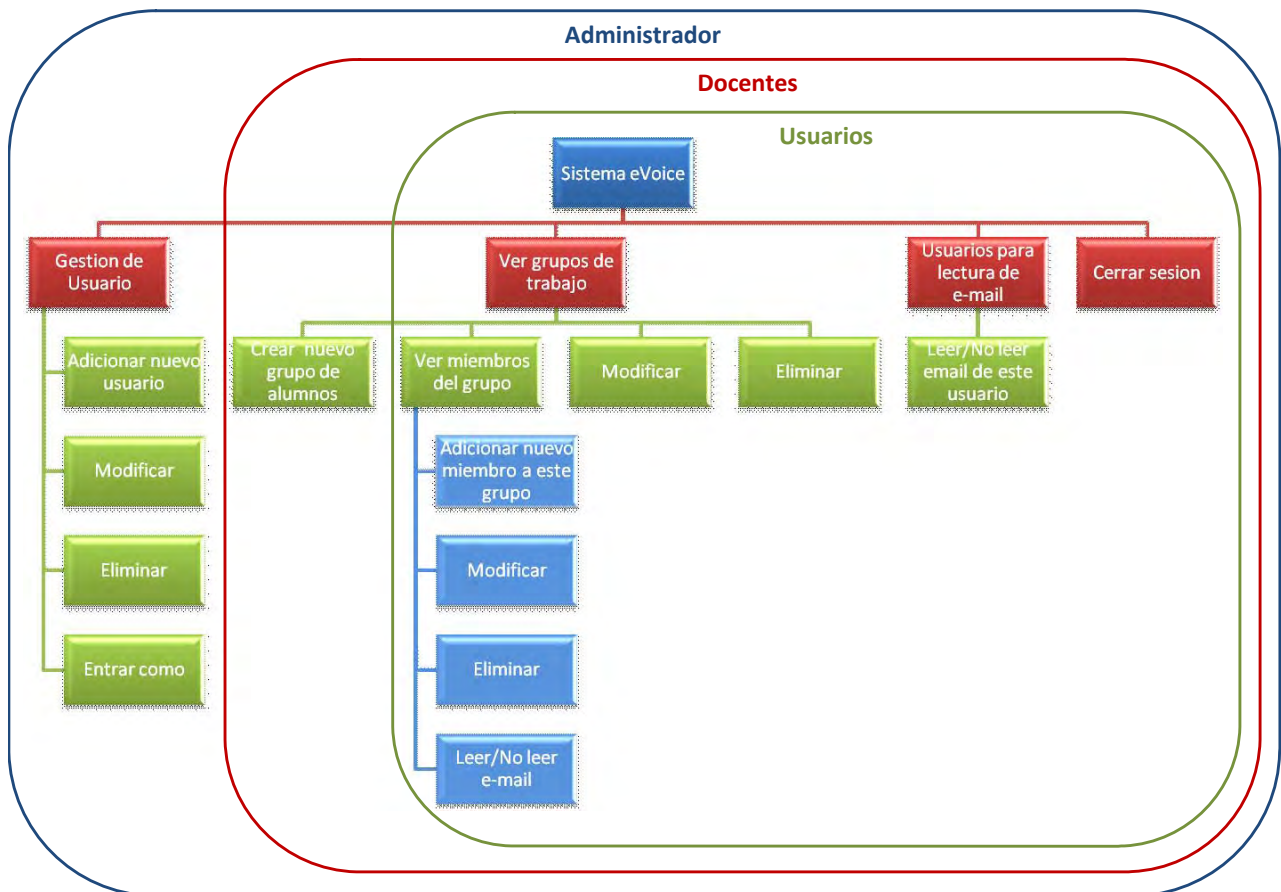
Nombre físico	Tipo	Longitud	Tipo de llave/opción	Comentario
idmateria	char	4	FK1, null	Número identificador de la materia
grupo	char	2	Null	Grupo de la materia
idprofesor	char	10	FK2, null	Número identificador del profesor

## Diagrama HIPO

Los perfiles Administrador, Docente y Usuario, tendrán accesos diferenciado a las opciones del sistema Web E-Voice.

- El perfil **Administrador** posee todos lo privilegios de acceso a la gestión de usuarios, grupos de trabajo, lectura de e-mail.
- El perfil **Docentes** posee los privilegios de acceso a la gestión de usuarios, grupos de trabajo, incluyendo grupos de alumnos, lectura de e-mail.
- El perfil **Usuarios** posee los privilegios de acceso a la gestión de usuarios, grupos de trabajo, lectura de e-mail, excepto a la creación de grupos de alumnos.

Diagrama HIPO incluyendo los tres diferentes privilegios de accesos



## ***Diseño de la Interfaz del Aplicativo Web E-VOICE***

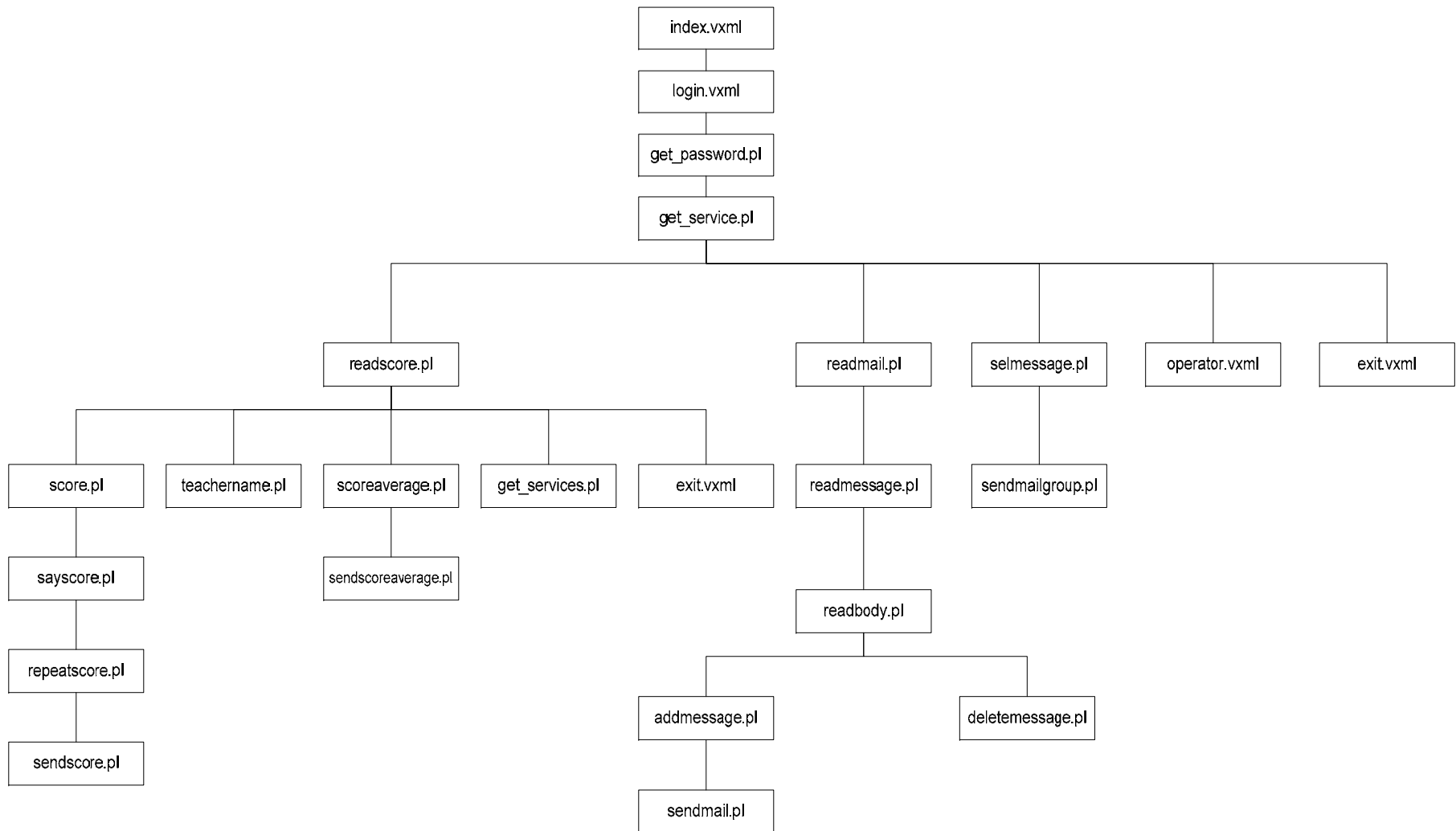
La interfaz Web del aplicativo Web, tendrá una estructura y organización basada en opciones.

Los elementos que integraran la interfaz son Banner, opciones de menú, <información de opciones>, detalle de opciones y el botón de ejecución.

- El banner, tendrá un pequeño diseño alusivo al servicio a desarrollar.
- Las opciones del menú, mostrara las diferentes opciones de menú a mostrar, dependiendo del usuario que ingreso.
- Información de opciones., muestra la información sobre el usuario, así como la información del servicio utilizado.
- Detalles de la opción, espacio diseñado para mostrar la información recuperada.
- Botón mostrara el texto de ejecución del programa.



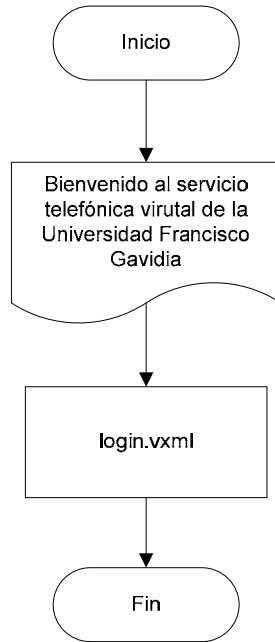
## Diagramas de flujo del programa Sistema E-Voice



## DIAGRAMAS DE FLUJO DEL SISTEMA E-VOICE

### DESCRIPCION:

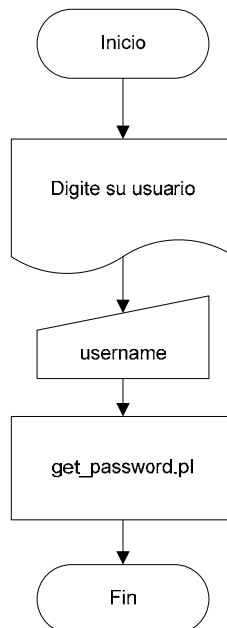
Este script, le da la bienvenida al sistema y llama a login.xml



### DIAGRAMA DE FLUJO - SCRIPT: login.vxml

#### DESCRIPCION:

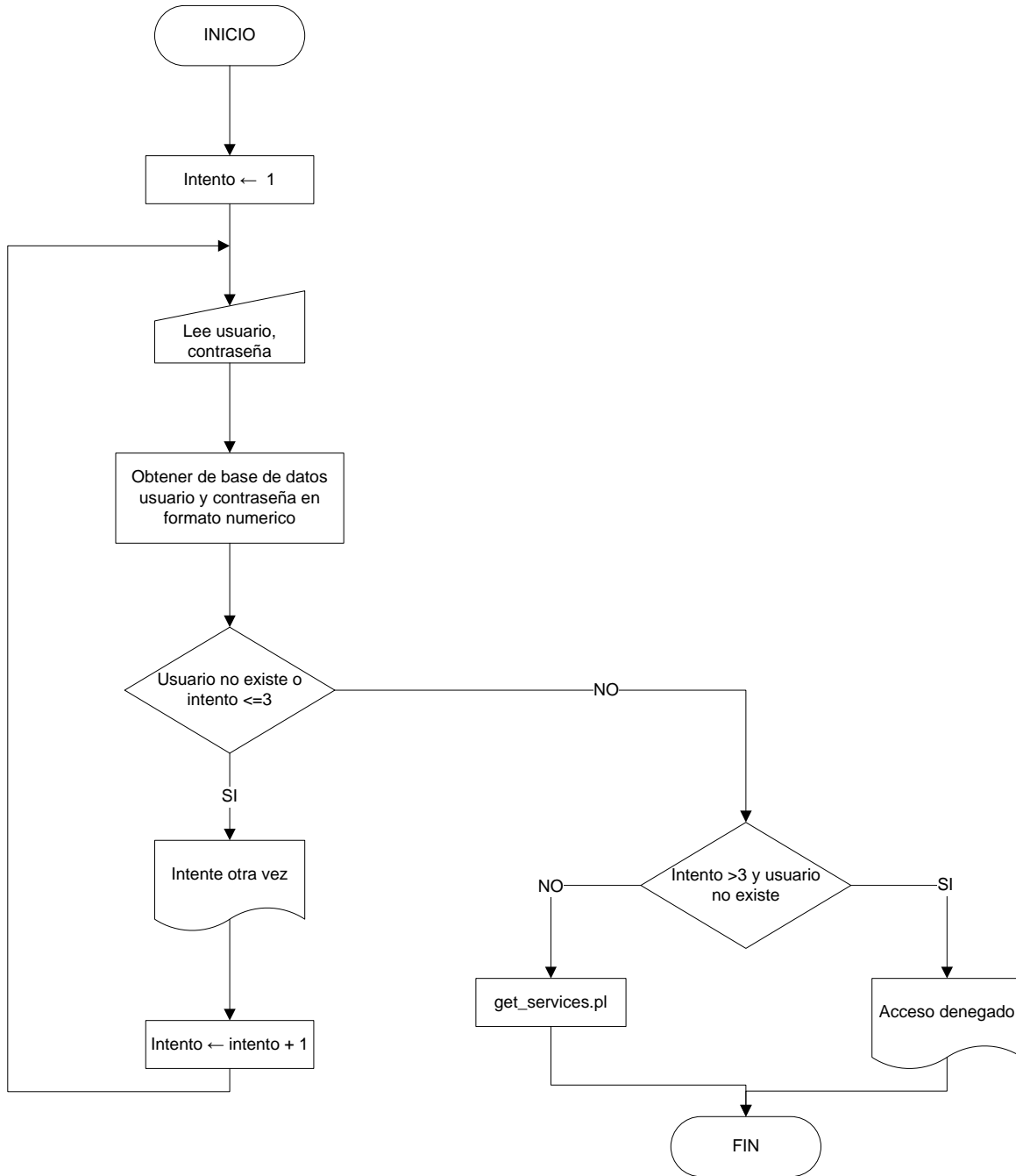
Este script que permite el ingreso al sistema.



## DIAGRAMA DE FLUJO - SCRIPT: getpassword.pl

### DESCRIPCION:

Script responsable de hacer la comparación entre el password ingresado por el usuario del sistema y el password almacenado en la base de datos.

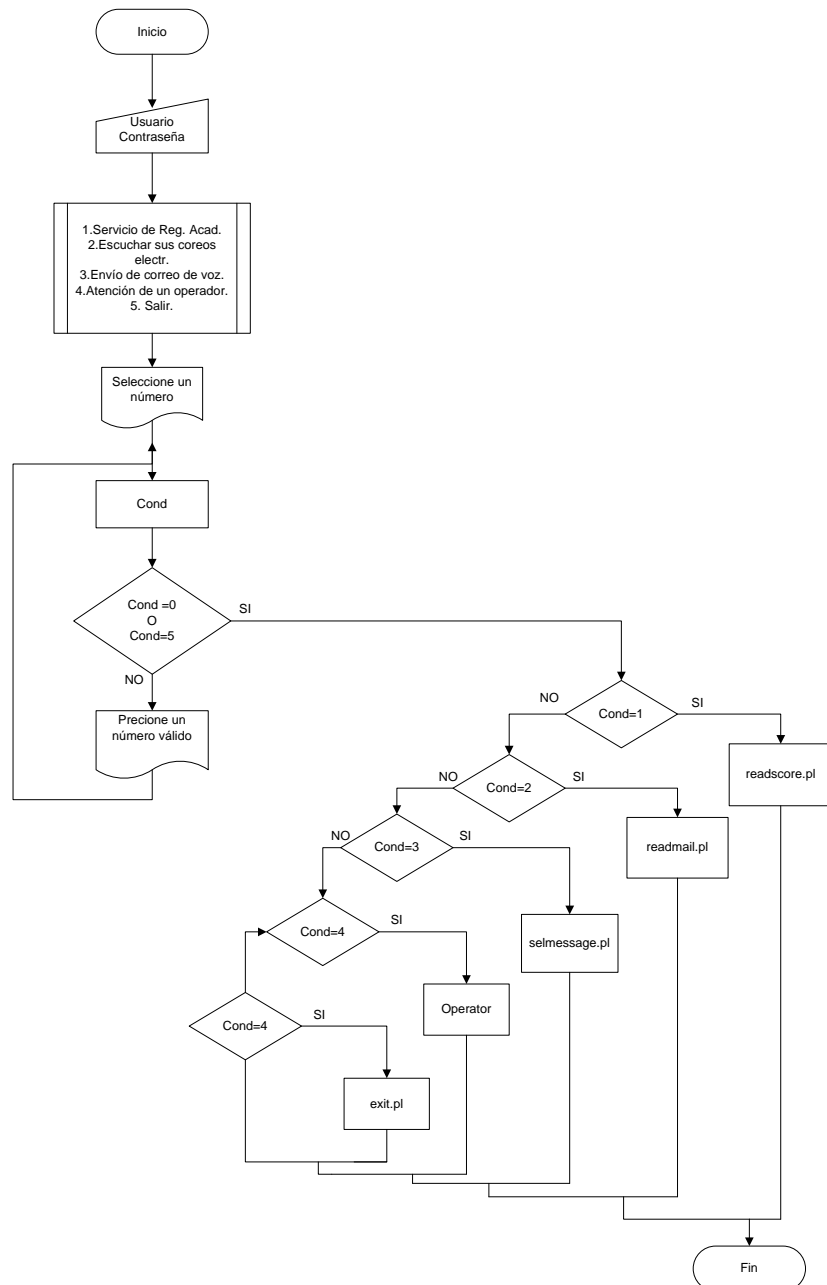


## DIAGRAMA DE FLUJO - SCRIPT: get\_services.pl

### DESCRIPCION:

Este script será el responsable de hacer la llamada de un procedimiento, dependiendo de la opción seleccionada. En el se resumen todas las opciones a las cuales se puede acceder al sistema, tal y como son:

1. Servicio de Registro Académico del estudiante.
2. Escuchar sus correos electrónicos.
3. Envío de correo de voz.
4. Atención de un operador.
5. Salir del sistema.

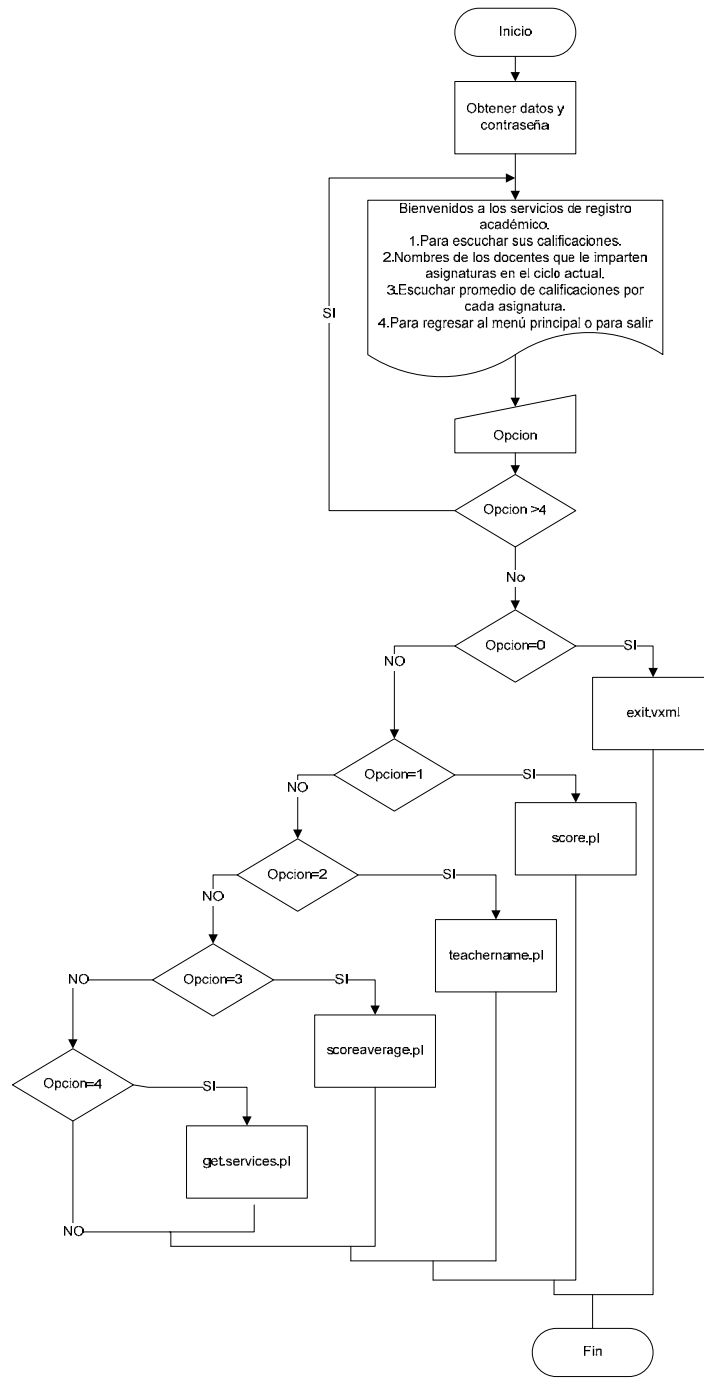




## DIAGRAMA DE FLUJO - SCRIPT: readscore.pl

### DESCRIPCION:

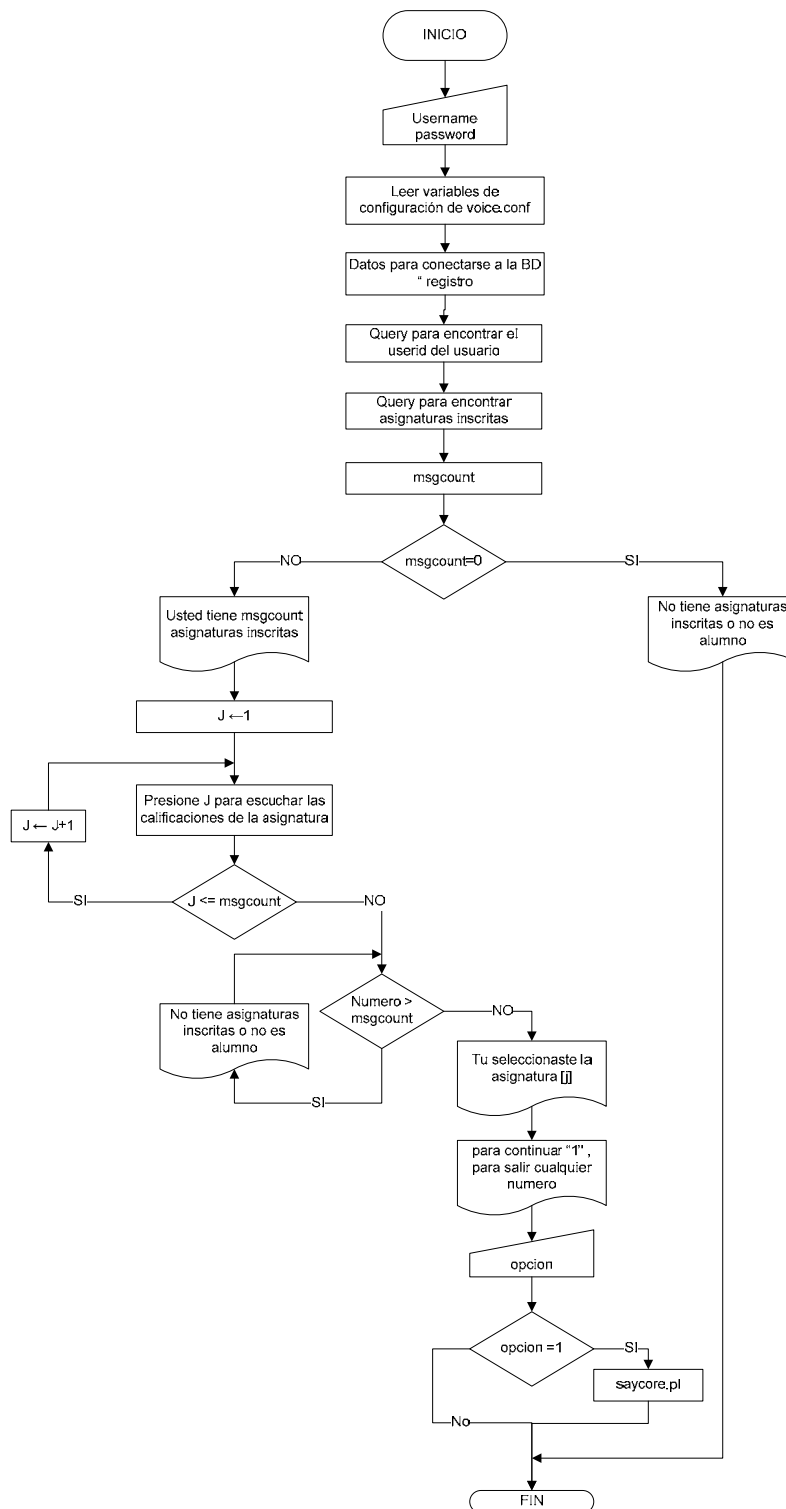
Este script será el responsable de la lectura de las calificaciones de un usuario obtenidas en el presente ciclo, así como también hará la lectura de los docentes que le imparten sus asignaturas y también será responsable de leerle la calificación final por asignatura inscrita, esto se hará a través de llamadas de procedimientos (scripts).



## DIAGRAMA DE FLUJO - SCRIPT: score.pl

### DESCRIPCION:

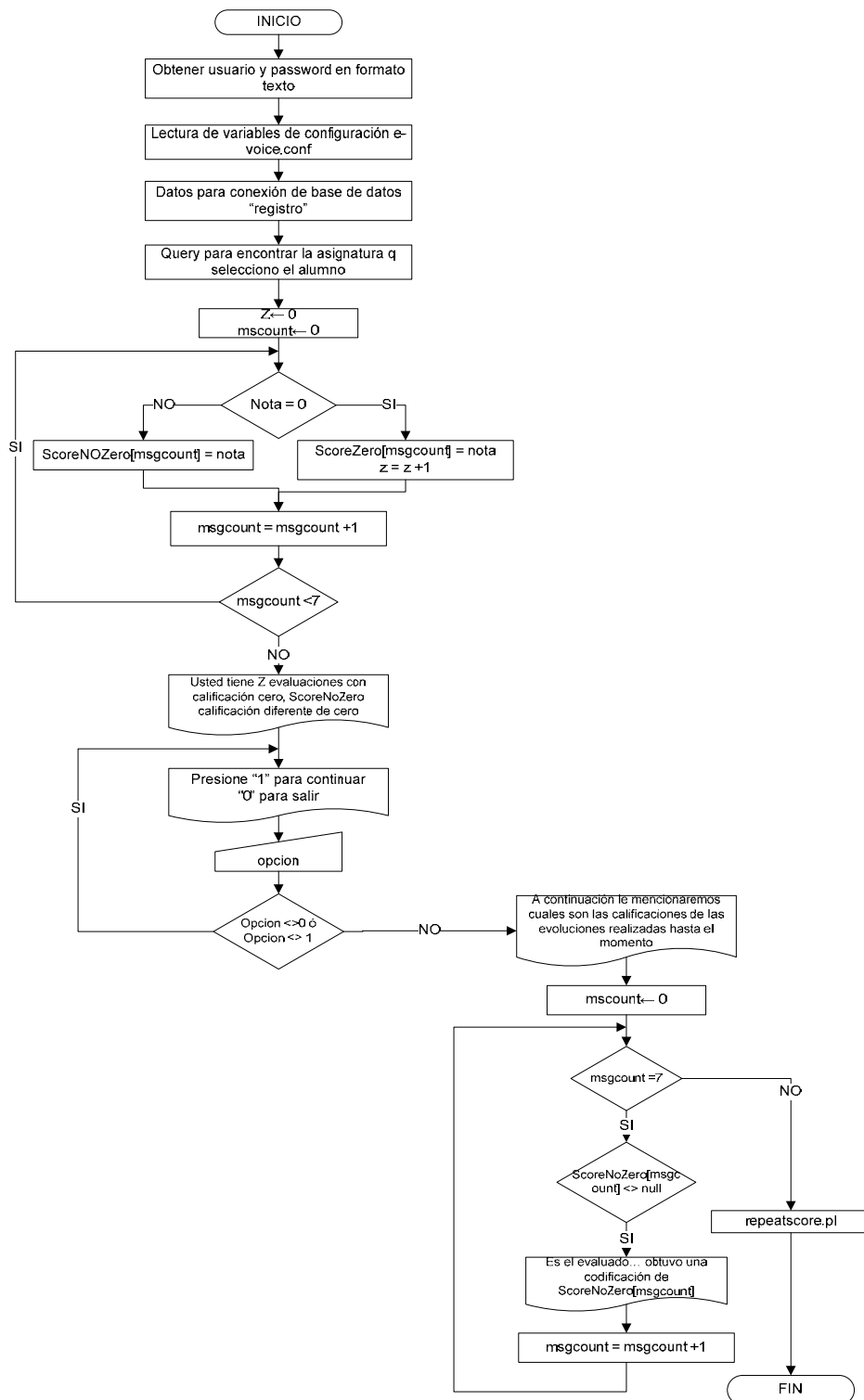
Script responsable de la lectura de las calificaciones de las asignaturas inscritas en el ciclo actual realizadas por un estudiante.



## DIAGRAMA DE FLUJO - SCRIPT: sayscore.pl

### DESCRIPCION:

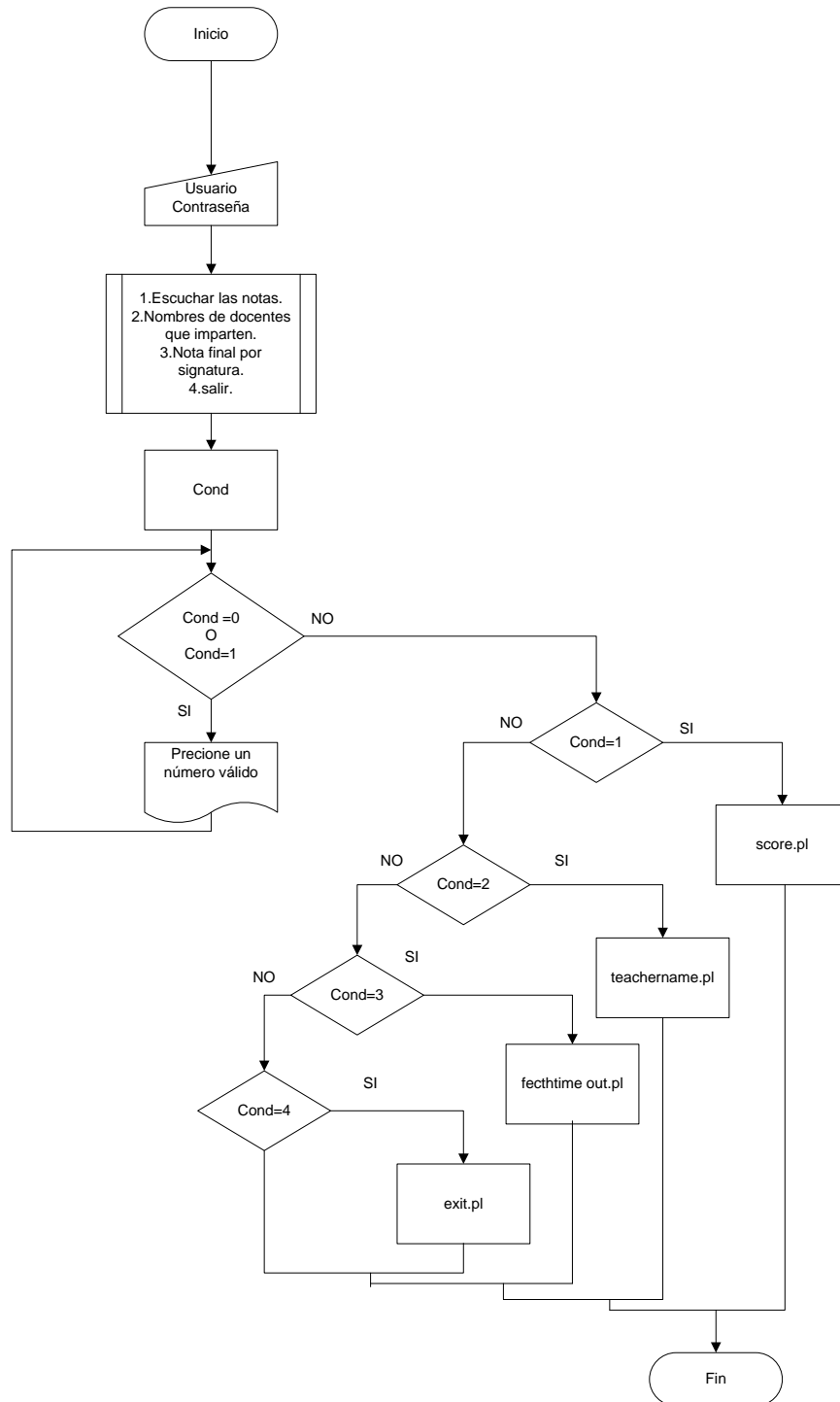
Este es responsable de la lectura por voz, de las calificaciones del ciclo actual y que tienen un valor diferente de cero.



## DIAGRAMA DE FLUJO - SCRIPT: repeatscore.pl

### DESCRIPCION:

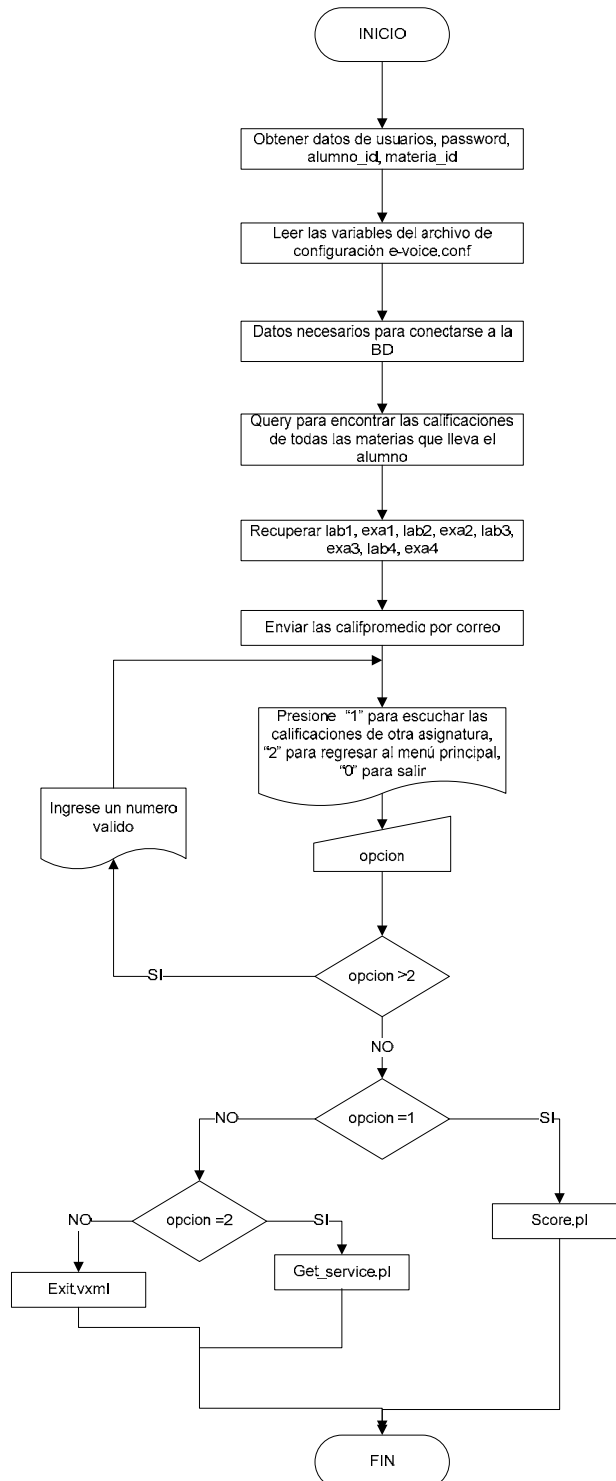
Responsable de escuchar las calificaciones de las asignaturas inscritas en el presente ciclo, adicionalmente le permite al usuario escucharlas nuevamente, enviárselas a su correo, regresar al menú principal o salirse del sistema.



## DIAGRAMA DE FLUJO - SCRIPT: sendscore.pl

### DESCRIPCION:

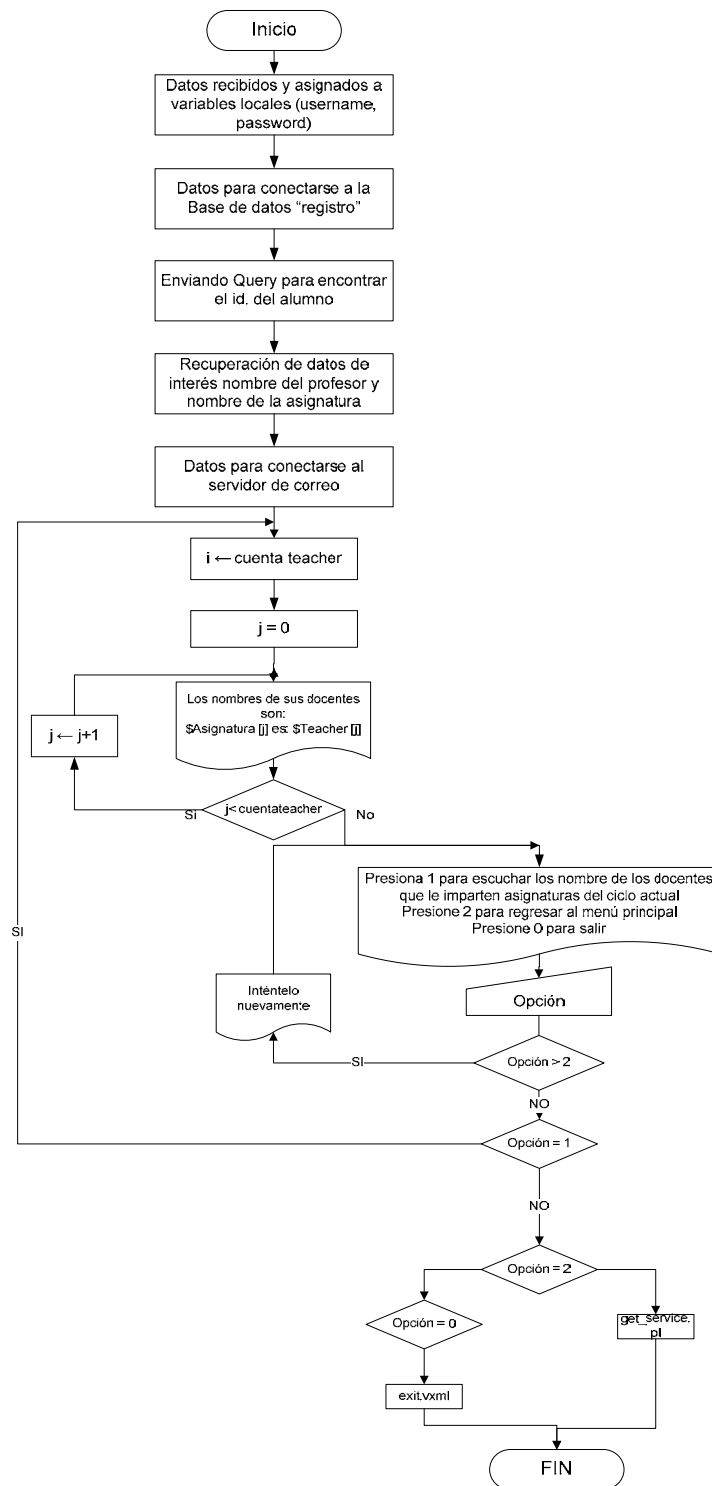
Responsable de escuchar las calificaciones de las asignaturas inscritas en el presente ciclo, adicionalmente le permite al usuario escucharlas nuevamente, enviárselas a su correo, regresar al menú principal o salirse del sistema.



## DIAGRAMA DE FLUJO - SCRIPT: teachername.pl

### DESCRIPCION:

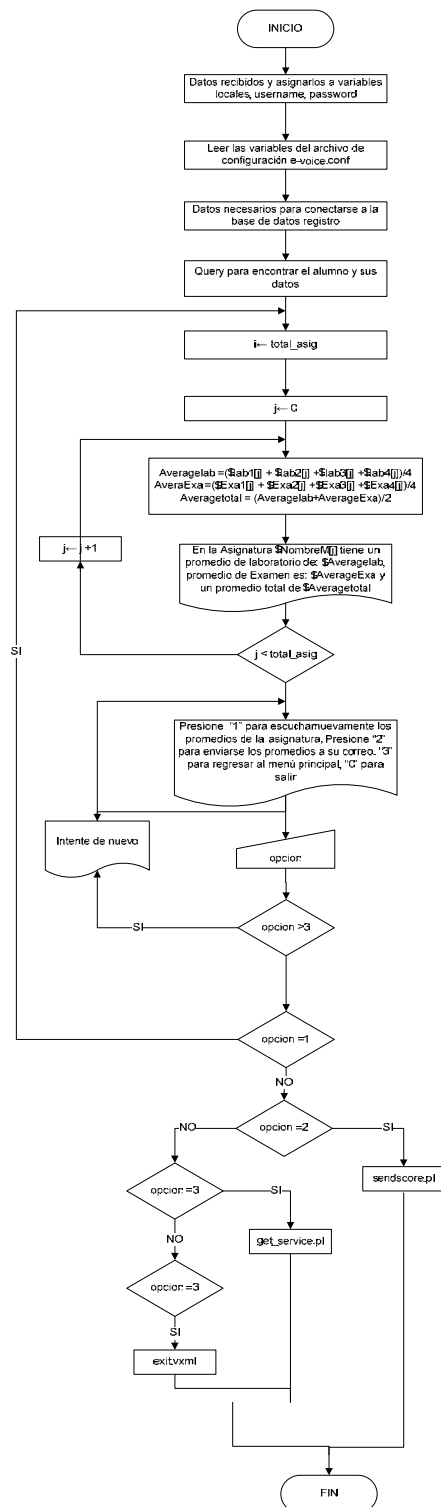
Script que tiene como responsabilidad la lectura de los docentes que le están impartiendo clases en las asignaturas inscritas en el presente ciclo.



## DIAGRAMA DE FLUJO - SCRIPT: scoreaverage.pl

### DESCRIPCION:

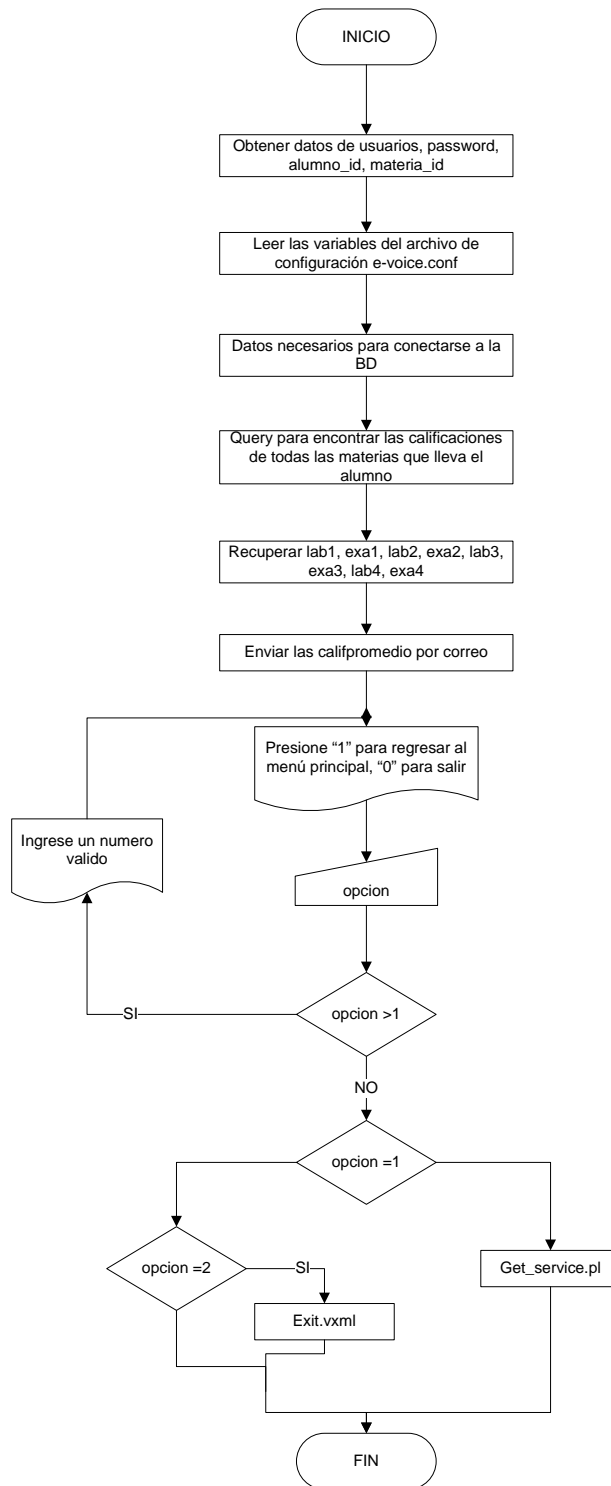
Script que tiene por objetivo promediar las calificaciones de los laboratorios, exámenes y calificación global del ciclo actual.



## DIAGRAMA DE FLUJO - SCRIPT: sendscoreaverage.pl

### DESCRIPCION:

Responsable de escuchar el promedio de las calificaciones de las asignaturas inscritas en el presente ciclo.

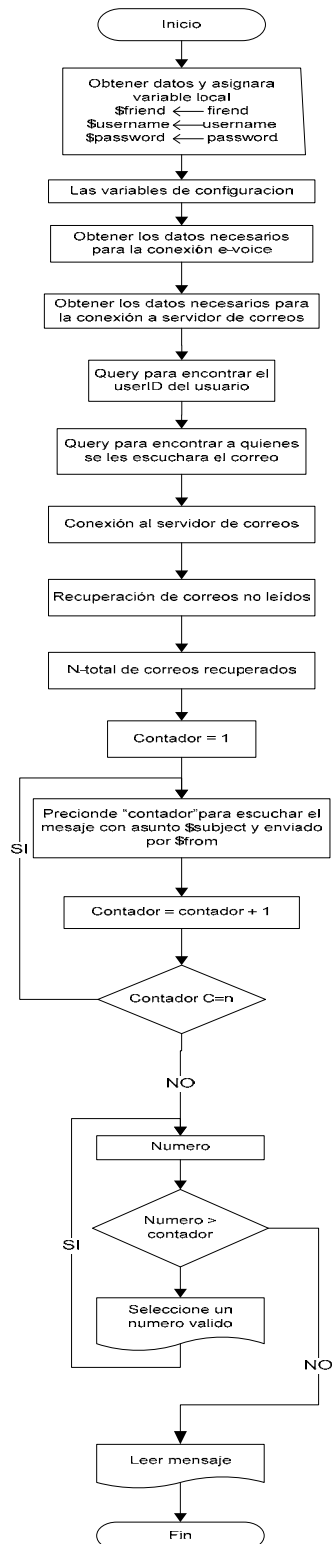




## DIAGRAMA DE FLUJO - SCRIPT: readmessage.pl

### DESCRIPCION:

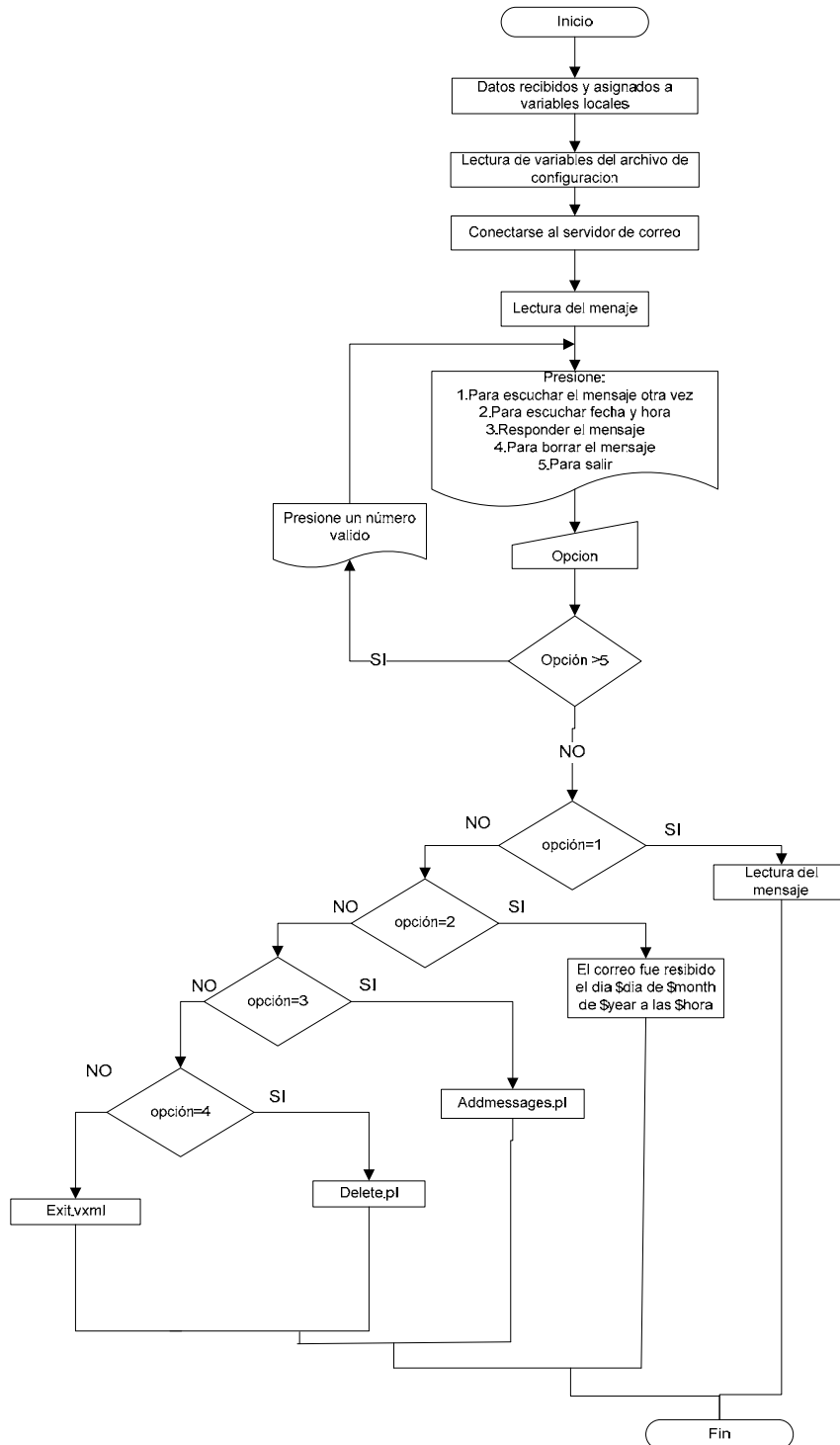
Script responsable de la lectura de los mensajes del usuario.



## DIAGRAMA DE FLUJO - SCRIPT: readbody.pl

### DESCRIPCION:

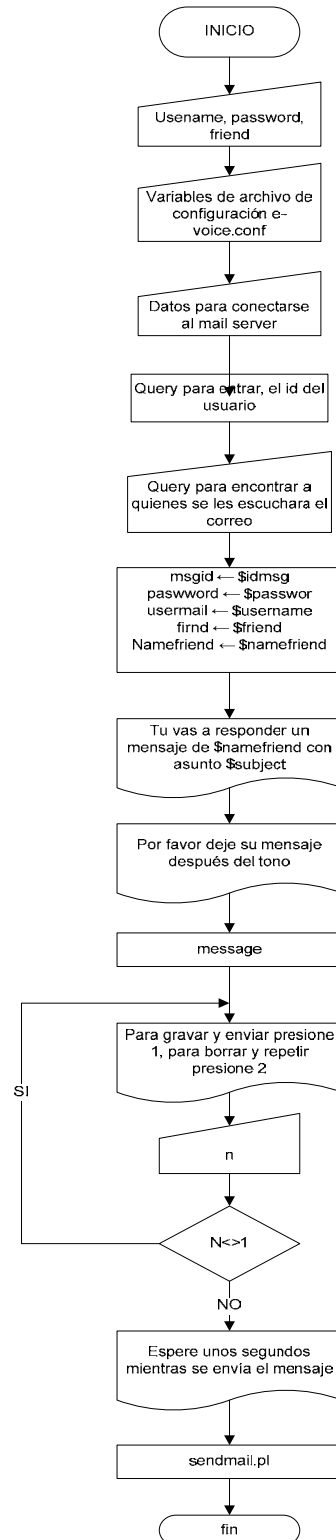
Responsable de leer el cuerpo de un correo.



## DIAGRAMA DE FLUJO - SCRIPT: addressmessages.pl

### DESCRIPCION:

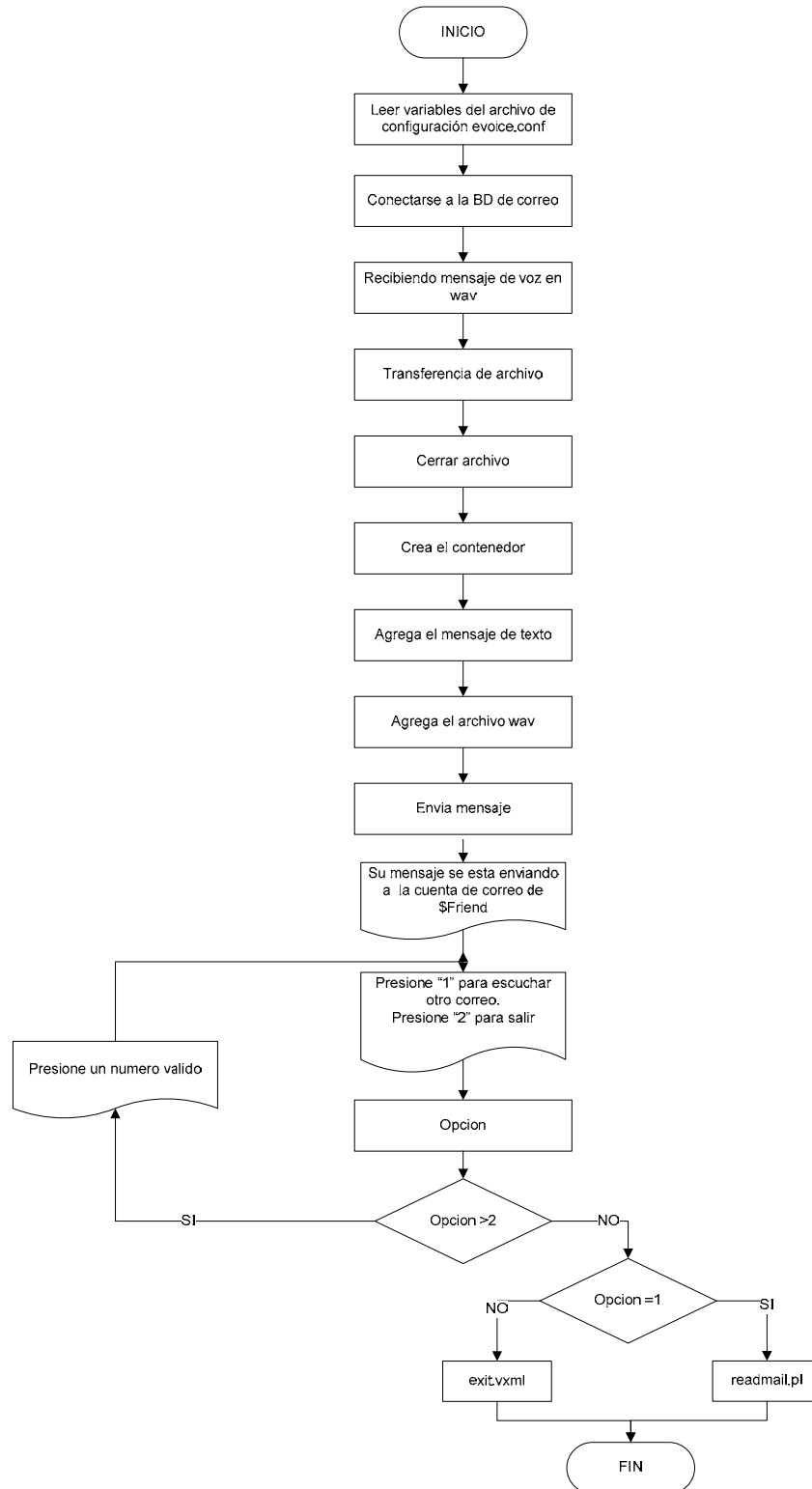
Script que le permitirá responder por voz a algún mensaje.



## DIAGRAMA DE FLUJO - SCRIPT: sendmail.pl

### DESCRIPCION:

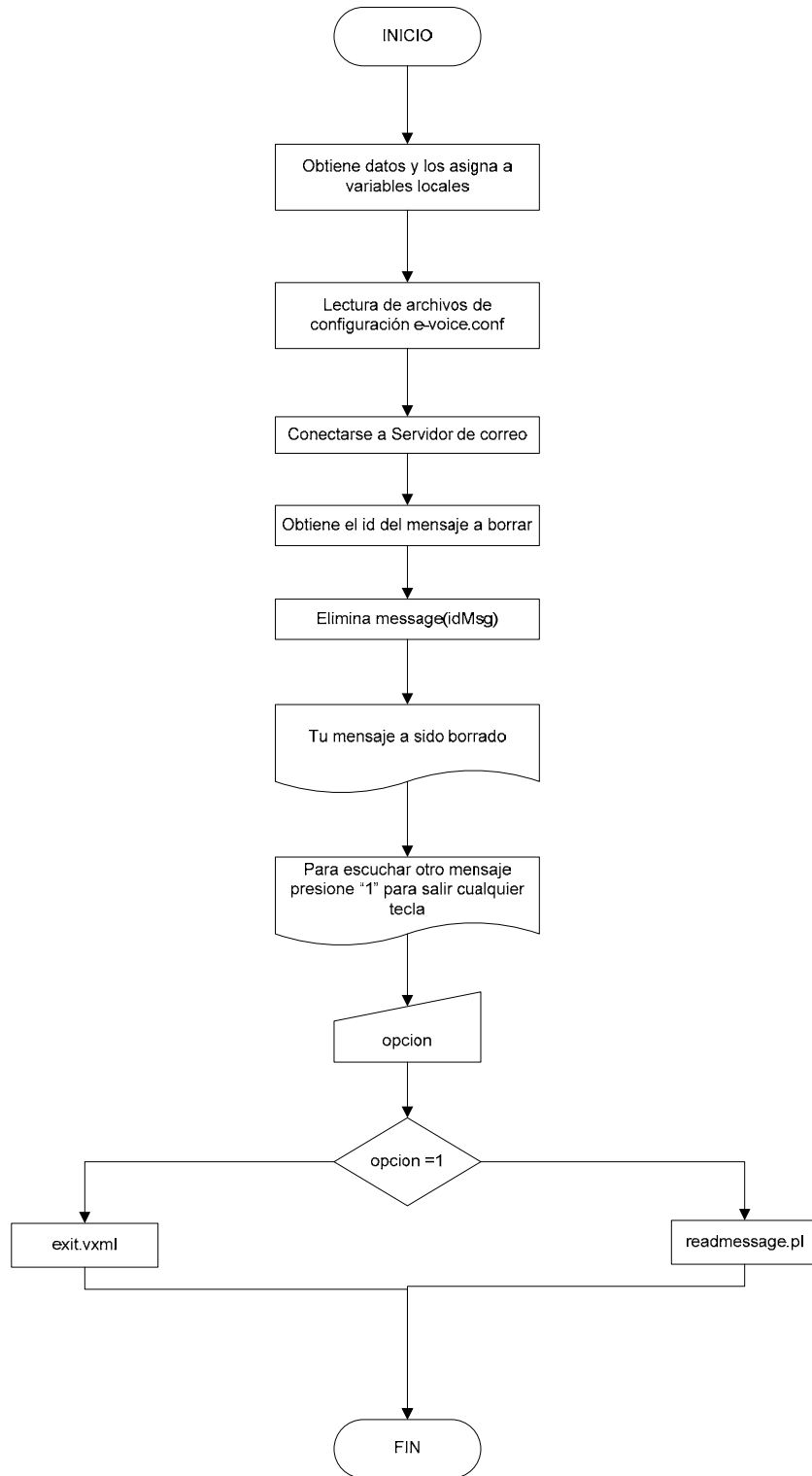
Script responsable del envío de un correo adjuntando el archivo de voz (formato wav).



## DIAGRAMA DE FLUJO - SCRIPT: deletemessage.pl

### DESCRIPCION:

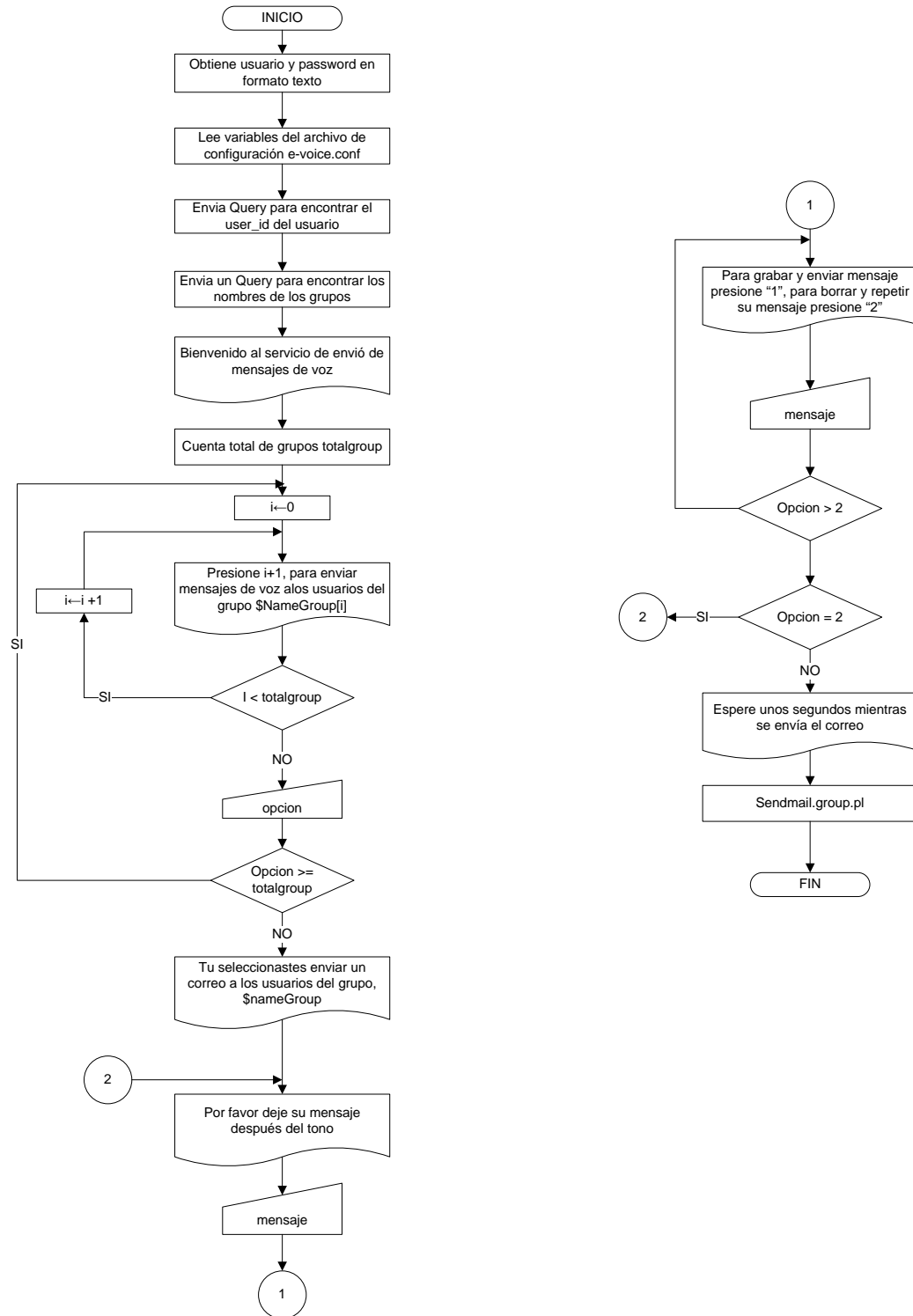
Script responsable de efectuar la eliminación de un mensaje, a petición del usuario.



## DIAGRAMA DE FLUJO - SCRIPT: selmessage.pl

### DESCRIPCION:

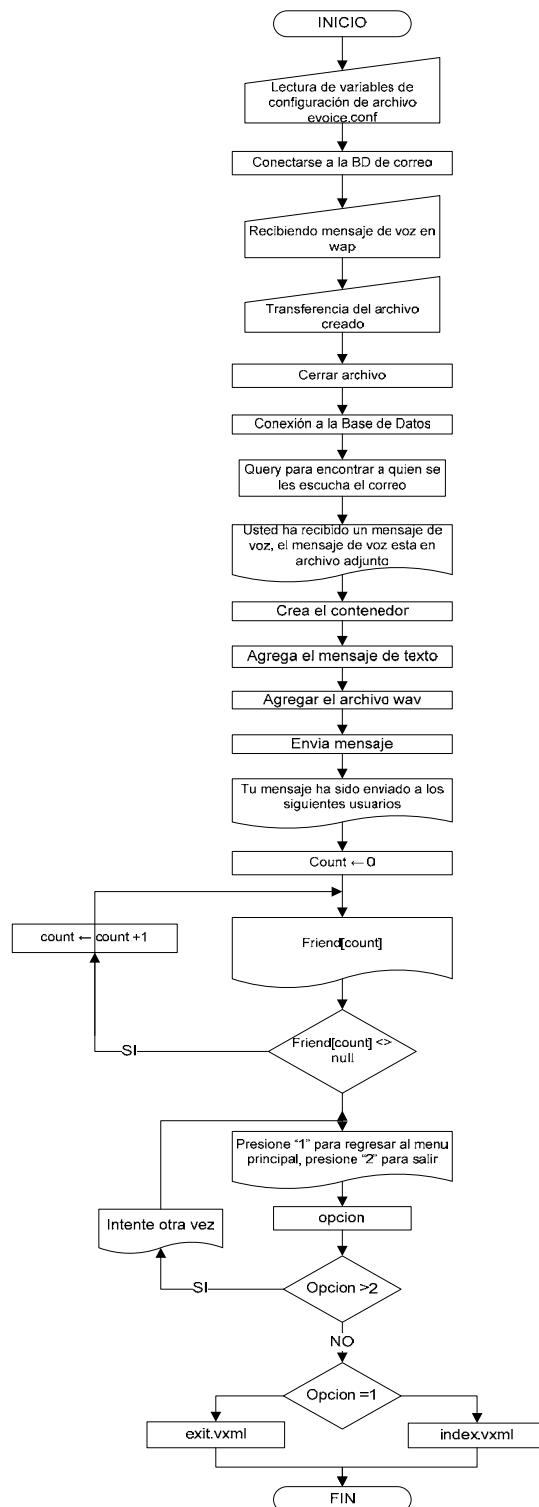
Script responsable de escuchar un correo, a petición del usuario.



## DIAGRAMA DE FLUJO - SCRIPT: sendmailgroup.pl

### DESCRIPCION:

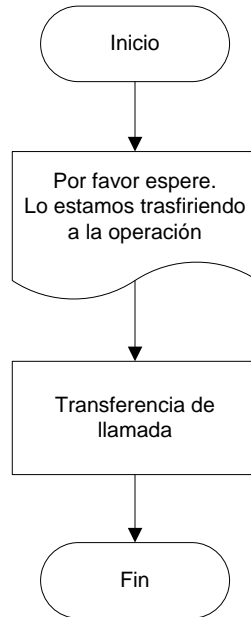
Script responsable del envío de un mensaje a un grupo de contactos conteniendo un archivo adjunto en formato wav.



## DIAGRAMA DE FLUJO - SCRIPT: operator.vxml

### DESCRIPCION:

Este script será el responsable de la transferencia de la llamada a la planta telefónica.





### 3. Construcción

## HERRAMIENTAS UTILIZADAS SOFTWARE

### **VXIAsterisk**

VXIAsterisk es un plugging para Asterisk PBX para Interpretar lenguaje VXML ( VoiceXML Browser). Basado en OpenVXI que es una librería OPenSource que interpreta dialogo de lenguaje de VoiceXML ([www.voicexml.org](http://www.voicexml.org)). Utilizando las especificaciones VoiceXML 2.0.

VXIAsterisk al igual que OpenVXI, es un componente de la plataforma completa VoiceXML, que incluye: Reconocimiento de Voz, capacidad de conversión de Texto a voz, y funciones de Telefonía.

OPenVXI fue desarrollado por SpeechWorks Internacional y mantenido por ScanSoft, ahora paso a manos de Vocalocity desde Agosto del 2004.

VXIAsterisk tiene un conector (script hecho en PHP) que es utilizado para la conexión entre el interprete VoiceXML y el TTS Engine.

### **Asterisk**

Es una aplicación de código abierto de una central telefónica (PBX); que puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una RDSI.

Asterisk incluye muchas características como: buzón de voz, conferencias, IVR, distribución automática de llamadas, y otras muchas más. Los usuarios pueden crear nuevas funcionalidades escribiendo un *dialplan* en el lenguaje de script de Asterisk o añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación soportado por Linux.

En nuestro proyecto se utilizó asterisk como un Gateway entre la Red Telefónica tradicional (PSTN), y la Aplicación VXIAsterisk es nuestro interprete utilizado para el Lenguaje VoiceXML.

### **Texto a Voz Cepstral (Engine TTS para Idioma Español)**

Cepstral text to speech (TTS) es un motor que convierte documento en formato texto a voz de alta calidad, para aplicaciones de avanzadas en las áreas de: Telefonía, Medicina, Industria, Navegación, etc.

Actualmente el motor TTS de Cepstral soporta 5 Idiomas. Para nuestra aplicación se compró una voz para Idioma Español llamada "Marta".

### **Texto a Voz Flite (Engine TTS para Idioma Inglés)**

Para el Caso del Reconocimiento de Texto en Inglés, se utiliza el motor Flite (una versión reducida de Festival TTS) con una voz gratuita desarrollada por la Universidad de Carnegie Mellon.

### **Dbaci**

Este programa de línea de comando se utiliza para averiguar en que lenguaje se encuentra el texto de los mensajes de correo recibidos y que se van a convertir a voz, esto con el objetivo de aplicar el motor TTS adecuado para su conversión. Actualmente la aplicación solo soporta español e inglés.

Dbaci es muy usado especialmente para clasificar muchos tipos de documento, cada categoría de documento se construye como modelo de lenguaje de Entropía máximo, con respecto a referencia basada en digrams (Parejas de caracteres).

Después de aprender, cualquier número de archivos de texto pueden estos ser utilizados como comparación de la nueva entrada en términos de probabilidades Bayesiana.

### **Apache**

El **servidor HTTP Apache** es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

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

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo y creciendo aún

su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

## **Perl**

**Perl, Lenguaje Práctico para la Extracción e Informe** es un lenguaje de programación diseñado por Larry Wall creado en 1987. Perl toma características del C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, muchos otros lenguajes de programación.

Estructuralmente, Perl está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

## **Php**

**PHP** es un lenguaje de programación usado frecuentemente para la creación de contenido para sitios Web con los cuales se puede programar las páginas html y los códigos de fuente. PHP es un acrónimo recursivo que significa "**PHP Hypertext Pre-processor**" (inicialmente PHP Tools, o, *Personal Home Page Tools*), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+.

## **HARDWARE.**

### **Tarjeta Digium TDM400P**

La Wildcard Digium TDM400P es una tarjeta mini-PCI 2.2 estándar configurable utilizando módulos FXO y FXS permitiendo conectar hasta 4 teléfonos análogos (o líneas RTB), cada una de estas tarjetas tiene capacidad hasta de cuatro módulos. Para el caso de nuestro proyecto utilizaremos 2 módulos FXO (TDM04B) para la tarjeta. Si se quiere aumentar la capacidad a 8 puertos es necesario adquirir 2 Tarjetas con sus respectivos 4 Puertos FXO.

## 4. Instalación

### ***Instalar Asterisk desde los archivos fuentes***

Obtener los archivos Fuentes del siguiente url: <http://www.asterisk.org/download>

Descomprimir el archivo:

```
# cd /root/src  
# tar zxvf asterisk-*.tar.gz
```

Posteriormente compilar los archivos, ejecutando los siguientes comandos:

```
# cd /root/src/asterisk-version  
# make clean  
# make  
# make install  
# make samples
```

Make samples se debe de ejecutar si se necesita que se generen archivos de configuración de ejemplos.

Si se esta usando un sistema que hace uso de los directorios /etc/init.d, se puede ejecutar el comando "make config" para instalar los script de inicio y configuración del sistema.

### ***Chequear la configuración de Asterisk***

Inicie Asterisk utilizando el siguiente comando:

```
# asterisk -cvvvvv
```

Si todo va bien, ejecute el siguiente comando para verificar la versión del Asterisk instalado, esto servirá par saber que versión de VXIAsterisk se deberá instalar.

```
CLI*> show versión
```

### ***Instalar VXI\****

Utilizar el usuario root para Instalar VXI\*.

Descomprima el paquete openvxi utilizando el siguiente comando:

```
# tar xvzf openvxi_V1.1_date.tar.gz
```

Vaya al directorio de openVxi y ejecute el comando:

```
# cd openvxi_V1.1_date  
# ./install.sh
```

#### Nota Importante:

VXIasterisk actualmente usa nuevas librerías y versiones de compilador. Antes de Instalar el VoiceXML Browser cheque su sistema Operativo para que tenga las librerías de GCC3 o GCC4:

VXI\* 1.4 es compilado con GCC3.3 / libstdc++5

VXI\* 1.5 es compilado con SARGE = GCC3.3 / libstdc++5 or ETCH = GCC4 / libstdc++6

VXI\* 2.0 es compilado con SARGE = GCC3.3 / libstdc++5 or ETCH = GCC4 / libstdc++6

#### *Chequear la configuración de VXI\**

Si todo salio bien. Se tuvo que haber agregado la aplicación (plugin) a Asterisk. Se puede chequear esto, simplemente reiniciando asterisk y ejecutando para mostrar ayuda de la aplicación Vxml.

```
# asterisk -r
```

```
CLI*> help vxml
```

Para mostrar el estatus de las licencia, use el siguiente comando:

```
CLI*> vxml show license
```

#### *Inicio y parado del demonio VXI\**

El programa VoiceXML browser se instala en /usr/sbin y /usr/lib/openvxi. El Script de inicio es /etc/init.d/openvxi. El Script openvxi llama a /usr/sbin/safe\_openvxi que funciona como un monitor y de auto-carga para el sistema VXI\*. Este Script safe\_openvxi inicializa el VoiceXML browser y lo monitorea para asegurarse que siempre este en ejecución. Si el proceso del VoiceXML browser muere, este Script intenta reiniciarlo.

Los comandos para iniciar el proceso de VXI\* es:

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

Si Asterisk no esta corriendo, debes de reiniciarlo antes con el siguiente comando:

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

Para para el proceso (o demonio), ejecutar el siguiente comando:

```
# /etc/init.d/openvxi stop
```

### *Línea de Comando para VXI\**

Ahora que el VXI\* VoiceXML browser y Asterisk esta corriendo, tu puedes observar algunas características del plugging VXML usando el CLI de Asterisk. Algunos comandos son los siguientes:

```
CLI*> help vxml
CLI*> vxml debug
CLI*> vxml no debug
CLI*> vxml reload
CLI*> vxml show license
```

### **Configure VoiceXML service**

Para configurar el servicio VoiceXML dentro de Asterisk, se debe de configurar el siguiente DialPlan (en el archivo extensions.conf):

```
# cd /etc/asterisk
# vi extensions.conf
```

Hay que agregar las siguientes líneas en dicho archivo.

```
exten => 888,1,Answer
exten => 888,2,Wait(3)
exten => 888,3,Vxml(http://vxml.i6net.com/index.vxml)
exten => 888,4,Hangup
```

Para en nuestro caso se seleccionaron las siguientes líneas:

```
exten => s,1,Answer
exten => s,2,Wait(3)
exten => s,3,Vxml(http://localhost/index.vxml)
exten => s,4,Hangup
```

Luego hay que asegurarse que los nuevos cambios hechos en este archive extension.conf, surtan efecto.. para ello hay que ejecutar el siguiente comando dentro de CLI de Asterisk.

```
CLI*> extensions reload
```

## **Configuración del Text-to-Speech**

El VXI\* VoiceXML browser integra un cliente HTTP para conectarse al Servidor HTTP Text-To-Speech. Esto permite generar dinámicamente el contenido de audio utilizando un motor TTS. Muchos VoiceXML browsers tienen un conector MRCP para acceder a algunos motores TTS. La ventaja de usar un conector HTTP es que la voz generada (speech) es cached. Algunos engine TTS que se pueden utilizar son:

- Flite TTS
- Cepstral TTS
- Loquendo TTS

Otros motores TTS también se pueden conectar.

### Nota Importante:

TTS necesita un servidor HTTP/Web con soporte del lenguaje PHP en la misma maquina en donde esta Asterisk y VXI\* instalado o en otro servidor.

En el archivo de configuración `/etc/openvxi/client.cfg` de VXI\*, se define la ubicación del mencionado conector HTTP.

```
#####  
# TTS server configuration #  
#####  
#client.prompt.resource.0.uri VXIString http://209.124.109.6/tts.php  
client.prompt.resource.0.method VXIString POST  
client.prompt.resource.0.cacheDir VXIString /tmp  
client.prompt.resource.0.format VXIString wav  
client.prompt.resource.0.maxage VXIInteger -1
```

En la líneas anteriores se observa que el conector `tts.php` se encuentra ubicado en el servidor con dirección IP `209.124.109.6`, que además es donde se encuentran los motores TTS: `switft` y `flite`.

A continuación se muestra un ejemplo del mencionado conector HTTP.

```
<?php  
  
$format = $_POST["format"];  
if (!$format)  
    $format = $_GET["format"];  
if (!$format)  
    $format = "wav"; // Default format  
    $text = $_POST["text"];  
if (!$text)
```

```

    $text = $_GET["text"];
if ($format=="gsm")
    {
    header("Content-Type: audio/x-gsm");
    header('Content-Disposition: attachment; filename="file.gsm"');
    }
else
    {
    header("Content-Type: audio/x-wav");
    header('Content-Disposition: attachment; filename="file.wav"');
    }
/*
// calc an offset of 24 hours
$offset = 3600 * 24;
// calc the string in GMT not localtime and add the offset
$expire = "Expires: " . gmdate("D, d M Y H:i:s", time() + $offset) . "GMT";
//output the HTTP header

Header($expire);
*/
if (($text) || ($_FILES['file']['tmp_name']))
    {
    if (($language == "text") && ($format == "gif"))
        {
        $text=str_replace("\", \"\", $text);
        $program = "convert";
        chdir("/tmp");
        @unlink("/tmp/text.gif");
        system("/usr/bin/convert -size 176x144 xc:transparent -fill black -pointsize 15 -
draw \"text 5,20 '$text'\" /tmp/text.gif")
        ;
        $filename="/tmp/text.gif";
        }
    else
    {
        $filename="/tmp/text.txt";
        $file=fopen($filename, "w");
        fwrite($file, $text);
        fwrite($file, "\n");
        fclose($file);
        $program_gramar="sudo /usr/local/bin/gramar.exe";
        $result = shell_exec($program_gramar." /tmp/text.txt");
        if (trim($result) == "espanol") {
        $program = "/opt/swift/bin/swift";
        exec($program." -f /tmp/text.txt -o /tmp/text.wav -p
audio/encoding=pcm16,audio/sampling-rate=8000,audio/channels=1", $return,$status);
        $filename="/tmp/text.wav";
    }
}

```



```
    }
    else {
        $program = "/usr/bin/flite";
        exec($program." -f /tmp/text.txt -o /tmp/text.wav", $return,$status);
        $filename="/tmp/text.wav";
    }
    }
    readfile($filename);
    unlink($filename);
}
else
{
header("HTTP/1.1 404 Not Found");
exit();
}
?>
```

## BIBLIOGRAFÍA

Digium, Inc., Digium Wildcard TDM400p, Empresa [en Línea], [citado 20 de Noviembre de 2006]. Disponible en <<http://www.digium.com/en/products/hardware/tdm400p.php>>.

Dreamweaver, software par diseño web, Empresa [en Línea], [citado 4 de Diciembre de 2006]. Disponible en <<http://www.adobe.com/es/products/dreamweaver/>>

Voice Extensible Markup Language (VoiceXML) Versión 2.0, Estándar [en Línea], [citado 10 de Noviembre de 2006]. Disponible en <<http://www.w3.org/TR/voicexml20/>>

The VoiceXML Forum, Organización [en Línea], [citado 22 de Noviembre de 2006]. Disponible en <<http://www.voicexml.org/>>

Asterisk, Plataforma Telefónica [en Línea], [citado 10 de Diciembre de 2006]. Disponible en <<http://www.asterisk.org/>>

OpenVXI, Interprete de VoiceXML [en Línea], [citado 13 de Diciembre de 2006]. Disponible en <<http://www.openvxi.org/>>

VXlasterisk, Interprete de VoiceXML [en Línea], [citado 13 de Diciembre de 2006]. Disponible en <<http://www.i6net.com/>>

DTMF, sistema multifrecuencial [en Línea], [citado 12 de febrero de 2006]. <<http://www.dialabc.com/sound/dtmf.html>>

SIGET, Superintendencia General de Electricidad y Telecomunicaciones, [citado 10 Enero de 2006]. Disponible en <<http://www.siget.gob.sv>>.

# **ANEXOS**



# eVoice

Servicio de Mensajería

**MANUAL DEL  
ADMINISTRADOR**

## GENERAL

	<b>Página</b>
1. Cómo visualizar el sistema de eVoice	1
2. Ingresando al sistema de eVoice	1
3. Pantalla de inicio del administrador (Opción: Gestión de usuario)	2
3.1 Adicionando un nuevo usuario	3
3.2 Modificando un registro de usuario o de administrador	3
3.3 Eliminando un registro de usuario o de administrador	4
3.4 Ingresando como otro usuario del sistema	4
4. Opción: Ver grupos de trabajo	5
4.1 Ver miembros de un grupo	6
4.2 Adicionar nuevo grupo	6
4.3 Crear nuevo grupo de alumnos	7
5. Opción: Usuarios para lectura de e-mail	8
6. Opción: Activación de usuario	9
7. Cómo salir del sistema de eVoice	10

## 1. Cómo visualizar el sistema de eVoice

1. Para ingresar al Sistema de Mensajería eVoice, en cualquier navegador de Internet, debe digitar en la barra de direcciones (ver Imagen No. 1) la siguiente URL: <http://209.124.109.7/app>.



Imagen No. 1

2. Luego aparecerá la pantalla que se observa en la Imagen No. 2



Imagen No. 2

## 2. Ingresando al Sistema de eVoice

1. Para ingresar al sistema debe digitar un **“Usuario”** y **“Clave”** de acceso válidos en los campos respectivos, como se observa en la Imagen No. 3, y luego debe hacer click sobre el botón **“Ingresar”** o presionar la tecla **“ENTER”** del teclado.



Imagen No. 3

### 3. Pantalla de inicio del administrador (Opción: Gestión de usuarios)

1. Luego de haber ingresado al sistema, aparecerá la pantalla de inicio del administrador (ver Imagen No. 4). En esta pantalla, pueden observarse 5 opciones principales: Gestión de usuarios, Ver grupos de trabajo, Usuarios para lectura de e-mail, Notificación de correo y Salir (estas opciones serán detalladas posteriormente).



Imagen No. 4

2. También esta pantalla visualizará una tabla con la lista de todos los usuarios y administradores del sistema eVoice. Junto a cada uno de los registros, se observan las opciones: "Modificar", "Eliminar" y "Entrar como" (ver Imagen No. 5).



Imagen No. 5

3. Otra opción importante, que esta disponible en esta pantalla es la de "Adicionar nuevo usuario" (ver Imagen No. 6).

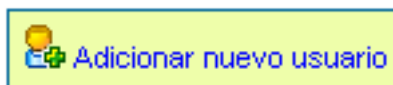


Imagen No. 6

### 3.1 Adicionando un nuevo usuario

1. Para adicionar un registro de usuario o de administrador, debe hacerse click sobre el enlace “Adicionar nuevo usuario” que se observa en la Imagen No. 6 y luego se mostrará una nueva pantalla con el formulario de Adición de nuevo usuario (ver Imagen No. 7)



Imagen No. 7

2. Si el registro a adicionar NO ES un docente, debe ingresar un e-mail en el campo respectivo, **el campo “ID de docente” no será visible**, luego debe elegir el tipo de usuario, escribir una breve descripción y hacer click en el botón “**Enviar**”. El registro adicionado aparecerá automáticamente en la lista de usuarios y administradores del sistema (remitirse a Imagen No. 4)
3. Si el registro a adicionar SI ES un docente, seleccione la opción “Docente”, luego debe ingresar el e-mail, seguido de un ID de docente válido, seleccionar el “Tipo de usuario”, escribir una breve descripción y hacer click en el botón “**Enviar**”. El registro adicionado aparecerá automáticamente en la lista de usuarios y administradores del sistema (remitirse a Imagen No. 4)

### 3.2 Modificando un registro de usuario o de administrador

1. Para modificar un registro de usuario o de administrador, debe hacerse click sobre el enlace “Modificar” que se observa en la Imagen No. 5 y luego se



mostrará una nueva pantalla con el formulario de Modificación de registro (ver Imagen No. 8)

The image shows a web interface for the 'Servicio de Información eVoice' of UFG. The main heading is 'Bienvenido al Servicio de Información eVoice'. Below this, there is a navigation bar with links: 'Cambiar de usuarios', 'Ver grupos de trabajo', 'Usuarios para lectura de email', 'Notificación de correos', and 'Salir'. The central focus is a form titled 'MODIFICACIÓN DE REGISTRO'. The form fields are as follows:

Usuario	eplaten
Clave	<input type="password"/>
Confirma clave	<input type="password"/>
Nombre	Edgardo Antonio Platen
E-mail	eplaten@ufg.edu.uy
Descripción	Edgardo Antonio Platen
Tipo	Administrador
Celular	<input type="text"/>
Servicio	ningun servicio

At the bottom of the form is a button labeled 'Modificar', which is highlighted by a yellow arrow.

Imagen No. 8

2. Deberá modificarse los campos deseados, y luego hacer click en el botón "Modificar" para generar los cambios en el registro.

### 3.3 Eliminando un registro de usuario o de administrador

1. Para eliminar un registro de usuario o de administrador, debe hacerse click sobre el enlace "Eliminar" que se observa junto a cada registro. Aparecerá un mensaje donde se deberá confirmar la eliminación del registro o bien, optar por cancelar esa acción.

### 3.4 Ingresando como otro usuario del sistema

1. Para iniciar sesión como otro usuario del sistema, debe hacerse click sobre el enlace "Entrar como" (remitirse a imagen No. 5). Hecho lo anterior, la persona en sesión actual, estará logueado como el otro usuario sobre el que hizo click en la opción "Entrar como" y podrá ver la sesión de ese usuario (ver imagen No. 9).
2. Para salir de esa modalidad, se debe "Salir", es decir se debe cerrar sesión y volver a ingresar nuevamente.



Imagen No. 9

#### 4. Opción: Ver grupos de trabajo

1. Al hacer click en la opción “**Ver grupos de trabajo**”, aparecerá la pantalla que se observa en la Imagen No. 10 donde se visualizará la lista de grupos de personas creados por el administrador.
2. Excepto para los grupos que tengan el servicio de “**Lectura de e-mail**”, para cada grupo de la lista existen tres opciones: Ver grupo, Modificar y Eliminar.
  - a) **Ver miembros del grupo:** Opción que visualiza todos los nombres de individuos que forman parte de un grupo.
  - b) **Modificar:** Opción que permite modificar el nombre de grupo.
  - c) **Eliminar:** Opción con la que se elimina un grupo.



Imagen No. 10

#### 4.1 Ver miembros de un grupo

1. Al hacer click sobre la opción “Ver miembros de un grupo” (ver opción “A” en Imagen No. 10) se mostrará el listado de miembros del grupo.
2. Junto a cada miembro del grupo hay 3 opciones: **Modificar**, **Eliminar** y **Leer e-mail** (ver Imagen No. 11). Si se hace click sobre la opción “Leer e-mail”, el registro es copiado al listado de “**Usuarios para lectura de e-mail**”.



Imagen No. 11

#### 4.2 Adicionar nuevo grupo

1. Al hacer click sobre la opción “Adicionar nuevo grupo”, se carga una nueva pantalla (ver Imagen No. 12) donde podrá agregarse un nuevo nombre de grupo y seleccionar el tipo de servicio a establecerse.



Imagen No. 12

### 4.3 Crear nuevo grupo de alumnos

1. Cuando se hace click sobre la opción “**Crear nuevo grupo de alumnos**” (ver imagen No. 10), se visualizará la lista de materias con el respectivo número de grupo que un usuario-docente tiene a su cargo. (ver Imagen No. 13)



Imagen No. 13

2. Luego, al hacer click sobre alguna de las materias que aparecen en la lista, se desplegará una nueva pantalla donde se verá la lista de alumnos que están inscritos en esa materia (ver Imagen No. 14).



Imagen No. 14

3. Esta pantalla posee otra serie de opciones con las cuales se puede crear un nuevo grupo seleccionando alumnos que se visualizan en la lista de la materia, o bien, copiar alumnos a otro grupo ya existente.



## 5. Opción: Usuarios para lectura de e-mail

1. Haciendo click sobre la opción **“Usuarios para lectura de e-mail”** (ver Imagen No. 15) se mostrara la pantalla de la Imagen No. 16



Imagen No. 15

2. En esta pantalla se visualizará la lista global de todas las personas de las cuales un usuario puede recibir emails pudiéndose observar a qué grupo pertenece cada uno y, si en algún caso, ya no se desea recibir emails de una determinada persona, esta disponible la opción de **“Eliminar usuario del grupo”**.



Imagen No. 16

## 6. Opción: Notificación de correo

1. La imagen No. 17 indica la opción “Notificación de correo”, que al hacer click sobre ese enlace abrirá la ventana que se muestra en la imagen No. 18 donde un usuario del sistema eVoice puede solicitar la activación del tipo de servicio que él desee.



Imagen No. 17



Imagen No. 18

2. Debe completarse el formulario que se muestra en la imagen anterior, seleccionando el servicio que se solicita que se active y si se considera necesario deberá agregarse un comentario. Finalmente debe hacerse click sobre el botón “**Enviar mensaje**” y aparecerá un mensaje que indicará que el mensaje fue enviado correctamente.

## 7. Cómo salir del sistema de eVoice

1. Para cerrar la sesión del sistema, bastará con hacer click en la opción “**Cerrar sesión**” que aparece en todas las pantallas del sistema (ver Imagen No. 19). Y para salir por completo del sistema de eVoice, se debe hacer click sobre la opción “**Salir del sistema**” en la pantalla de ingreso (ver Imagen No. 20).



Imagen No. 19



Imagen No. 20



# eVoice

SERVICIO DE MENSAJERÍA

**MANUAL DEL USUARIO**



## INDICE GENERAL

	<b>Página</b>
1. Cómo visualizar el sistema de eVoice	1
2. Ingresando al sistema de eVoice	1
3. Pantalla de inicio del usuario	2
4. Opción: Ver grupos de trabajo	2
4.1 Ver miembros de un grupo	3
4.2 Adicionar nuevo grupo	3
4.3 Crear nuevo grupo de alumnos	4
5. Opción: Usuarios de los que puedo leer emails	5
6. Opción: Activación de servicio	6
7. Cómo salir del sistema de eVoice	7

## 1. Cómo visualizar el sistema de eVoice

1. Para ingresar al Sistema de Mensajería eVoice, en cualquier navegador de Internet, debe digitar en la barra de direcciones (ver Imagen No. 1) la siguiente URL: <http://209.124.109.7/app>.



Imagen No. 1

2. Luego aparecerá la pantalla que se observa en la Imagen No. 2



Imagen No. 2

## 2. Ingresando al Sistema de eVoice

1. Para ingresar al sistema debe digitar un **“Usuario”** y **“Clave”** de acceso válidos en los campos respectivos, como se observa en la Imagen No. 3, y luego debe hacer click sobre el botón **“Ingresar”** o presionar la tecla **“ENTER”** del teclado.

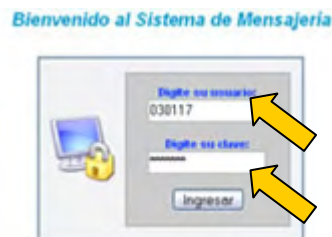


Imagen No. 3

### 3. Pantalla de inicio del usuario

1. Luego de haber ingresado al sistema, aparecerá la pantalla de inicio del usuario (ver Imagen No. 4), donde se observan 4 opciones: Ver grupos de trabajo, Usuarios para lectura de e-mail, Notificación de correo y la opción Salir.



Imagen No. 4

### 4. Opción: Ver grupos de trabajo

1. Con la opción “**Ver grupos de trabajo**”, se visualizará la lista de grupos de personas creados por un usuario. (ver Imagen No. 5)
  2. Para cada grupo de la lista existen tres opciones: Ver grupo, Modificar y Eliminar.
- d) **Ver miembros del grupo:** Opción que se visualiza todos los nombres de individuos que forman parte de un grupo.
- e) **Modificar:** Opción que permite modificar el nombre de grupo.
- f) **Eliminar:** Opción con la que se elimina un grupo.



Imagen No. 5

#### 4.1 Ver miembros de un grupo

1. Al hacer click sobre la opción “Ver miembros de un grupo” (ver opción “A” en Imagen No. 5) se mostrará el listado de miembros del grupo.
2. Junto a cada miembro del grupo hay 3 opciones: **Modificar**, **Eliminar** y **Leer e-mail** (ver Imagen No. 6). Si se hace click sobre la opción “Leer e-mail”, el registro es copiado al listado de “**Usuarios para lectura de e-mail**”.



Imagen No. 6

#### 4.2 Adicionar nuevo grupo

1. Al hacer click sobre la opción “Adicionar nuevo grupo”, se carga una nueva pantalla (ver Imagen No. 7) donde podrá agregarse un nuevo nombre de grupo y seleccionar el tipo de servicio a establecerse.



Imagen No. 7

### 4.3 Crear nuevo grupo de alumnos

1. Cuando se hace click sobre la opción “**Crear nuevo grupo de alumnos**” (ver Imagen No. 5) se visualizará la lista de materias con el respectivo número de grupo que un usuario-docente tiene a su cargo. (ver Imagen No. 8)



Imagen No. 8

2. Luego, al hacer click sobre alguna de las materias que aparecen en la lista, se desplegará una nueva pantalla donde se verá la lista de alumnos que están inscritos en esa materia (ver Imagen No. 9).



Imagen No. 9

3. Esta pantalla posee otra serie de opciones con las cuales se puede crear un nuevo grupo seleccionando alumnos que se visualizan en la lista de la materia, o bien, copiar alumnos a otro grupo ya existente.



## 5. Opción: Usuarios para lectura de e-mail

1. Haciendo click sobre la opción **“Usuarios para lectura de e-mail”** (ver Imagen No. 10) se cambia a la pantalla de la Imagen No.11



Imagen No. 10



Imagen No. 11

2. En esta pantalla se visualizará la lista global de todas las personas de las cuales un usuario puede recibir emails pudiéndose observar a qué grupo pertenece cada uno de los individuos y, si en algún caso, ya no se desea recibir emails de una determinada persona, esta disponible la opción de **“Eliminar usuario del grupo”**.

## 6. Opción: Notificación de correo

1. La imagen No. 12 indica la opción “Notificación de correo”, que al hacer click sobre ese enlace abrirá la ventana que se muestra en la imagen No. 13 donde un usuario del sistema eVoice puede solicitar la activación del tipo de servicio que él desee.



Imagen No. 12

A screenshot of a web browser window showing a form titled 'ACTIVACIÓN DE SERVICIO'. The browser address bar shows 'http://localhost - Mozilla Firefox'. The form contains the following fields: 'Su nombre:' with the value 'MARIO RAFAEL RUIZ'; 'Su e-mail:' with the value 'mario@utg.edu.sv'; 'Su número de celular:' (empty); 'Tipo de notificación:' with a dropdown menu set to 'Ningun servicio'; and 'Comentario:' (empty). At the bottom of the form are two buttons: 'Enviar mensaje' and 'Cerrar esta ventana'. The browser status bar at the bottom shows 'Terminado'.

Imagen No. 13

2. Debe completarse el formulario que se muestra en la imagen anterior, seleccionando el servicio que se solicita que se active y si se considera necesario deberá agregarse un comentario. Finalmente debe hacerse click sobre

el botón “**Enviar mensaje**” y aparecerá un mensaje que indicará que el mensaje fue enviado correctamente.

## 7. Cómo salir del sistema de eVoice

1. Para cerrar la sesión del sistema, bastará con hacer click en la opción “**Salir**” que aparece en todas las pantallas del sistema (ver Imagen No. 14). Y para salir por completo del sistema de eVoice, se debe hacer click sobre la opción “**Salir del sistema**” en la pantalla de ingreso (ver Imagen No. 15).



Imagen No. 14

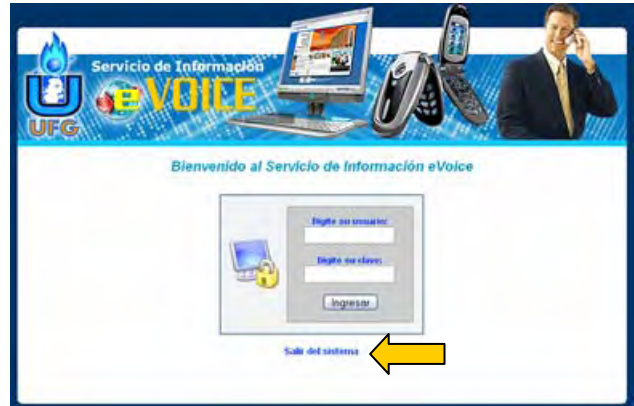


Imagen No. 15