Ingeniería Técnica en Informática de Gestión

PROYECTO FIN DE CARRERA

DESARROLLO DE UN SERVICIO TELEFÓNICO DE LECTURA DE NOTICIAS RSS MEDIANTE EL ESTÁNDAR VOICEXML

Autor: Víctor Gil Borrego

Tutor/Director: Dr. David Griol Barres

Leganés, julio de 2011

Título: DESARROLLO DE UN SERVICIO TELEFÓNICO DE LECTURA DE NOTICIAS RSS MEDIANTE EL ESTÁNDAR VOICEXML

Autor: Víctor Gil Borrego

Director: David Griol Barres

EL TRIBUNAL

Presidente: _____

Vocal:

_____

Secretario:

_____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

Me gustaría agradecer en estas líneas todo el apoyo que he recibido durante los años que he estado realizando esta carrera, y durante mi época de estudiante, a mis padres. Sin ellos, nada de esto hubiera sido posible, apoyándome en todo momento y dándome fuerzas en los momentos difíciles.

Quisiera agradecerle también a mi hermana por los buenos momentos que hemos pasado y que pasaremos.

De manera muy especial quiero agradecer todo el apoyo que he recibido día a día de mis amigos y de mis compañeros de universidad. Gracias a ellos todo ha parecido mucho más fácil, compartiendo momentos inolvidables en estos años de universidad.

A mis amigos de toda la vida que siempre han estado ahí para darme ánimos cuando más lo necesitaba.

Al tutor del presente proyecto David Griol por ofrecerme todos sus consejos y experiencia en la realización del trabajo.

Resumen

El presente Proyecto Final de Carrera tiene como principal objetivo describir una aplicación práctica del estándar VoiceXML para la implementación de un sistema de diálogo. Este tipo de aplicaciones permite la interacción con los que los usuarios utilizando una de las formas de comunicación más sencillas y natural, la voz.

Gracias a la comunicación mediante la voz conseguimos un gran abanico de aplicaciones prácticas para este tipo de sistemas, desde programas para la ayuda a personas con discapacidades visuales o motoras hasta aplicaciones que posibilitan el acceso a información o servicios en entornos en los que el uso de los interfaces tradicionales (teclado y ratón) impediría la utilización del sistema desarrollado.

Para el Proyecto hemos seleccionado como tarea práctica de nuestro sistema la realización de un servicio telefónico de lectura de noticias *RSS (Really Simple Syndication)*, formato XML ampliamente utilizado hoy en día para sindicar o compartir contenidos actualizados en la web. Para la extracción de la información se han utilizado diferentes fuentes de noticias generadas por periódicos y radios de ámbito nacional, que actualizan estos contenidos (*feeds*) casi constantemente.

La aplicación desarrollada refleja claramente los conceptos anteriormente mencionados sobre la utilidad de los sistemas de diálogo oral, en este caso, facilitando el acceso a las noticias diarias mediante el uso de la voz, ofreciendo un método sencillo y rápido de poderse mantenerse informado sobre la actualidad diaria.

Además, esta aplicación no sólo utiliza la tecnología VoiceXML, sino que también ha sido necesario el empleo de varios lenguajes adicionales, como son PHP y JavaScript, que posibilitan que la aplicación funcione de una manera dinámica y flexible. Para el mantenimiento actualizado de las noticias se ha diseñado una base de datos, creada mediante el lenguaje SQL a modo de repositorio auxiliar para la lectura de las noticias y cuyos contenidos son actualizados periódicamente por un módulo específico desarrollado para la aplicación.

El sistema desarrollado permite seleccionar la fuente de dónde obtener las noticias, las categorías correspondientes y temas populares. El usuario dispone en todo momento de las opciones de navegación para avanzar, ampliar o repetir, regresar a menús previos, etc. Por último, se ha incorporado además una funcionalidad que permite adaptar el funcionamiento del sistema a cada usuario teniendo en cuenta sus interacciones previas y recibiendo recomendaciones personalizadas en función de ellas.

Palabras Clave: Sistemas de Diálogo Hablado, Feeds RSS, VoiceXML, Interacción Oral.

Abstract

This Bachelor Project has the main objective of describing a practical application of the VoiceXML standard to implement a spoken dialogue system. This kind of applications allows interaction with the users using one of the simplest and more natural communication forms, speech communication.

Speech communication offers a number of practical applications of spoken dialog systems, from programs to help people with visual or motor disabilities to applications that allow the access to information or services in environments in which the use of traditional interfaces (keyboard and mouse) would avoid the use of the system.

For our system we have selected the task of developing a newsreader in the RSS standard (Really Simple Syndication), XML-based format now widely used to syndicate or share updated contents on the web. To extract the required information of information we have used different sources with RSS news generated by a set of Spanish newspapers and radio stations, which update these feeds almost constantly.

The system clearly reflects the concepts required to develop spoken dialogue systems, in this case, providing access to daily news through the use of speech, and offering a quick and easy method to be informed daily.

In addition, the system not only uses the VoiceXML technology, but also several additional languages such as PHP and JavaScript, which allow the application to work in a dynamic and flexible way. Databases technology and the SQL language have also been incorporated to keep the information updated and design an auxiliary repository for reading the news. The contents are regularly updated by a specific module developed for the application.

The developed system allows the selection of the source of the news, and provides relevant categories and popular topics. During the interaction, users are provided with the options required to select additional news, extend the information, repeat, back to previous menus, etc. In addition, the system incorporates personalization options achieved taking into account users' previous interactions to then receive customized recommendations based on them.

Keywords: Spoken Dialogue Systems, RSS Feeds, VoiceXML, Oral Interaction.

Índice

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción y Objetivos

1.1 Introducción

La Informática y los campos de estudio referentes a las Tecnologías del habla y el Procesamiento del Lenguaje Natural han sufrido numerosos cambios desde la aparición de los primeros ordenadores en los años 50 hasta los dispositivos móviles más modernos desarrollados durante los últimos años. La sociedad se ha ido adaptando a estos cambios, acostumbrándose al uso de este tipo de dispositivos de tal manera, que actualmente se podría prácticamente decir que dependemos de ellos en nuestra vida diaria. De este modo, cada vez usamos más este tipo de dispositivos en nuestro día a día para acceder a la información y a los servicios que nos ofrecen, con lo que en los últimos años ha surgido la necesidad de poder acceder a la información en la red en cualquier instante y desde cualquier lugar.

Estos cambios en la forma de comunicarnos y de socializar requieren nuevos interfaces que establezcan unos medios de comunicación entre los seres humanos y las máquinas que sean más eficientes e intuitivos, y que nos ayuden a manejarlos de una manera más sencilla y accesible por cualquier usuario. Por este motivo, han surgido recientemente sistemas que utilizan una de las formas más naturales de comunicación de las que disponemos los seres humanos, la comunicación mediante la voz.

Mediante la utilización de los sistemas de diálogo oral (Delgado y Araki, 2005; McTear, 2004) se facilita el manejo mediante la voz de programas informáticos de uso cotidiano, así como el control de aplicaciones o sistemas. De hecho, el número de aplicaciones de estos sistemas es enorme, como por ejemplo, posibilitar la utilización de aplicaciones en entornos en los que el uso de los interfaces tradicionales (teclado y

ratón) no está permitido, por ejemplo, a la vez que conducimos un automóvil, o facilitar la accesibilidad a las aplicaciones para personas con discapacidades visuales o motoras que les imposibilitan utilizar los interfaces tradicionales. Pero no todo son ventajas, uno de los mayores inconvenientes de estos sistemas es la dificultad actual de realizar un reconocimiento exacto de la frase pronunciada por el interlocutor debido a la diferente forma de pronunciar las palabras que tenemos cada persona, así como la influencia del ruido ambiental que puede aparecer dependiendo de cada entorno.

1.2 Aplicación desarrollada y objetivos

El objetivo fundamental de este Proyecto Fin de Carrera no es otro que desarrollar un sistema de diálogo mediante una aplicación que posibilite una utilidad práctica. En este caso, se ha elegido el dominio de las noticias por Internet y en concreto del uso de documentos RSS, conocidos con el término inglés *feed* (http://www.rssboard.org/rss-specification). A su vez, desarrollar este proyecto ha servido para ampliar los conocimientos sobre este tipo de sistemas, así como sobre el lenguaje VoiceXML (Voice eXtensible Markup Language) (W3C, 2004), definido como el estándar para el acceso oral a la información en Internet, además de otros lenguajes como PHP (www.php.net), JavaScript (www.javascript.com) o SQL (www.sql.org). Paralelamente, se ha podido poner en práctica técnicas sobre el desarrollo de proyectos adquiridas de forma teórica durante la carrera.

El sistema desarrollado comprende los siguientes módulos y funcionalidades principales:

- **Identificación de la llamada:** En esta utilidad simplemente se identifica el número del llamante, para remitírselo a los siguientes módulos de la aplicación, con la intención de posteriormente almacenarlo también en la base de datos. De este modo, es posible reconocer si el usuario ya ha utilizado previamente el sistema para poder atenderle así de una forma más personalizada.

- **Elección de fuente:** En este módulo de la aplicación se da la bienvenida al usuario, y se le pregunta acerca de la fuente de noticias de la que quiere obtener los *feeds* (podrá elegir entre las siguientes: ABC, Cadena Cope, El País, El Mundo y Cadena Ser). Además, se ofrece la opción de obtener ayuda sobre los controles definidos para la interacción y su forma de uso.

- **Extracción de las noticias:** Una vez que se ha obtenido la fuente y la categoría de noticias, se redirige al usuario a este módulo. En él se realiza la extracción de la información requerida de los *feeds*, que se almacenan en las base de datos para su posterior uso. A continuación, se analizan los titulares obtenidos en busca de los temas más populares de esa categoría. Una vez completado este análisis, se ofrece al usuario la selección entre las opciones general, correspondiente a la lectura de todas las noticias de esa categoría, o de los temas populares recientemente extraídos, que consistirá en la lectura únicamente de las noticias del tema seleccionado.

- **Lectura de noticias:** En el momento que la aplicación conoce la fuente, la categoría y el tema sobre el que el usuario desea escuchar las noticias, genera dinámicamente la página correspondiente para la lectura de noticias. Esta página contiene la información necesaria para leer de la base de datos las noticias deseadas, así como las instrucciones para el funcionamiento de los controles de la interacción, estos son:

    - **Ampliar:** Para obtener una descripción más detallada de la noticia que se está leyendo en ese momento.
    - **Siguiente:** Para pasar a la siguiente noticia de la que se está escuchando en ese momento.
    - **Anterior:** Para pasar a la anterior noticia de la que se está escuchando en ese momento.
    - **Repetir:** Para poder repetir la noticia que se está leyendo en ese momento.

- **Volver:** Para poder regresar a un menú anterior y hacer una selección diferente de noticias. Esta opción está disponible durante todo el uso de la aplicación.

Tras describir a grandes rasgos la aplicación desarrollada, enumeramos a continuación los objetivos principales que se definieron para la misma:

- Utilizar los lenguajes VoiceXML y PHP para la generación dinámica de los diferentes módulos de la aplicación, con los que se interactué mediante el uso de la voz únicamente, demostrando así las posibilidades que ofrecen los sistemas de diálogo hablado.

- Maximizar la accesibilidad del sistema mediante el desarrollo de una aplicación que pueda ser utilizada en cualquier lugar a través de una simple llamada de teléfono, así como ofrecer esta modalidad de acceso para usuarios con discapacidades motoras o visuales.

- Facilitar la lectura de noticias en tiempo real, mediante la utilización de *feeds* RSS, contando que estas noticias puedan pertenecer a diferentes medios. Una vez obtenidas estas noticias, el sistema debe ser capaz de clasificarlas de manera que el usuario pueda elegir entre las categorías y temas populares definidos en ese momento.

- Proveer la lectura de las noticias y navegación por la aplicación mediante la incorporación de menús y controles avanzados.

- Personalizar el sistema de modo que si se detecta que el usuario ya ha utilizado el sistema anteriormente, sea posible ofrecerle una serie de recomendaciones de forma dinámica, basándose en su historial y sus consultas favoritas.

1.3 Fases de desarrollo

Para la realización del Proyecto Fin de carrera se definieron las siguientes fases de desarrollo:

- Análisis de los requisitos de la aplicación: a lo largo de esta primera fase se definieron los requisitos necesarios para el correcto funcionamiento y desarrollo de la aplicación. Para el establecimiento de estos requisitos se realizó un estudio sobre las posibilidades de los lenguajes de desarrollo y sus limitaciones, así como de las posibilidades del sistema en sí, estableciendo tanto los requisitos funcionales como no funcionales del mismo. La definición de estos requisitos se hizo a petición del cliente, que en este caso se trata del tutor del proyecto, David Griol. Además de los requisitos establecidos por el cliente, se han añadido otros propios de los sistemas basados en voz, y en concreto, del acceso telefónico al sistema. El conjunto de estos requisitos se detalla en la Sección 3.1 de la memoria.

- Estudio de los lenguajes de programación VoiceXML, PHP y JavaScript, así como de las funcionalidades proporcionadas por la plataforma VoiceXML utilizada para la aplicación, en este caso, el servidor de voz Voxeo (www.voxeo.com), que se describe con detalle en la Sección 2.5. Para la realización de este estudio previo se consultaron diferentes manuales y artículos científicos proporcionados por el tutor y los foros facilitados por la plataforma Voxeo. Esta fase se prolongó durante la fase de desarrollo del sistema, ya que según se iba implementando la misma, fueron surgiendo diferentes dudas y problemas que fue necesario resolver mediante la consulta de estos tutoriales y manuales.

- Desarrollo y evaluación del sistema: Se trata de la fase más laboriosa y que más tiempo ha requerido, pero a su vez también ha sido la fase más amena del proyecto, personalmente hablando. Tal y como se ha comentado en la fase anterior, debido a que las tecnologías y los lenguajes eran

desconocidos inicialmente para mí, he tenido que consultar diferentes tutoriales y manuales durante el desarrollo de esta fase del proyecto.

- Elaboración de la memoria explicativa del Proyecto Final de Carrera: Se trata de una de las últimas fases del proyecto, en la que se ha desarrollado el presente documento con el principal objetivo de exponer de la manera más clara y detallada posible, todos los aspectos referentes a la aplicación desarrollada.

- Realización de la presentación: es la última de las fases del proyecto, consistente en la realización de una breve presentación en la que se resuma y se exponga de la manera más clara y concisa posible, las características y los fines de la aplicación desarrollada.

1.4 Medios empleados

Respecto a los medios empleados para el desarrollo de la aplicación, podemos dividir los mismos en dispositivos hardware y aplicaciones software. En cuanto a los dispositivos hardware, hemos necesitado un ordenador que cumpla con los requisitos para programar en Notepad, y con conexión a Internet preferiblemente de banda ancha para realizar la conexión al servidor de voz. A su vez, hemos necesitado la contratación de un servidor web, que disponga de soporte para PHP y SQL, incluyendo la aplicación PHPMyAdmin (http://www.phpmyadmin.net). También ha sido necesaria la utilización de la plataforma Voxeo a modo de servidor de voz, para la gestión de las llamadas y la ejecución del código VoiceXML. Por supuesto, es necesario disponer de un sistema de altavoces y micrófono para poder realizar las llamadas a la aplicación durante su desarrollo y pruebas.

En cuanto a software, simplemente hemos utilizado el programa Notepad++, para programar la aplicación, y un navegador web para acceder a las diferentes funcionalidades proporcionadas tanto por la plataforma Voxeo como el servidor web.

- Dispositivos Hardware:
  - Ordenador que cumple con los requisitos para ejecutar un navegador web y el editor de texto Notepad, así como conexión a Internet, preferiblemente de banda ancha. El ordenador es un Intel Core 2 Duo 1,8 GHz con 2GB de memoria RAM.
  - Altavoces, micrófono.
  - Periféricos habituales: teclado, ratón, pantalla.
  - Servidor de VoiceXML Voxeo
  - Hosting web con soporte para PHP y Bases de datos MySQL.

- Aplicaciones Software:
  - Navegador web (véase Mozilla Firefox, www.firefox.com, o Google Chrome, www.google.com/chrome).
  - Editor de textos Notepad++.

- Otros:
  - Servidores de noticias con *feeds* RSS.

Respecto a los dispositivos necesarios para utilizar la aplicación, únicamente es necesario un teléfono desde el que podamos realizar una llamada al teléfono nacional asignado a la aplicación de forma gratuita por Voxeo.

Es posible además utilizar la aplicación mediante otras vías de acceso, por ejemplo, a través de Skype (http://www.skype.com), para lo que será necesario un equipo que pueda ejecutar esta aplicación y disponga de una conexión a Internet.

1.5 Estructura de la memoria

La presente memoria del proyecto se estructura en un total de cinco capítulos, así como un glosario de términos y un apartado referente a la bibliografía consultada.

- **Capítulo 1:** ***Introducción y Objetivos.*** Se trata del capítulo inicial de la memoria, en el que se incluye una pequeña introducción a las temáticas fundamentales tratadas en el Proyecto Final de Carrera, y se definen los

objetivos principales del mismo. Además, se realiza una explicación sobre los medios empleados, así como las fases de desarrollo.

- **Capítulo 2:** ***Estado del Arte.*** En este capítulo se describe detalladamente qué es un sistema de diálogo, su arquitectura, características de sus módulos fundamentales, aplicaciones, historia y sistemas destacados. Con respecto a las principales tecnologías para su desarrollo, se realiza un especial énfasis a la descripción del lenguaje VoiceXML. Este lenguaje se ha definido, tal y como se ha comentado previamente, como el estándar para el acceso oral a la información en Internet.

- **Capítulo 3:** ***Descripción general de la aplicación desarrollada.*** Durante este capítulo se ofrece una visión global de la aplicación desarrollada. El capítulo recopila los requisitos del sistema, tanto funcionales como no funcionales, y realiza un estudio del funcionamiento de la aplicación mediante diagramas de casos de uso. Por último, se ofrece una visión general de la arquitectura definida para el desarrollo de la aplicación.

- **Capítulo 4:** ***Descripción detallada de los módulos del sistema.*** En este capítulo se incluye una amplia explicación sobre cada uno de los módulos que componen la aplicación, explicando su funcionamiento y su arquitectura los más detalladamente posible. Por último, se describe la evaluación llevada a cabo de la aplicación y los resultados obtenidos.

- **Capítulo 5:** ***Conclusiones y líneas futuras.*** En este capítulo se resumen las principales conclusiones obtenidas tras la realización del proyecto, exponiendo además las aportaciones personales que he obtenido con la realización del mismo, así como los nuevos conocimientos adquiridos. Además, en este capítulo se detallan posibles ideas para mejorar la aplicación de cara incluso a una distribución comercial.

- **Glosario:** En este apartado se incluye una breve definición de diferentes términos y conceptos fundamentales que se describen a lo largo de la memoria, y que pudieran suponer cierta dificultad para el entendimiento del trabajo realizado.

- **Bibliografía:** En este capítulo se enumeran las diferentes citas bibliográficas que se han consultado para la realización del proyecto.

1.6 Diagrama de costes, planificación temporal y presupuesto

A continuación, se muestra el desglose de las diferentes tareas fundamentales llevada a cabo para la consecución del proyecto, mediante su enumeración en la Figura 1 y el diagrama de Gantt de la Figura 2. Este diagrama posibilita una visión global de la planificación del proyecto y de las distintas etapas que lo componen, así como su duración global. Debido a que se trata de un Proyecto Final de Carrera, no se han tenido en cuenta todos los posibles costes derivados de materiales necesarios para su realización, simplemente los principales, mostrados en el presupuesto que se detalla a continuación.

	Nombre de tarea	Duración	Comienzo	Fin
1	**Fase de Análisis**	**23 días**	**mar 15-06-10**	**jue 15-07-10**
2	Análisis de requisitos	18 días	mar 15-06-10	jue 08-07-10
3	Diagrama de casos de	5 días	vie 09-07-10	jue 15-07-10
4	**Estudio de diferentes t**	**40 días**	**vie 16-07-10**	**jue 09-09-10**
5	VoiceXML	10 días	vie 16-07-10	jue 29-07-10
6	PHP	10 días	vie 30-07-10	jue 12-08-10
7	Integración de PHP con	10 días	vie 13-08-10	jue 26-08-10
8	MySQL y PHPmyAdmin	10 días	vie 27-08-10	jue 09-09-10
9	**Fase de Diseño**	**33 días**	**vie 10-09-10**	**mar 26-10-10**
10	Diseño de las páginas	17 días	vie 10-09-10	lun 04-10-10
11	Diseño de la Base de D	16 días	mar 05-10-10	mar 26-10-10
12	Fase de desarrollo de la a	135 días	mié 27-10-10	mar 03-05-11
13	Evaluación del sistema	20 días	mié 04-05-11	mar 31-05-11
14	Documentación del proyec	30 días	mié 01-06-11	mar 12-07-11
15	Preparación de presentaci	5 días	mié 13-07-11	mar 19-07-11

Figura 1. Desglose de las tareas del proyecto y sus duraciones

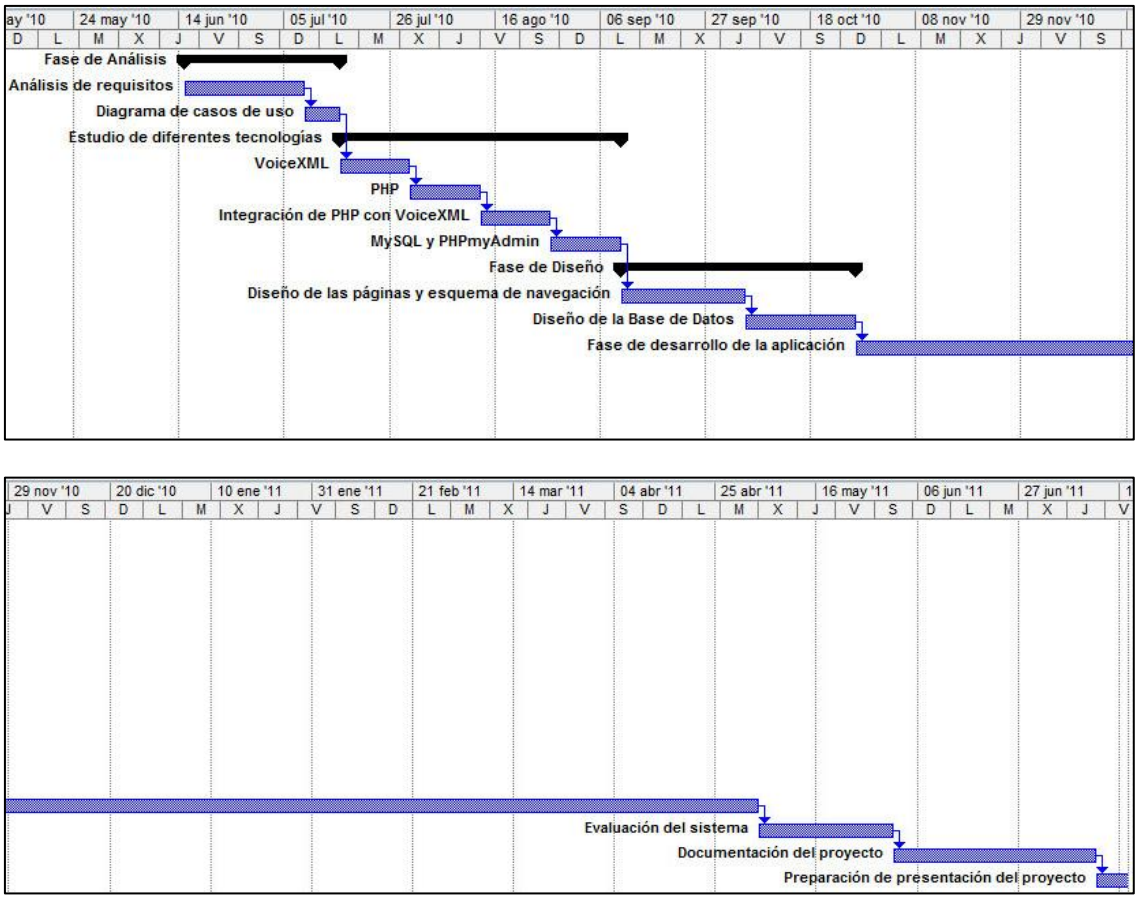


Figura 2. Diagrama de Gantt con la planificación temporal del proyecto

Para la elaboración del presupuesto del Proyecto se ha tenido en cuenta los diferentes gastos realizados en materiales o herramientas, así como el coste imaginario que podrían tener las horas dedicadas al desarrollo de la aplicación.

En la Tabla 1 podemos observar el coste de personal. Al ser solo una persona la que ha trabajado en la implementación del proyecto, los costes de personal se reducen al sueldo de un único ingeniero.

Apellidos y Nombre	Categoría	Dedicación (meses)	Coste (por mes)	Coste Total
Gil Borrego, Víctor	Ingeniero	4	1500 €	6000 €

Tabla 1. Presupuesto de personal

La Tabla 2 muestra los costes de los equipos. En ella se detalla el coste de alquiler del alojamiento en un servidor web por mes, y el coste del equipo en el que desarrolló la aplicación. El coste de este último es único, por lo que no es necesario realizar un pago mensual.

Descripción	Coste	% Uso dedicado	Dedicación (meses)	Coste Total
Hosting	7 €	100	4	24 €
Equipos de desarrollo	500 €	100	4	500 €
Total				524 €

Tabla 2. Presupuesto de equipos

Por último, se muestra el resumen del presupuesto total sobre el desarrollo e implementación de la aplicación.

Presupuesto Costes Totales	Presupuesto Costes
Personal	6000 €
Equipos	524 €
Total	6524 €

Tabla 3. Presupuesto total del Proyecto

Capítulo 2

Estado del Arte

En este capítulo describimos las principales características de los sistemas de diálogo, tanto técnicamente como respecto a sus principales aplicaciones, así como del formato RSS para sindicar o compartir contenidos en la web. El capítulo comienza con una pequeña introducción sobre este formato, para seguidamente definir en qué consisten los sistemas de diálogo y detallar algunas de las aplicaciones más relevantes para las que se han utilizado estos sistemas en los últimos años, así como las diferentes líneas de investigación que engloba el desarrollo de estos sistemas. Seguidamente, se describe la arquitectura típica utilizada para el desarrollo de estos sistemas, describiendo cada uno de los módulos, comentado qué funcionalidades realizan e incorporando un pequeño ejemplo que clarifica cómo funcionan y qué técnicas fundamentales se utilizan actualmente para el desarrollo de estos módulos. Posteriormente, detallamos la evolución seguida por estos de los sistemas, comentando los principales retos actuales. Finalmente, se detallan las principales características del estándar VoiceXML, así como de la plataforma Voxeo, que implementa las especificaciones de este estándar.

2.1 El formato RSS

RSS (Really Simple Syndication) es un formato XML definido para la sindicación de documentos y poder publicar contenidos y mantenerlos actualizados, como puede ser el contenido de blogs o de noticias. Un documento de RSS, o *feed*, incluye una serie de información en formato de texto e indexada mediante etiquetas.

El beneficio principal que obtienen los publicadores al utilizar RSS es el de poder publicar sus artículos o información de manera automática. El beneficio, que por otra parte, obtienen los lectores, es la posibilidad de subscribirse automáticamente a

contenidos en Internet, así como la posibilidad de agrupar publicaciones de contenidos de muchas webs, en un solo programa u otra web.

Los archivos RSS pueden ser leídos utilizando un software específico, o incluso mediante herramientas de lectura online, como por ejemplo "Google Reader" (http://www.google.es/reader/), además de ello también existen diferentes programas para móviles que soportan el uso de RSS.

La historia del formato RSS viene precedida por varios intentos de agrupar la publicación de contenidos web, que no tuvieron demasiado éxito. La idea básica de reestructurar la información del contenido de la web proviene de 1995, cuando Ramanathan V. Guha (Guha, 1999) y otros componentes del "Apple Computer's Advanced Technology Group" desarrollaron un conjunto de metaetiquetas para la web.

El formato RDF, la primera versión de RSS, fue desarrollado por Guha para el navegador Netscape (http://netscape.aol.com/) en marzo de 1999. Esta versión pasó posteriormente a ser conocida como RSS 0.9. En Julio de 1999, Dan Libby, de la empresa Netscape, produjo la nueva versión, RSS 0.91, que simplificaba el formato de las anteriores eliminando elementos de RDF e incorporaba otros nuevos elementos de un sistema de formato de noticias perteneciente a Dave Winer (RSS Advisory Board, 2009). Ésta fue la última aportación de Netscape en el desarrollo de RSS en los siguientes ocho años, mientras el formato RSS era utilizado por diferentes publicadores para colocar sus "feeds" en el portal de Netscape.

En diciembre del año 2000, un grupo de los desarrolladores de las primeras versiones de RSS, entre ellos Guha, crean el "RSS-DEV Working group". Este grupo publicaría la versión 1.0 de RSS, en el mismo mes de su fundación, añadiendo al estándar el soporte para las etiquetas de XML. Durante este tiempo, el grupo RSS-DEV Working group y la empresa de Dave Winer, UserLand Software, se encargaron de publicar algunas herramientas para la lectura de RSS fuera del portal de Netscape.

En septiembre de 2002 se introdujo la nueva versión de RSS, conocida como RSS 2.0. Esta nueva versión preservaba la compatibilidad con las versiones anteriores de RSS, e incluía algunas mejoras en el soporte de los contenidos y su estructura.

En julio de 2003, UserLand Software asignó los derechos de autor de la especificación de RSS 2.0 al centro de desarrollo de Harvard, "Berkman Center for Internet & Society". Al mismo tiempo, Winer lanzaba un foro de consulta sobre el formato RSS, junto a Bret Simmons y Jon Udell, con el fin de dar soporte y ayuda a las preguntas que se planteaban los desarrolladores sobre este formato.

Por último, en diciembre de 2005, el equipo de Microsoft Internet Explorer y de Microsoft Outlook, anunciaron en sus blogs que adoptaban el icono , que por primera vez se utilizó en Mozilla Firefox para la representación de este formato. Otras empresas de desarrolladores de herramientas y contenidos para la web, como Opera Software, han ido progresivamente aceptando el formato RSS para la publicación de noticias y contenidos en la web.

La Figura 3 muestra un ejemplo de un fichero RSS, en el que puede apreciarse la estructura de estos ficheros. Tal y como puede observarse, se trata de fichero en formato XML, comenzando con las cabeceras propias de este lenguaje y utilizando a continuación las etiquetas de RSS en las que se va intercalando información sobre los contenidos, delimitados entre las etiquetas <*item*>.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
        <title>RSS Title</title>
        <description>This is an example of an RSS feed</description>
        <link>http://www.someexamplerssdomain.com/main.html</link>
        <lastBuildDate>Mon, 06 Sep 2010 00:01:00 +0000
</lastBuildDate>
        <pubDate>Mon, 06 Sep 2009 16:45:00 +0000 </pubDate>

        <item>
                <title>Example entry</title>
                <description>Here is some text containing an
interesting description.</description>
                <link>http://www.wikipedia.org/</link>
                <guid>unique string per item</guid>
                <pubDate>Mon, 06 Sep 2009 16:45:00 +0000 </pubDate>
        </item>

</channel>
</rss>
```

Figura 3. Ejemplo de un fichero RSS

La Figura 4 muestra otro ejemplo de un fichero RSS, en este caso conteniendo una noticia de prensa utilizada como fuente para el repositorio de la aplicación desarrollada para el Proyecto Final de Carrera. Cabe destacar la estructura y delimitación de campos facilitada por el formato RSS para posibilitar así la extracción de los diferentes campos de forma sencilla. Tal y como vemos en la figura, las diferentes noticias aparecen estructuradas dentro de cada par de etiquetas de *<ítem>*.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rss version="2.0" xmlns:media="http://search.yahoo.com/mrss/">
<channel>
<title><![CDATA[ELPAIS.com - Sección España]]></title>
<link><![CDATA[http://www.elpais.com/espana/]]></link>
<description><![CDATA[ELPAIS.com - Sección España]]></description>
<lastBuildDate>Thu, 03 Mar 2011 11:43:06 +0100</lastBuildDate>
<language>es-es</language>
<copyright><![CDATA[Copyright Prisa Digital S.L.]]></copyright>
<ttl>0</ttl>
<image>
<url>http://www.elpais.com/im/tit_logo.gif</url>
<title>ELPAIS.com - Sección España</title>
<link>http://www.elpais.com</link>
</image>
<item>
<title><![CDATA[El Gobierno alega que ETA es "motor y parte actora de Sortu"]]></title>
<link><![CDATA[http://www.elpais.com/articulo/espana/Gobierno/alega/ETA/motor/parte/actora/Sortu/elpepu
<description><![CDATA[La Abogacía del Estado considera que<a href="http://www.elpais.com/articulo/espar
<guid isPermaLink="true"><![CDATA[http://www.elpais.com/articulo/espana/Gobierno/alega/ETA/motor/parte/
<author><![CDATA[JULIO M. LÁZARO]]></author>
<pubDate><![CDATA[Thu, 03 Mar 2011 11:11:00 +0100]]></pubDate>
</item>
<item>
<title><![CDATA[Varios encapuchados queman neumáticos y atacan una sucursal bancaria en Vitoria]]></tit
<link><![CDATA[http://www.elpais.com/articulo/espana/Varios/encapuchados/queman/neumaticos/atacan/sucur
<description><![CDATA[Varios encapuchados han protagonizado dos ataques callejeros en dos lugares muy p
<guid isPermaLink="true"><![CDATA[http://www.elpais.com/articulo/espana/Varios/encapuchados/queman/neum
<author><![CDATA[EFE]]></author>
<pubDate><![CDATA[Thu, 03 Mar 2011 09:36:00 +0100]]></pubDate>
</item>
<item>
<title><![CDATA[El Gobierno maneja tres niveles de alerta por si empeora la crisis petrolera]]></title>
<link><![CDATA[http://www.elpais.com/articulo/espana/Gobierno/maneja/niveles/alerta/empeora/crisis/petr
<description><![CDATA[La crisis del petróleo va en serio, y puede empeorar. Al menos eso es lo que pien
<guid isPermaLink="true"><![CDATA[http://www.elpais.com/articulo/espana/Gobierno/maneja/niveles/alerta/
<author><![CDATA[CARLOS E. CUÉ]]></author>
<pubDate><![CDATA[Thu, 03 Mar 2011 07:51:00 +0100]]></pubDate>
</item>
```

Figura 4. Fichero RSS mostrando el contenido de una noticia (extraído de la versión on-line del periódico El País)

2.2 Sistemas de Diálogo

Tal y como se ha comentado en el capítulo de introducción de esta memoria, un sistema de diálogo hablado puede definirse como un programa informático que emula la capacidad de diálogo entre dos seres humanos, utilizando la voz como modalidad de comunicación.

La implementación de un sistema de diálogo constituye un problema complejo que se suele descomponer en varios subproblemas, los cuales se corresponden con módulos que realizan funciones concretas, tal y como se muestra en la Figura 5.

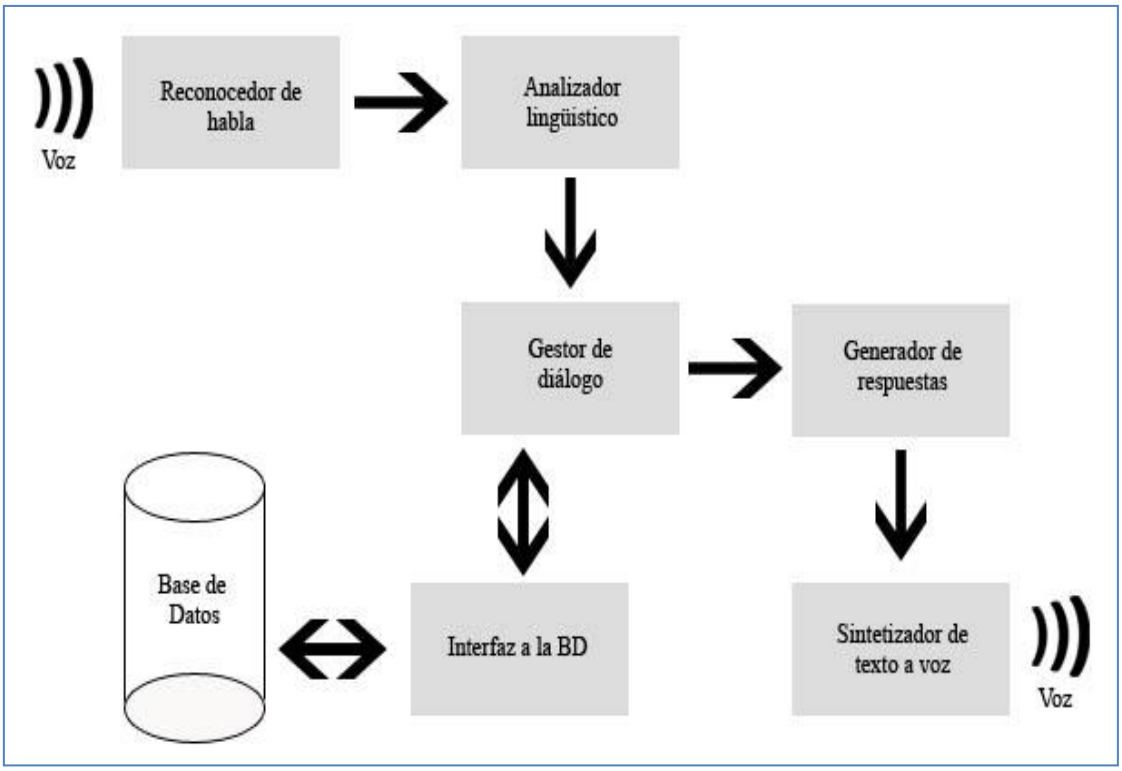


Figura 5. Arquitectura modular de un sistema de diálogo

Estos módulos son:

- Módulo reconocedor del habla (*Automatic Speech Recognition*)
- Módulo analizador lingüístico (*Natural Language Understanding*)
- Módulo de gestión de diálogo (*Dialogue Management*)
- Módulo de generación de respuestas (*Natural Language Generation*)
- Módulo sintetizador de voz (*Text-to-Speech Synthesis*)

Cada uno de estos módulos se ocupa de una tarea concreta dentro de la función global del sistema de diálogo. El reconocedor de habla recibe la voz del usuario y la transforma en una cadena de palabras en forma de texto. A partir de la cadena obtenida, el analizador lingüístico se ocupa de obtener el significado de la frase reconocida, dentro del dominio específico del sistema.

Esta representación semántica de la frase reconocida se le suministra al gestor de diálogo, cuya finalidad principal es dar coherencia entre la pregunta del usuario y la respuesta del sistema, resolviendo posibles anáforas y elipsis y prediciendo las

reacciones del usuario. Éste módulo, determina qué acción debe realizar el sistema en cada momento, por lo que podemos decir que es el módulo fundamental del sistema.

A continuación, una vez que el gestor de dialogo ha decidido la acción a realizar, ésta se transmite al módulo de generación de respuestas, que genera una frase en lenguaje natural a partir de la representación interna proporcionada por el sistema. Esta frase será la entrada del sintetizador de voz que generará la respuesta sonora que se reproducirá al usuario.

Distinguimos, representado en la Figura 5, los diferentes módulos citados anteriormente y cómo se incorporan adicionalmente las bases de datos de la aplicación, que se consultan mediante un gestor que se encarga de tratar dichas consultas y proporcionar la información obtenida por las mismas al módulo de gestión de diálogo.

A continuación describimos de forma más detallada cada uno de los módulos del sistema (Callejas, 2008).

2.2.1 Módulo de reconocimiento del habla

Nos referimos al módulo de reconocimiento del habla cuando hablamos del módulo que obtiene la frase que con mayor probabilidad se asigna a la señal de voz del usuario. Esta tarea depende de muchos factores como la lingüística del interlocutor, el canal por el que se transmite, el contexto de la frase y la psicología del interlocutor. Es por estas razones por las que es uno de los módulos más complejos. En cuanto a la lingüística del interlocutor podemos encontrar factores determinantes como el tono de voz de cada individuo y los componentes fonéticos, sintácticos y semánticos que afectarían directamente a la señal de la voz. Cuando nos referimos al canal por el que se transmite, hablamos de los posibles ruidos acústicos que se puedan filtrar al recoger la voz, así como los diferentes tipos de micrófonos que puedan utilizar los usuarios con diferentes tipos de sensibilidades, que obligarán al módulo de reconocimiento de la voz, a tener cierta robustez y a adaptarse a cada situación.

Respecto al contexto de la frase, podemos encontrar diferencias si el interlocutor está leyendo un texto, mencionando palabras o frases aisladas o si se está expresando en lenguaje natural (Griol, 2007). Por ejemplo, mientras que esté hablando de forma natural puede darse la situación en la que se repitan palabras, o se generen pausas producidas por el interlocutor, que el sistema deberá interpretar. Dependiendo también de la psicología de cada individuo, en cuanto a la forma de hablar, el sexo, la edad, o la nacionalidad, podemos encontrar también grandes diferencias en cuanto a la interpretación de cada frase, incluso un mismo individuo puede pronunciar de diferente forma una misma palabra dependiendo de la situación y de las palabras que le precedan o sigan a continuación.

Para poder ejecutar la tarea de reconocer el habla del interlocutor, el sistema puede utilizar diferentes metodologías, pero sin duda, la más utilizada es la aproximación estadística. Ésta se puede resumir en tratar de encontrar una secuencia de palabras *W* dada una secuencia de ondas acústicas *A*. Esta secuencia *W* puede determinarse a partir de la siguiente expresión:

$$W = \max_{W} P(W|A)$$

Para deducir la probabilidad $P(W|A)$ suele aplicarse la regla de Bayes, con lo que podemos redefinir la expresión anterior de esta forma:

$$P(W|A) = \frac{P(A|W)P(W)}{P(A)}$$

Esta expresión, $P(W|A)$, se define como la probabilidad de obtener la señal acústica *A* cuando se han pronunciado la secuencia de palabras *W*. $P(W)$ se refiere a la probabilidad de pronunciar la secuencia de palabras *W*, probabilidad que nos proporciona por el modelo del lenguaje del reconocedor. Por último, dado que la probabilidad referente al modelo acústico es independiente de la secuencia de palabras, podemos reescribir la expresión inicial de la siguiente forma:

$$\mathrm{W} = \max_{W} P(A|W)P(W)$$

2.2.2 Módulo de análisis lingüístico

Cuando el módulo de reconocimiento de la voz ha obtenido el texto correspondiente a la señal oral es el momento de que el módulo de análisis lingüístico trate de desvelar su significado en el dominio del sistema. Para poder completar esta tarea el módulo de análisis lingüístico lleva a cabo análisis morfológicos y léxicos, sintácticos, semánticos y de discurso de la frase suministrada.

Los análisis morfológico y léxico analizan de forma independiente los diferentes morfemas y lexemas de cada palabra, los lexemas para obtener la parte semántica de la palabra y los morfemas para tratar de conseguir los diferentes infijos y sufijos y así obtener los diferentes tipos de clases de palabras. A continuación, se realiza un análisis sintáctico de la frase, que al tratarse de una frase recogida de forma oral, puede verse afectado debido a las pausas, correcciones o repeticiones realizadas por el usuario al introducir la frase. Posteriormente, el análisis semántico trata de extraer el significado de la estructura de la frase completa a partir de sus diferentes componentes. Por último, se realiza un análisis dentro del contexto del discurso para poder dar el significado correcto a la frase y enmarcarlo dentro del tipo contexto en el que se haya dicho la frase.

Para poder afrontar el problema de la comprensión existen dos tipos de aproximaciones: la comprensión basada en reglas y la comprensión basada en modelos estadísticos. La primera alternativa, basada en reglas, extrae la información semántica a partir de un análisis semántico-sintáctico de la frase, de la manera que se ha descrito anteriormente. Existen analizadores que para conseguir un análisis más detallado, utilizan gramáticas para extraer la información semántica relevante. La metodología basada en reglas con un uso más extendido consiste en extraer la información semántica de la frase mediante la utilización de gramáticas previamente definidas y la detección de palabras clave.

En el caso de los modelos estadísticos, el proceso se basa en la definición de unidades lingüísticas y la obtención de modelos a partir de muestras etiquetadas. Este tipo de análisis (Minker, 1998; Segarra et al., 2002) emplea un modelo probabilístico

para identificar los conceptos, valores, y decodificar semánticamente las locuciones del usuario. Este modelo se genera durante una primera fase de entrenamiento (aprendizaje) en la que se capturan las correspondencias entre las entradas de texto y su representación semántica. Una vez que el modelo de entrenamiento se ha aprendido, éste se utiliza a modo de decodificador para generar la representación semántica de la entrada. De este modo el proceso de comprensión se realiza de forma similar al del reconocimiento del habla.

2.2.3 Gestor de diálogo

Cuando hablamos del gestor de diálogo nos referimos al módulo más complejo y que más tareas realiza en el sistema de diálogo. Actualmente se suelen resumir estas tareas en cuatro principales: actualizar el contexto del diálogo, proveer del contexto en el que basar las interpretaciones, coordinar el resto de módulos, decidir cuál es la información que se entrega al usuario y en qué momento ha de entregarse.

Para poder completar estas tareas, el gestor de diálogo debe tratar con la información obtenida de los demás módulos, como los resultados que obtiene del trabajo del módulo de procesamiento del lenguaje natural, las información obtenida a partir de las consultas realizadas con la base de datos del sistema, la información que posea acerca del tipo de usuarios que interactúa con el sistema e información del contexto. Debido a la gran diversidad de fuentes de información con las que debe tratar y el gran trabajo que realiza, podríamos definir a este módulo como el cerebro dentro del sistema de diálogo, ya que es el encargado de decidir y tomar decisiones para que se produzca una correcta interacción persona-máquina.

Para su implementación se han empleado diferentes metodologías. La metodología más sencilla se basa en modelar el diálogo como una máquina de estados finitos, donde el sistema utiliza las acciones de los usuarios para modelar las transiciones entre las respuestas del sistema. La metodología basada en *frames* se basa en crear una estructura compuesta de campos (conceptos y atributos) que el sistema se encargará de ir cumplimentando según se va recibiendo la información del usuario. La ventaja principal de esta metodología es dotar de mayor flexibilidad la

interacción posibilitando que el usuario pueda proporcionarla información al sistema en cualquier orden.

Para tareas más complejas el sistema puede utilizar estrategias basadas en planes (Allen y Perault, 1980; Appelt, 1985; Cohen y Levesque, 1988). Este tipo de metodologías se fundamentan en la definición de un objetivo común al diálogo entre el usuario y el sistema. De este modo, un sistema de diálogo que esté basado en planes trabaja de forma cooperativa con el usuario para alcanzar el objetivo común (Allen et al., 2000). A través de cada intervención, el sistema tratará de ir detectando las intenciones del usuario, por lo que cada frase del interlocutor se deberá tratar como una declaración de intenciones. El sistema se encargará de ir repitiendo recursivamente estas acciones hasta alcanzar el objetivo de la tarea.

La denominada Teoría del Estado de la Información (Traum et al., 1999) está íntimamente relacionada con la estrategia basada en planes. Esta estrategia trata de almacenar las conversaciones ya realizadas con el usuario para poder identificarlas de manera unívoca y que en el futuro se puedan utilizar para la contextualización de nuevos diálogos y conversaciones con los usuarios. Según esta teoría, el sistema deberá decidir la siguiente acción a realizar y actualizar el estado de la información, basándose en los comportamientos observados de los usuarios.

Si tuviéramos la necesidad de ejecutar operaciones en un dominio que cambie dinámicamente, podríamos utilizar la estrategia basada en agentes. Esta estrategia nos permite combinar las ventajas de los modelos basados en máquinas de estados finitos y los modelos basados en *frames*, a la vez que otras metodologías de gestión de diálogo.

Durante la última década han surgido diferentes iniciativas para el modelado estadístico del diálogo y el aprendizaje automático de la estrategia del mismo (Hurtado et al., 2005). Estos métodos se fundamentan en el modelado de la interacción del usuario con la máquina a partir de corpus de diálogos etiquetados. También es posible combinar diferentes técnicas de modelos basados en reglas y modelos de aprendizaje automático. Ejemplos de estas aproximaciones son el gestor Agenda desarrollado por

la CMU utilizando la arquitectura Ravenclaw (Bohus y Rudnicky, 2009), Queen's Communicator (O'Neill et al., 2003) y SesaMe (Broekstra et al., 2002).

2.2.4 Módulo de generación de respuestas

El módulo de generación de respuestas es el encargado de transformar la acción seleccionada por el gestor del diálogo en una frase en lenguaje natural (Jordán, 1992). Para conseguir este objetivo pueden utilizarse diferentes métodos, aunque en todos se suele seguir estos pasos: organización del contenido, distribución del contenido de las frases, lexicalización, generación de expresiones referenciales y realización lingüística. Existen tres metodologías principales para el desarrollo del módulo de generación de respuestas: la utilización de frases predeterminadas, la utilización de plantillas y la más compleja de todas, los sistemas basados en características (Farrús, et al., 2005).

En la metodología basada en frases predeterminadas, el sistema utiliza un conjunto predeterminado de frases que expresan, de una manera sencilla y fácil de comprender, las acciones más comunes que el sistema requiera para comunicar al usuario el total de acciones definidas para el sistema. El principal problema de esta metodología es su falta de flexibilidad, ya que sólo podemos ceñirnos a utilizar un número reducido de frases, por lo que en el caso de sistemas más complejos necesitaremos otro tipo de estrategia.

Una de las alternativas a la metodología basada en frases predeterminadas, consiste en la utilización de plantillas en las que parte del contenido sea estático y otra parte pueda personalizarse. En esta metodología, el sistema debe seleccionar primero un patrón entre todas las plantillas definidas para posteriormente completarla sustituyendo los nombres de conceptos y/o atributos de la tarea por su correspondiente valor, de forma que se obtenga al final de este proceso una frase en lenguaje natural.

Por último, la metodología más sofisticada y que más alternativas y flexibilidad ofrece es la basada en características. En esta metodología, cada posible variable de una frase se etiqueta como una característica. De este modo, por ejemplo, una frase

puede ser positiva o negativa, puede emplearse de modo imperativo, de modo de pregunta o de afirmación, de diferente forma en función de su tiempo verbal, etc. Para poder manejar y ordenar todas estas características, el sistema requiere de un conocimiento lingüístico con el objetivo de poder construir las elocuciones de la manera más adecuada y legible. Éste que puede aprenderse a partir de corpus de diálogos etiquetado (Oh y Rudnicky, 2000).

2.2.5 Módulo sintetizador de voz

Para completar el proceso realizado por el resto de módulos y facilitar una salida oral, el sistema debe convertir el texto resultante de la salida del módulo de generación de respuestas en una señal de voz. Este proceso se lleva a cabo mediante un sintetizador de texto a voz.

Un sintetizador de texto a voz (text-to-speech) se comprende usualmente de dos partes fundamentales, el *front-end* y el *back-end*. El *front-end* realiza dos tareas fundamentales. La primera de ellas consiste en la extracción de las palabras de un texto plano. Este proceso se conoce como tokenización o normalización del texto. Para llevarlo a cabo el sistema puede usar diferentes marcadores, como puede ser los espacios en blanco, para ir diferenciando las diferentes palabras, así como identificar las posibles abreviaturas o números que aparezcan en el texto. En segundo lugar, el *front-end* se encarga de transcribir las palabras obtenidas, en su forma fonética, y de identificar y clasificar las palabras dentro de sus diferentes frases y oraciones. Por lo tanto, la salida del *front-end* es la transcripción fonética de la frase suministrada en formato texto.

A continuación, el *back-end* se encarga de convertir en voz los resultados obtenidos por el *front-end*. Para la realización de esta tarea existen diferentes métodos, que deben seleccionarse según el uso para el que va a ser destinado la voz sintetizada.

El primer método para la transcripción a voz es la síntesis concatenativa que consiste en ir concatenando pequeños fragmentos de voz que se encuentran pregrabados y almacenados en el sistema para lograr así la generación completa de las

ondas de sonido. Este método de síntesis, a su vez, tiene diferentes formas de aplicación. La primera de ellas consiste en el uso de grandes bases de datos en el que se tienen almacenadas las locuciones de las diferentes frases y palabras, por lo que el sistema concatenará estas locuciones para crear la señal de voz. La segunda de ellas consiste en la utilización de los diferentes difonos que contiene un idioma, que pueden almacenarse en una base de datos de un tamaño considerablemente menor. El tamaño de esta base de datos dependerá del idioma para el que se esté desarrollando el sintetizador, por ejemplo, el sintetizado en español de Loquendo (www.loquendo.com) tiene unos 800 difonos, mientras que el alemán tiene unos 2500. Por último, es común utilizar este método cuando el sintetizador de voz va a ser utilizado dentro de un dominio reducido de palabras, donde la variedad de textos es limitada, como por ejemplo, anuncios de salidas de trenes o información meteorológica.

El segundo método para la síntesis de voz es el conocido como la síntesis de formantes. Consiste en utilizar los valores de la frecuencia, sonoridad y ruido para poder crear la onda de voz artificial. El resultado es una señal de voz un tanto mecánica o robótica, mucho menos natural que lo formada por síntesis concatenativa, pero que puede ser de mayor utilidad para los desarrolladores, por ejemplo, para la implementación de un lector de pantalla en el que se requiera acelerar la velocidad de lectura sin perder demasiada inteligibilidad.

Uno de los sintetizadores gratuitos más conocidos, es el llamado Festival (http://www.cstr.ed.ac.uk/projects/festival/). Se trata de un sintetizador distribuido como software libre y desarrollado por el Centro de Investigación de Tecnologías del Lenguaje de la Universidad de Edimburgo, y la Universidad Carnegie Mellon. El proyecto está escrito en C++.

2.2.6 Ejemplo de interacción con el sistema de diálogo

A continuación vamos a describir de manera muy abreviada el funcionamiento de cada uno de los módulos descritos utilizando un ejemplo concreto de un sistema de

dialogo que nos ofreciera información sobre los horarios de salida de trenes. Tenemos un usuario que realiza esta petición a nuestro sistema:

"Hola mmm, quería saber a qué hora sale la… el tren, eh… hacia Barcelona, desde Madrid"

Esta frase será detectada por el módulo de reconocimiento del habla**,** que se encargará de transformar los sonidos y las ondas de voz obtenidas en texto. Utilizando las cadenas de texto obtenidas por el modulo anterior, el módulo de análisis lingüístico se encargará analizar cada una de estas palabras y darles el significado al que se refieren, resultando la salida de este módulo de la siguiente manera (traducción a *frames* semánticos):

PETICIÓN

DESTINO="Barcelona"

ORIGEN="Madrid"

A partir de esta salida, el gestor de diálogo decide que se requiere información adicional para poder completar la petición del usuario, en este caso, la fecha en la que se va a realizar el viaje:

PREGUNTAR_FECHA

El módulo generador de respuestas genera a continuación la pregunta en lenguaje natural correspondiente a la acción seleccionada por el gestor de diálogo y el sintetizador de voz la transmite oralmente al usuario:

"¿Qué día y a qué hora quiere salir?"

El usuario responde a esta consulta para poder completar la información requerida por el sistema:

"El último tren que salga el sábado"

El sistema detectará esta frase, el módulo de reconocimiento del habla, la transcribirá y el analizador lingüístico se encargará de detectar la información relevante para el dominio del sistema:

FECHA=”Sábado”

HORA=”Último”

A partir de todos estos datos, el gestor de diálogo realiza una consulta a la base de datos del sistema para generar una respuesta que sea coherente con los requisitos proporcionados por el usuario. Posteriormente y una vez decidida la respuesta, el generador de respuestas generaría una frase en lenguaje natural informando al usuario del resultado de su consulta:

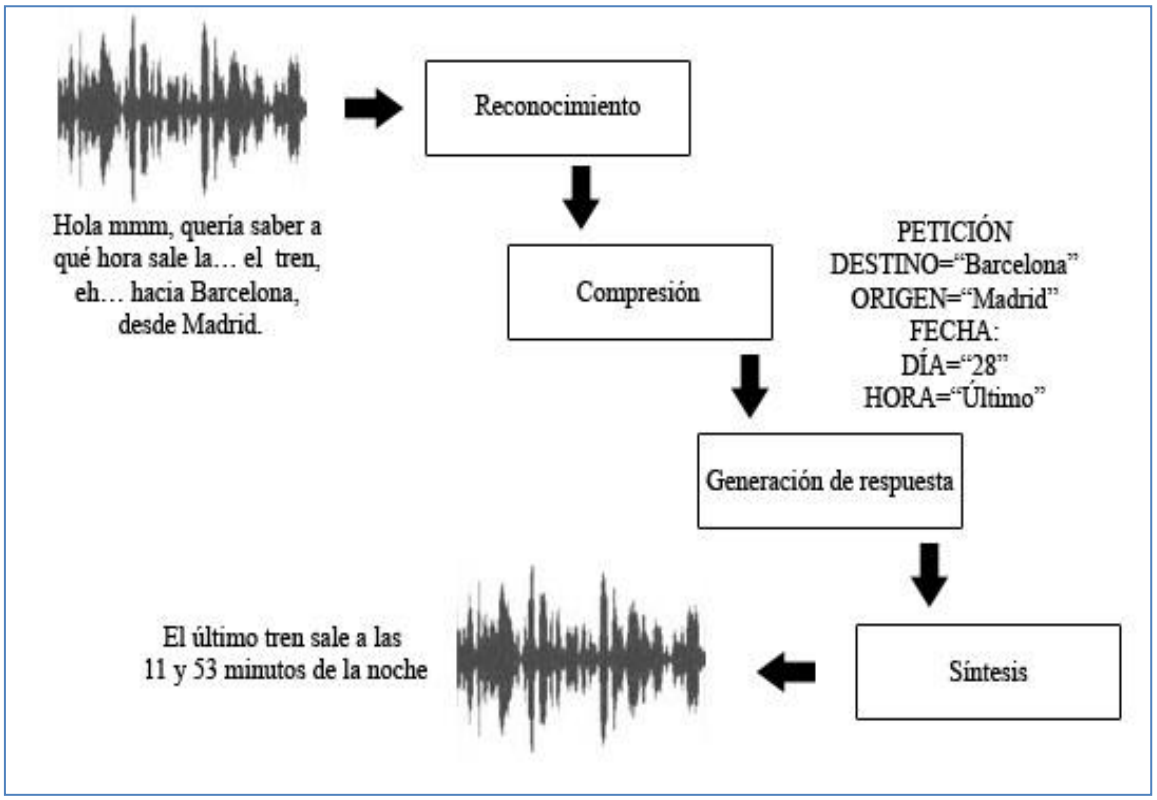


Figura 6. Representación esquemática del proceso de consulta a un sistema de diálogo

Podemos apreciar en la Figura 6 cómo funciona gráficamente nuestro ejemplo de sistema de diálogo, como a partir de la elocución del usuario el sistema es capaz de reconocer la elocución del usuario, comprenderla e interpretarla para generar una respuesta coherente y transmitida al usuario en forma de señal de voz.

2.3 Historia de los sistemas de diálogo

A continuación daremos unas pinceladas a la historia de los sistemas de diálogo, desde sus orígenes en los siglos XVIII y XIX hasta los sistemas desarrollados actualmente.

2.3.1 Orígenes de los sistemas de diálogo

Durante siglos la humanidad ha imaginado la idea de poder comunicarse de forma oral con un sistema inteligente que pudiera mantener una conversación prolongada con un individuo. Durante los siglos XVIII y XIX se llevaron a cabo los primeros intentos científicos de conseguir sistemas que trataran de imitar el habla humana, estos se basaban mayoritariamente en el diseño de autómatas. En el año 1770 el barón Von Kempelen creó el primer autómata que era capaz de generar frases y palabras completas, y que posteriormente el investigador Josef Faber estudió para mejorarlo y crear la conocida como máquina Euphonia en el año 1857. Estos primeros sistemas utilizaban métodos mecánicos y no fue hasta finales de siglo XIX cuando salieron a la luz los primeros sistemas que utilizaban y generaban voz de forma eléctrica.

Durante los primeros años del siglo XX J.Q. Stewart desarrolló una máquina que generaba los sonidos vocálicos eléctricamente, y durante los años 30 se desarrollaron los primeros sistemas eléctricos que completaban el resto de sonidos. A principios de los años 40 científicos como Allan Turing desarrollaron las bases para las primeras computadoras y utilizaron su potencial para la producción de los primeros sistemas inteligentes.

Beneficiándose de las mejoras que se desarrollaban en las áreas del Reconocimiento del Habla, el Procesamiento del Lenguaje Natural y de la Síntesis del Habla, las primeras iniciativas de investigación relacionadas con los sistemas de diálogo oral tal y como los conocemos ahora surgen a principios de los años 80. El origen de esta área de investigación está ligado a dos grandes proyectos: el programa DARPA (Hirshman, 1989) Spoken Language Systems en los E.E.U.U. y Esprit SUNDIAL (Peckham, 1993) en Europa.

Uno de los principales objetivos del proyecto DARPA fue el estudio y el desarrollo de tecnologías basadas en el reconocimiento del habla y de la síntesis de voz bajo el dominio de la reserva de vuelos mediante vía telefónica que denominaron Air Travel Informatión Services. Una de las partes del sistema ATIS, en concreto el corpus de diálogo, todavía es utilizado por algunos de los desarrolladores de sistemas de diálogo, este corpus de diálogo favoreció la aparición de nuevos proyectos como los desarrollados por AT&T.

A nivel europeo, el proyecto SUNDIAL trataba con información sobre horarios de trenes o aviones en diferentes idiomas. Gracias a la investigación realizada en SUNDIAL otros proyectos tuvieron el punto de partida y de financiación por la Comunidad Europea. Estos proyectos eran relativos principalmente al modelado de diálogo, como por ejemplo, VERMOBIL (Klein at al., 1999) o DISC (Dybkjær y Bernsen, 1997).

Las líneas de investigación más importantes durante los años 90 estuvieron relacionadas con la tasa de éxito de los diferentes módulos de los sistemas de diálogo. En un principio, con respecto al reconocimiento del habla la mayor preocupación fue la robustez. Los investigadores dedicaron sus esfuerzos a estudiar la degradación de funcionamiento repentina que experimentaban los sistemas debido a cambios de menor importancia como variaciones en los micrófonos o en los canales de comunicación, e indicaron que la tecnología utilizada en esos años no era capaz de proveer soluciones adecuadas. En Europa, la Acción COST 249, con un equipo investigador proveniente de más de 20 países europeos y desarrollada entre 1994 y 2000, investigó el reconocimiento continuo del habla recibida vía telefónica, tratando todas las temáticas mencionadas anteriormente, además de la selección de modelos acústicos, métodos de clasificación fonéticos y adaptación a las características propias del canal telefónico.

La aparición de de los teléfonos móviles hizo que se tuvieran que dedicar parte de los esfuerzos en la investigación para adaptar los sistemas a la nueva telefonía. Los nuevos sistemas requerían la capacidad de trabajar con anchos de banda estrechos y bajas relaciones señal-ruido. Además, la incorporación de los teléfonos móviles implicó

la necesidad de estudiar una gran cantidad de nuevos entornos desde donde los usuarios podían interaccionar con los nuevos dispositivos, con una robustez cada vez más exigente para poder gestionar la comunicación en ambientes muy ruidosos. Por este motivo, uno de los estudios principales del sistema SDR TREC-8 en el LIMSI (Computer Sciences Laboratory for Mechanics and Engineering Sciences) trató el desarrollo de técnicas de reducción de ruido (Gauvain et al, 2002).

Durante los últimos diez años se han venido desarrollando e investigando aplicaciones y lenguajes que puedan utilizar la voz para su manejo. A comienzos del año 2000 se desarrolló la primera versión del lenguaje VoiceXML (W3C, 2004), que más adelante, a finales del año 2001, se combinará con XHTML para crear XHTML+Voice (W3C, 2001). Al mismo tiempo, se desarrollaron aplicaciones unimodales como "BusLine" (http://www.speech.cs.cmu.edu/BusLine/), que utilizaba la voz para dar información a los usuarios acerca de horarios de autobuses, Oackland y Squirrel Hill o "CTT-Bank", aplicación desarrollada por el CTT sueco (Centre of Speech Technology) para acceder a los sistemas bancarios telefónicamente autentificando al usuario mediante su voz.

Poco a poco fueron surgiendo los primeros sistemas multimodales. Estos sistemas combinan diferentes modalidades, principalmente visuales y la voz, para el manejo de la aplicación. Ejemplos de estos sistemas son "Communicator" (Stallard, 2000), desarrollado por AT&T en EE.UU. para tramitar reservas de vuelos, hoteles e incluso el alquiler de coches, o el sistema "Intelligent Procedure Assistant" (IPA) (Aist, et al., 2003), que proveía soporte a los astronautas de la estación espacial internacional (ISS) durante la comprobación de los sistemas de la estación, desarrollado por RIALIS Group, empresa perteneciente a la NASA.

2.3.2 Retos Actuales

Una de las tendencias actuales en la industria para el desarrollo de sistemas de diálogo consiste en crear bloques reutilizables y aplicaciones que permitan una fácil adaptación a nuevas tareas. La reusabilidad de estos componentes reduce

notablemente los costes y los riesgos del desarrollo de aplicaciones y, al mismo tiempo, simplifica el diseño de aplicaciones más sofisticadas.

El constante crecimiento de las conexiones inalámbricas y de los terminales móviles hace que cada día se programen más aplicaciones para estos dispositivos. El problema es que estos dispositivos cada vez son más pequeños, mientras que nuestros dedos siguen con el mismo tamaño. Es ésta una de las principales razones que han impulsado combinar el uso del teclado con el de la voz para el manejo de estos dispositivos.

Las investigaciones actuales van también dirigidas hacia al desarrollo de sistemas multimodales que no sólo utilicen la voz, la vista o el sonido para interactuar con el usuario sino que sean capaces de reconocer emociones y estados de ánimo del usuario, y adaptar su funcionamiento según los mismos. Las aplicaciones multimodales tienen como principales objetivos conseguir un manejo más rápido y eficiente de las mismas, reducir la tasa de errores del usuario debido a que éste pueda elegir la forma que más le convenga de manejar la aplicación, así como facilitar nuevas modalidades de acceso para usuarios con discapacidades que tuviesen dificultades para el uso y manejo de estas interfaces.

2.4 El estándar VoiceXML

Con la descripción y ejemplos suministrados de los sistemas de diálogo, hemos podido comprobar que existen ciertas similitudes con los tradicionales formularios web en los que el sistema espera una elección o respuesta del usuario para continuar con su ejecución.

Esta respuesta normalmente está acotada a un conjunto posible de opciones, aunque también pueden ser respuestas abiertas. Si es acotada, se obliga al usuario a elegir una de estas opciones para poder continuar. Sucede algo parecido cuando utilizamos un sistema de diálogo, éste nos realizará una pregunta, y no continuará hasta que no recibe del usuario una respuesta que esté dentro de las permitidas en cada instante del diálogo.

Para controlar que las respuestas introducidas por el usuario son las correctas, el sistema de diálogo suele utilizar gramáticas. Estas gramáticas se componen de todas las palabras posibles (y combinaciones de las mismas) que el usuario puede pronunciar para responder la pregunta del sistema, por lo que el sistema esperará a detectar una de estas combinaciones válidas para pasar al siguiente punto de su ejecución.

El lenguaje VoiceXML utiliza este tipo de métodos. Se compone de campos ligados a una serie de gramáticas y a unas locuciones o "prompts", en las que el sistema pregunta al usuario y el usuario debe responder con una de las opciones permitidas por las gramáticas. De esta forma, el funcionamiento de estos sistemas es muy similar al funcionamiento de los formularios web tradicionales.

2.4.1 Introducción

Tal y como hemos introducido previamente, VoiceXML es un lenguaje basado en XML con el objetivo de ser utilizado para la creación de sistemas de diálogo que utilicen voz sintetizada y reconocimiento de voz. Al igual que los documentos de HTML son interpretados por un navegador web, los documentos de VoiceXML deben ser interpretados por un navegador de voz, que puede estar incluido en el navegador web.

Los orígenes de VoiceXML se remontan a 1995 con el objetivo de simplificar la aplicación de reconocimiento de voz dentro de un proyecto de AT&T denominado “Phone Markup Language” (PML). Al igual que AT&T, otras empresas como Lucent o Motorola, también trabajaban en proyectos relacionados, aunque utilizando sus propios lenguajes de desarrollo.

En 1998, el World Wide Web Consortium (W3C) organizó una conferencia sobre navegadores de voz. Para entonces, AT&T y Lucent tenían distintas variedades de sus lenguajes PML, mientras que Motorola había desarrollado su propio lenguaje “VoxML”, IBM (http://www-01.ibm.com/software/pervasive/embedded_viavoice) había desarrollado “SpeechML”, “TalkML” de HP o “VoiceHTML” de PipeBeach.

Gracias a la unión de fuerzas de AT&T, Lucent, IBM y Motorola se formó el VoiceXML Forum (www.voicexml.org/). La misión de este foro era desarrollar un lenguaje de diseño estándar de diálogo que los desarrolladores pudieran utilizar para

construir aplicaciones de diálogo. Eligieron XML como base para este esfuerzo ya que quedaba claro que esa era la dirección en que se iba a mover la tecnología en Internet.

En el año 2000 el VoiceXML Forum lanza la primera versión del lenguaje, VoiceXML 1.0 (www.w3.org/TR/voicexml/). Poco después VoiceXML se presentó a la W3C como base para la creación de una nueva norma internacional. VoiceXML 2.0 (www.w3.org/TR/voicexml20/) y VoiceXML 2.1 (www.w3.org/TR/voicexml21/) son el resultado del trabajo conjunto entre empresas colaboradoras con el W3C, universidades y desarrolladores. Actualmente se ha lanzado la última versión de este lenguaje, VoiceXML 3.0 (www.w3.org/TR/voicexml30/), con el objetivo principal de mejorar la extensibilidad, posibilidades y funcionalidades de este lenguaje.

VoiceXML describe la interacción persona-máquina que ofrecen los sistemas de diálogo, incluyendo:

- La generación de síntesis de voz.
- La reproducción de archivos de audio.
- Reconocimiento de la entrada de voz.
- Reconocimiento de pulsaciones de las teclas del teléfono o marcación por tonos (DTMF, Dual-Tone Multi-Frequency).
- Grabación de la entrada de voz.
- Control del flujo de diálogo.
- Funciones de telefonía como transferencia de llamada y de desconexión.

2.4.2 Motivaciones y objetivos

El objetivo principal de VoiceXML es llevar todo el poder de las aplicaciones web a aplicaciones accesibles mediante la voz, con el objetivo de liberar a los desarrolladores de la programación a un bajo nivel que supondría no contar con este lenguaje.

VoiceXML permite la integración de servicios de voz con servicios de datos utilizando el paradigma familiar de cliente-servidor. Un servicio de voz es visto como una secuencia de estados de diálogo que posibilitan la interacción entre el usuario y la

plataforma de la aplicación. Estos diálogos son proporcionados por servidores que pueden ser externos a la plataforma de la aplicación y que se encargan de mantener la lógica general del servicio, de realizar operaciones con la base de datos y de guardar los historiales de seguridad.

Un documento VoiceXML especifica además que intérprete tiene que llevar a cabo cada interacción del diálogo. La interacción con el usuario afecta la interpretación del diálogo y se recoge en las solicitudes presentadas al servidor de documentos. Este servidor deberá responder con otro documento de VoiceXML para continuar el diálogo con el usuario. El funcionamiento básico consiste, por tanto, en recoger la entrada del usuario (voz o datos), asignar esta entrada a variables y tomar decisiones en función de ellas, estableciéndose enlaces con documentos VoiceXML, como muestra la Figura 7.

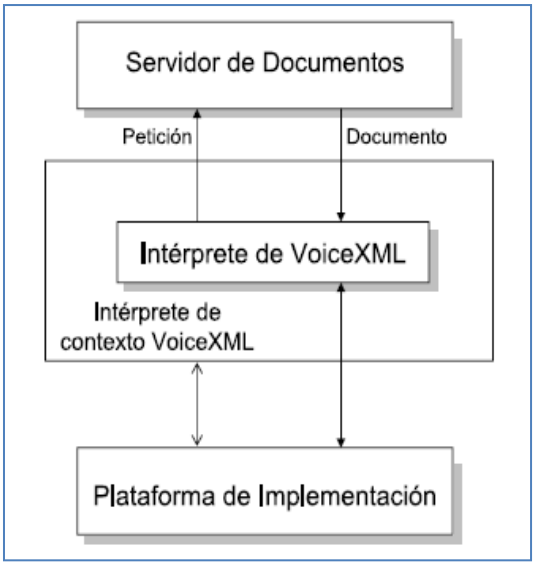


Figura 7. Arquitectura de un sistema VoiceXML

Por lo tanto VoiceXML es un lenguaje de etiquetas que:

- Minimiza la transmisión de información entre cliente/servidor mediante la especificación de interacciones múltiples por cada documento.
- Evita a los autores el uso de aplicaciones de bajo nivel, y de los detalles específicos de la plataforma.
- Separa el código de la interacción con el usuario (en VoiceXML) de la lógica de servicios (por ejemplo, Scripts CGI).
- Promueve el servicio de portabilidad entre las plataformas de ejecución.

VoiceXML es un lenguaje común entre los proveedores de contenidos, proveedores de herramientas y proveedores de plataformas.

- Es fácil de usar para interacciones simples, y sin embargo ofrece un amplio abanico de características para apoyar interacciones de diálogos más complejas.

Los principios de diseño de VoiceXML incluyen:

- Estandarización (portabilidad).
- Permitir diversos tipos de plataforma gracias a los diferentes formatos de ficheros soportados.
- Reconocimiento semántico a través de gramáticas.
- Proveer diferentes opciones al programador.

2.4.3 Estructura y funcionamiento

Un documento VoiceXML puede describirse como una máquina de estados finitos. El usuario está siempre en un estado de la conversación, indicándose el siguiente estado que le continúa mediante la definición del próximo documento VoiceXML a utilizar utilizando URIs (**U**niform **R**esource **I**dentifier). La ejecución termina cuando un documento no especifica su sucesor o si especifica directamente el fin del diálogo.

Tal y como se describe en la subsección siguiente, existen dos tipos de diálogos en VoiceXML: los formularios y los menús. En los formularios se realiza una interacción que recoge los valores de un conjunto de variables relacionadas con dicho formulario. Cada campo puede especificar una gramática, que defina los valores que se esperan para ese campo, o podemos tener una gramática global al formulario que puede ser utilizada para rellenar varios campos del mismo. Un menú presenta al usuario directamente las opciones que pueden seleccionarse, efectuándose a continuación transiciones sobre la base de esa elección.

Un subdiálogo es equivalente a una llamada a una función, ya que proporciona un mecanismo para invocar a una nueva interacción y después volver al punto en el que se realizó la llamada. Variables, gramáticas e información de estado se guardarán y

estarán disponibles al regresar al documento de llamada. Un subdiálogo puede utilizarse, por ejemplo, para crear una secuencia de confirmación que puede requerir una consulta a una base de datos, especificar recursos que pueden ser compartidos entre los documentos en una sola aplicación, o incluso generar una biblioteca reutilizable de diálogos para compartir entre varias aplicaciones.

Otros conceptos importantes son:

- Sesiones: Desde que el usuario comienza a interactuar con el Intérprete de Contexto VoiceXML hasta que finaliza la interacción.

- Aplicaciones: Una aplicación es un conjunto de documentos que comparten el mismo documento raíz (*Root*). El *Root* se carga cuando el usuario interactúa con cualquier documento de la aplicación y está disponible hasta que el usuario cambie de aplicación.

- Gramáticas: Cada diálogo posee una o varias gramáticas de voz o de DTMF. Las gramáticas pueden estar activas aunque los diálogos asociados no lo estén.

- Eventos: Se definen mecanismos para tratar eventos como que el usuario no responda, falta de inteligibilidad, solicitud de ayuda, etc. Cada evento define su función de tratamiento.

- Enlaces: Permiten especificar gramáticas, transferir el control a otra URI, etc.

2.4.4 Constructores de Diálogo

2.4.4.1 Formularios

Nos referimos a los formularios como el componente fundamental de los documentos VoiceXML. Éstos contienen:

- Campos de entrada (ítems) y de control.
- Declaración de variables.

- Tratamiento de eventos.
- Acciones a ejecutar cuando se completen determinados campos.

Se definen dos tipos de formularios siguiendo el criterio del modo de interpretación:

- **Formularios directos**: La estrategia se basa en ir recorriendo los campos de forma secuencial, eludiendo aquellos que se van completando adecuadamente. La Figura 8 muestra un ejemplo de un formulario VoiceXML (www.evolution.voxeo.com). Para realizar la interpretación de los formularios se utiliza el algoritmo FIA (Form Interpretation Algorithm), que recorre el formulario almacenando el contenido de los campos, reproduce los *prompts* asociados al formulario y comprueba si la condición *<filled>* (acciones a ejecutar cuando un determinado campo se completa) del formulario se cumple.

```
<form id="F1" scope="document">
<grammar scope="document" type="text/gsl">
 <![CDATA[
.MYRULE
      [[(david hasselhoff)] {<MySlot "dave">}]]]>
        </grammar>
<!-- the utterance of 'david hasselhoff' anywhere in the
  application will fill this namelist-->
<filled namelist="MySlot">
        <goto next="#F3"/>  </filled>
<field name="F_1"> //Se define F1 como un campo de entrada
<grammar type="text/gsl">[(moe green)]</grammar>
      <prompt>
        Who should get the next crack at playing Hamlet ?
        His initials are d h .
      </prompt>
</field> <filled namelist="F_1"> //Se define lo que realiza el
sistema cuando rellena el campo F1
        <prompt>
          You said <value expr="F_1"/> Are you insane?
        </prompt>
        <goto next="#F2"/>
      </filled>

</form>
```

Figura 8. Código de ejemplo de formulario directo en VoiceXML

- **Iniciativa mixta**: Definiendo *form-level grammars* (gramáticas definidas externamente al formulario), las entradas se pueden completar en cualquier orden y se pueden rellenar varios campos en una única iteración. Además, pueden haber varias gramáticas correspondientes a formularios diferentes activas al mismo tiempo, determinando el sistema a qué campo del formulario se ha dado respuesta.

En un formulario pueden aparecer dos tipos de campos:

- **Ítems de entrada**: Incorporan *prompts* para informar al usuario sobre qué debe hacer, gramáticas que determinan si la entrada es válida o no y tratamiento de eventos. Se añade la acción *<filled>* para incorporar la acción a realizar cuando una variable de entrada se completa adecuadamente.

    - **Ítems de control**: Se trata de las etiquetas *<block>* (presenta información y no pide entrada de datos) e *<initial>* (bloque inicial del formulario para presentar información o ayuda sobre el mismo).

Cada campo del formulario tiene asociada una variable, que al final contendrá el resultado de interpretar el formulario y que se puede referenciar a lo largo de la aplicación.

2.4.4.2 Menús

Se trata del caso más simple de formulario, incluyendo un único campo anónimo para forzar al usuario a realizar una elección y realizar una transición en función de ésta. Las opciones se especifican mediante la etiqueta *<choice>*, donde se indica una gramática de voz y/o de DTMF y un enlace a acudir cuando se selecciona la alternativa. El atributo *accept* se utiliza para determinar si el usuario debe indicar la opción completa *(exact)* o se acepten partes de la misma *(approximate).*

Respecto al modelo de interpretación, el funcionamiento es idéntico al de un formulario con un único campo que se encarga de realizar todo el trabajo,

estableciéndose las indicaciones correspondientes al enlace mostrado en el objeto <*choice*>. La Figura 9 muestra un ejemplo de menú VoiceXML (www.evolution.voxeo.com).

```
    <menu id="M1" scope="dialog" dtmf="true">
    <prompt>
      Who is the most infamous celebrity in the weekly world
news tabloid?
    </prompt>
    <prompt>
      <enumerate>
        For <value expr="_prompt"/>, press <value expr="_dtmf"/>
      </enumerate>
    </prompt>

  <choice event="bigfoot_event"> //Se especifican las opciones
    lie za manelly
  </choice>
  <choice event="bigfoot_event">
    bride of bigfoot
  </choice>
  </menu>
```

Figura 9. Ejemplo de menú de VoiceXML

2.4.5 Gramáticas

El mecanismo elegido por VoiceXML para introducir un modelo de lenguaje se basa en la utilización de gramáticas, definidas mediante el objeto <*grammar*>. Una gramática determina la secuencia de palabras aceptables durante el proceso de reconocimiento automático del habla, y asimismo, incluye en el proceso de comprensión de las frases, pues indica la interpretación semántica de unidades sintácticas de las mismas.

En mayo de 2004 el W3C editó una serie de recomendaciones sobre el formato de gramáticas para el reconocimiento de voz denominadas Speech Recognition Grammar Specification (W3C, 2004). Este documento puede verse como un complemento a las especificaciones de VoiceXML, tratando los casos de entradas de voz y entradas en formato DTMF.

En especificaciones SRGS se proponen dos formatos fundamentales de gramáticas:

- ABNF (Augmented BNF): Se trata de un formato en texto plano basada en formatos anteriores como el BNF (Backus Naur Form) y JSGF (Java Speech Grammar Format).

- XML: Se utiliza el formato XML, delimitando los diferentes campos entre las marcas correspondientes.

Ambos formatos son independientes del contexto y son semánticamente equivalentes, desde el punto de vista que aceptan el mismo lenguaje como entrada y procesan cadenas de entrada de forma idéntica. En VoiceXML también se contemplan modelos de n-gramas, introducción de medidas de confianza y métodos de aplicación de suavizado como *Back-Off*, basado en el "rellenado" de la matriz de la gramática, sustituyendo los ceros de probabilidad existentes por no disponer de muestras de entrenamiento por valores adecuados.

Las gramáticas pueden ser introducidas de forma interna o externa. Las gramáticas internas se incluyen directamente en la etiqueta *<grammar>.* Las gramáticas externas se incluyen a través de URIs.

El peso de las gramáticas se indica mediante el atributo *weight* siguiendo el formato indicado en el SRGS, estableciéndose de este modo una jerarquía entre las diferentes gramáticas que conforman un formulario.

Las gramáticas internas a los campos de entrada sólo están activas cuando se ejecuta el campo correspondiente. Las gramáticas por enlaces poseen el alcance del elemento que engloba el link. Las gramáticas asociadas a formularios poseen el alcance del formulario, del documento o de la aplicación completa (caso de estar englobadas en el documento ROOT).

Las gramáticas asociadas a menús poseen como alcance por defecto el del diálogo correspondiente, pero pueden darse los mismos casos que para las gramáticas del formulario.

Un formulario puede tener activas varias gramáticas de forma simultánea y desactivar otro conjunto. Cuando el intérprete permanece a la espera de una entrada tras visitar un campo, las gramáticas activas son:

- Gramáticas asociadas a dicho campo, inclusive la de los campos link.
- Gramáticas del formulario.
- Gramáticas contenidas en links dentro del documento.
- Gramáticas contenidas en el documento *Root* (enlaces, menús, formularios).
- Gramáticas definidas en la captura de eventos de la plataforma (ayuda, salida, cancelar).

El orden de precedencia es el mostrado. Si existe más de una gramática activa del mismo nivel, la precedencia la marca el orden de los documentos. Como mínimo debe haber una gramática activa cuando se espera la respuesta de usuario. Se pueden desactivar gramáticas, de forma que simplemente éste vigente la del formulario en cuestión.

La interpretación semántica para el reconocimiento viene descrita en la especificación *Semantic Interpretation for Speech Recognition* (W3C, 2007). El resultado de la interpretación semántica debe trasladarse a variables ECMAScript, funcionalidad que debe ser soportada por el reconocedor. Cada campo de entrada tiene asociado un *slot name*, que almacena el resultado del reconocimiento para determinar si la entrada suministrada concuerda o no con la gramática.

En caso de gramáticas asociadas a formularios, el resultado de reconocer adecuadamente uno o varios campos supone la asignación de las variables ECMAScript y slots correspondientes. Las gramáticas asociadas a campos concretos sólo se activan cuando el algoritmo visita dicho campo.

2.5 Plataforma Voxeo

2.5.1 Introducción

Voxeo Corporation (www.voxeo.com) es una de las empresas líderes en el sector de los sistemas de diálogo y en concreto en IVR (Interactive Voice Response) y

soluciones a través de voz sobre IP (VoIP) gracias a sus servidores y sistemas de desarrollo, apoyados en las tecnologías de VoiceXML y CCXML.

Para la realización de la aplicación se ha utilizado la plataforma de desarrollo gratuita Voxeo Evolution (evolution.voxeo.com). Gracias a esta plataforma podemos disponer de los módulos propios de un sistema de diálogo, y además nos ofrece un servicio con un número de llamada gratuito para que podamos probar el sistema desarrollado tantas veces como queramos. En la siguiente subsección se describe brevemente cómo crear una aplicación con Voxeo Evolution.

2.5.2 Utilización de Voxeo Evolution

Para poder comenzar con el desarrollo de nuestra aplicación en la plataforma de Voxeo, comenzaremos por entrar en su web, como vemos en la Figura 10, en concreto en la parte en la que esta publicada su herramienta para desarrolladores. Una vez dentro comenzaremos por registrarnos como usuarios para poder tener acceso privado a nuestra aplicación.

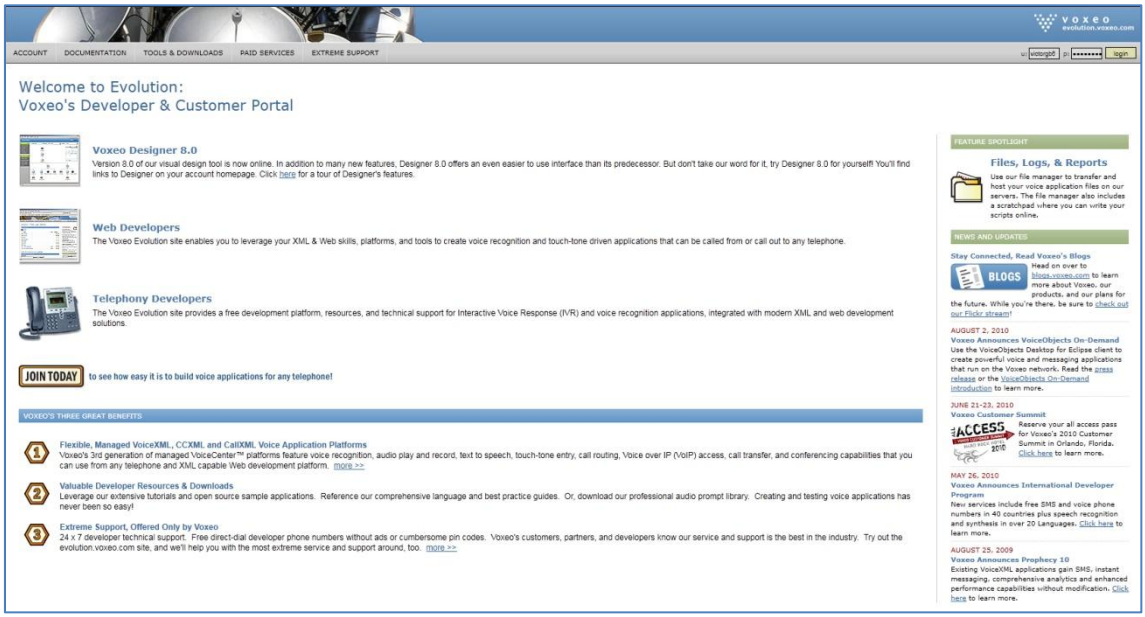


Figura 10. Página principal de la herramienta Voxeo

Cuando completemos el registro obtenemos acceso privado para la creación de nuestra aplicación, además tendremos un pequeño espacio de alojamiento web, como vemos en la Figura 11, en el que sólo es posible almacenar ficheros con extensión XML,

gramáticas o ficheros de texto. Dentro de este espacio podremos encontrar las carpetas en las que se guardaran posibles grabaciones de locuciones que utilicemos en nuestra aplicación, así como los "logs" que generará el sistema de las llamadas realizadas a la aplicación.

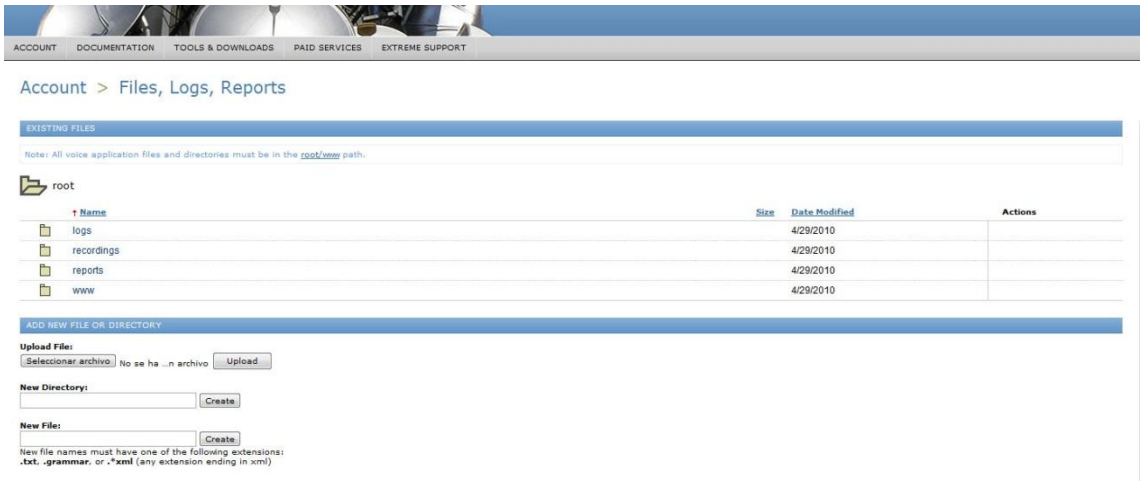


Figura 11. Gestor de archivos de la plataforma Voxeo

Para crear nuestro primer sistema accederemos con nuestra cuenta y entraremos en el menú "Account", seleccionando la opción "Application Manager". Una vez ahí, y tal y como muestra la Figura 12, crearemos una aplicación seleccionando el tipo de aplicación correspondiente e indicando la ruta del fichero principal que inicia la interacción con el usuario. Voxeo Evolution asigna gratuitamente un número de teléfono local, así como otras vías para contactar telefónicamente con la aplicación, tales como Skype o su propio teléfono interno.

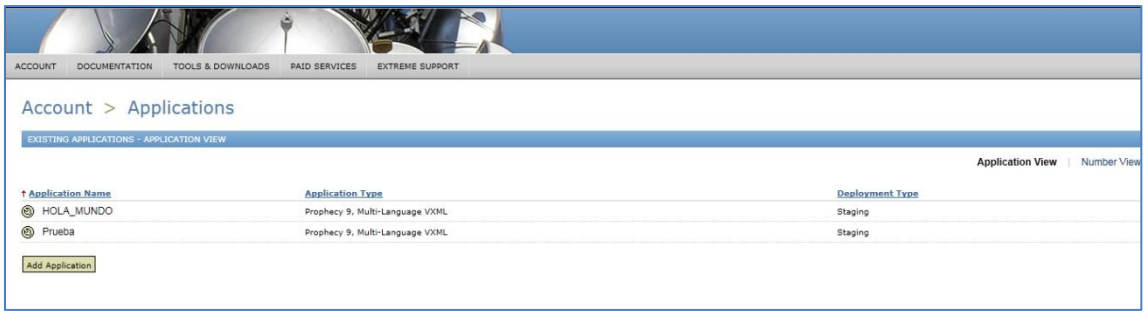


Figura 12. Sección de aplicaciones de la herramienta Voxeo

Voxeo Evolution ofrece además servicios de depuración de código, para tratar de detectar los posibles errores que genere la interacción con la aplicación. Además, disponemos de un soporte técnico, que nos ofrece gran ayuda cuando tenemos

problemas con el desarrollo de nuestras aplicaciones. La Figura 13 muestra una captura de pantalla de "*Application Debugger*", herramienta que nos ofrece Voxeo para la depuración de nuestras aplicaciones.

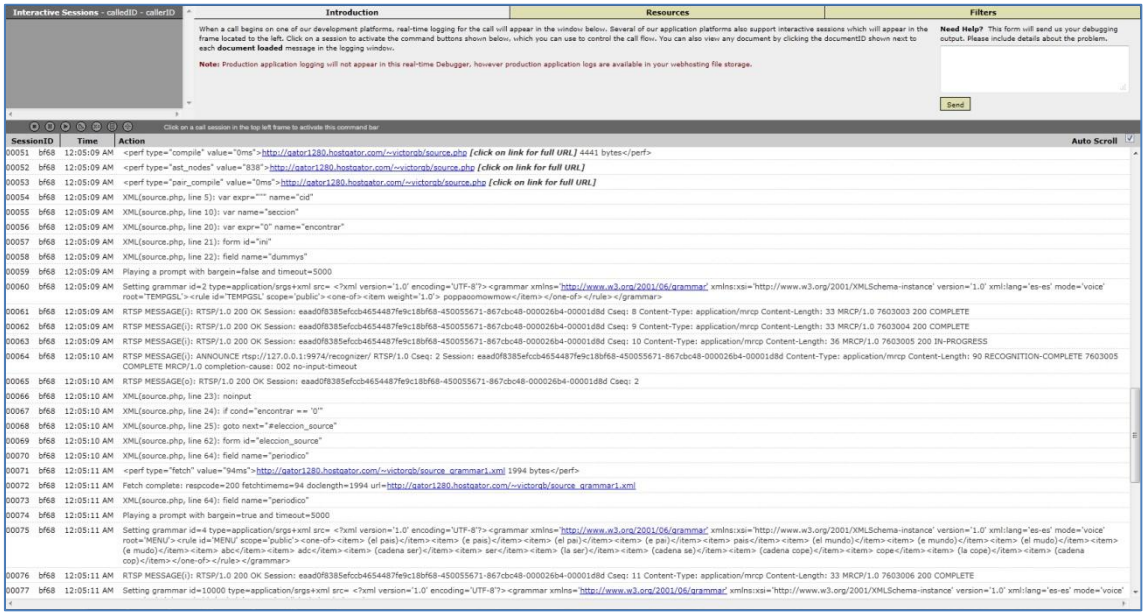


Figura 13. *Application Debugger* de la herramienta Voxeo

Como hemos visto las herramientas que nos ofrece Voxeo Corporation son excelentes si queremos iniciarnos en el desarrollo de sistemas de diálogo accesibles telefónicamente, y todo ello a coste cero.

Capítulo 3

Descripción general de la aplicación desarrollada

En este capítulo se detallan las principales características de la aplicación desarrollada para el Proyecto, así como principales sus componentes. Se describe la especificación de requisitos, tanto funcionales como no funcionales, el diagrama de casos de uso y un resumen de las funcionalidades de la aplicación.

3.1 Especificación de requisitos software

A lo largo de este apartado se describen los diferentes requisitos software que nos permitirán reconocer las funcionalidades y las características que presenta la aplicación desarrollada.

Los requisitos funcionales son los siguientes:

- **RF1: Elección de fuentes:** El usuario debe ser capaz de elegir la fuente de información de la que quiere obtener las noticias, en nuestro caso periódicos y radios de ámbito nacional
- **RF2: Elección de categorías:** El usuario debe ser capaz de elegir una categoría dentro de las fuentes elegidas anteriormente, estas categorías se definieron como: nacional, internacional, deportes, economía y cultura.
- **RF3: Elección de temas populares:** El usuario dispondrá de un listado de temas populares generados dinámicamente a partir de las noticias de la categoría seleccionada.
- **RF4: Controles avanzados de reproducción:** Durante la escucha de las noticias el usuario podrá controlar mediante la voz la reproducción de

las mismas utilizando opciones como: siguiente, anterior, repetir, ampliar y volver.

- **RF5: Detección de usuarios frecuentes:** Si un usuario ha utilizado la aplicación anteriormente, se almacenará el resultado de las interacciones apara darle la opción de seleccionar posteriormente su fuente y categorías favoritas.

Los requisitos no funcionales son los siguientes:

- **RNF1:** Las noticias de las fuentes deberán estar actualizadas constantemente con las últimas noticias ocurridas en el momento de la llamada.
- **RNF2:** Las fuentes de noticias podrán ser ampliables en el futuro, por lo que mediante la implementación de la aplicación, se facilitará la incorporación de nuevas fuentes de noticias, así como de nuevas funcionalidades y consultas.
- **RNF3:** La lectura de las diferentes noticias se realizará utilizando voz sintetizada con la mayor inteligibilidad posible.
- **RNF4:** La aplicación debe poder utilizarse mediante una llamada local desde cualquier teléfono, o con cualquier terminal con Skype, micrófono y altavoces.
- **RNF5:** En todo momento de la interacción se dará la posibilidad de ofrecer ayuda al usuario sobre los controles de reproducción avanzados, así como se le informará sobre el estado de la interacción.
- **RNF6:** Para el desarrollo de la aplicación se combinará el lenguaje VoiceXML con lenguajes de alto nivel pensados para la interacción en Internet, como es el caso del lenguaje PHP.
- **RNF7:** Los tiempos de carga de la aplicación deberán ser inferiores a dos segundos, para no dar la impresión al usuario de que se ha producido algún error en la ejecución de la aplicación.

3.2 Diagrama de casos de uso

La Figura 14 muestra el diagrama con los principales casos de uso de la aplicación desarrollada, en el que pueden observarse las principales funcionalidades ofrecidas al usuario.

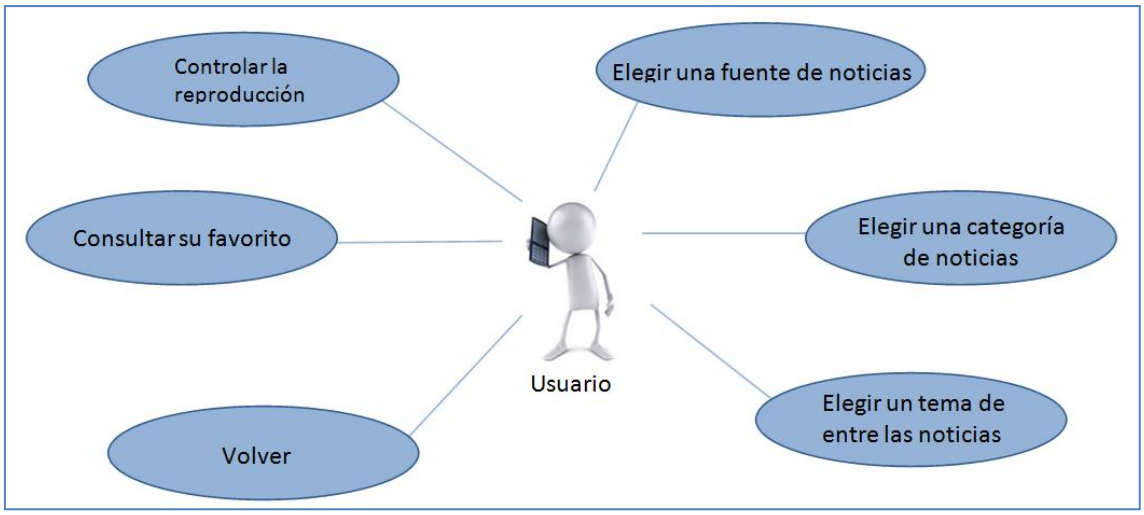


Figura 14. Diagrama de casos de uso de la aplicación desarrollada

Como vemos las acciones que el usuario puede realizar con el sistema son bastante variadas dentro del dominio de la aplicación: elección de cualquiera de las fuentes de noticias que se incluyen en la aplicación, selección de categorías de noticias, presentación de las categorías y noticias más populares de diarios nacionales y radios, así como adaptación personalizada teniendo en cuenta las preferencias y consultas anteriores de cada usuario.

Una vez que el usuario se encuentra escuchando las noticias se le ofrece la posibilidad de controlar la reproducción de las mismas, controlando si quiere avanzar o retroceder en las noticias o escuchar de nuevo las noticias que más interesantes le hayan parecido. Además, el usuario es capaz de moverse entre los diferentes menús hacia delante y hacia atrás, maximizándose la accesibilidad de la aplicación.

Por último, se ha pensado no solo en la primera vez que el usuario utilice la aplicación, sino en propiciar que los usuarios vuelvan a utilizarla. Para ello, se ha desarrollado una funcionalidad que detecta si el usuario ya ha utilizado la aplicación previamente y le propone, antes de elegir cualquiera de las opciones, si desea ir a la

sección que más veces ha consultado, por lo tanto, su sección favorita. Así conseguimos que el usuario acceda de manera más rápida, si ya está familiarizado con el sistema.

3.3 Arquitectura de la aplicación

Como vemos en la Figura 15, cuando el usuario llama a la aplicación está en realidad conectándose con la plataforma de Voxeo, funcionando a modo de navegador de voz. Este navegador, que se encargará de ejecutar la parte de código escrita en el lenguaje VoiceXML, interpreta la llamada y busca el archivo asociado al número de la aplicación, este archivo, y todos los que conforman la aplicación, se encuentran alojados en el servidor web externo, que será el encargado de ejecutar las partes escritas en el lenguaje PHP.

El primer módulo de los que se compone la aplicación, "*pre_source.php*", es el encargado de obtener el identificador de la llamada del usuario y continuar con el siguiente módulo, "*source.php*", en el que se requiere información al usuario para obtener la información sobre la fuente de *feeds*, de la que desea obtener las noticias.

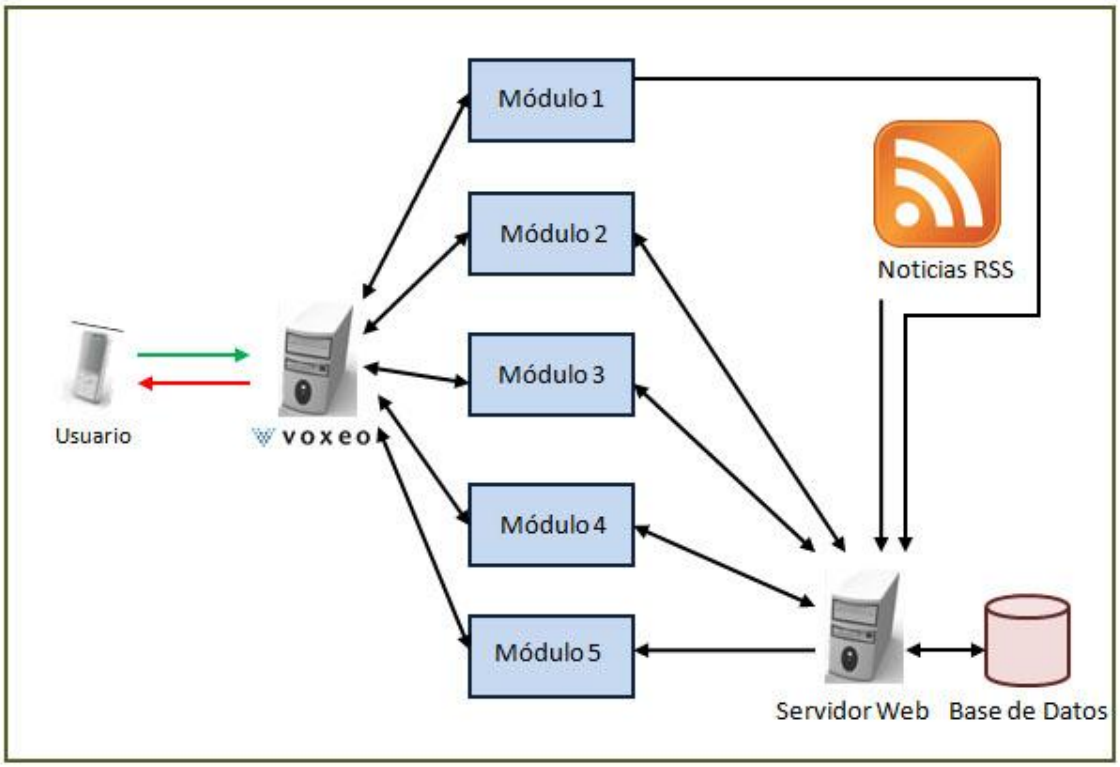


Figura 15. Arquitectura de la aplicación

A partir de esta información el servidor web obtendrá los *feeds* con las noticias deseadas y los almacenará en una base de datos SQL de manera temporal, tal y como se describe en la siguiente subsección. Una vez que se han obtenido las noticias deseadas, se pasará al siguiente módulo, "*lector.php*" Este módulo se encarga de analizar las noticias y, de manera dinámica, generar los temas generales y las gramáticas correspondientes para poder preguntar al usuario y que éste elija

Una vez que el sistema sabe el tema que quiere oír el usuario se pasa al siguiente módulo, "*lector2.php*". Este módulo es el encargado de generar el siguiente módulo "*lector_noticias.php*" de forma completamente dinámica para proceder a la lectura de las noticias, así como para el reconocimiento de los comandos avanzados de reproducción.

3.3.1 Estructura de la base de datos

La aplicación desarrollada utiliza dos bases de datos SQL como repositorio de la información que necesita almacenar la aplicación. En la primera de ellas se almacena de forma temporal las noticias, en la segunda se almacenan los enlaces a las *feeds* de las diferentes fuentes de noticias y un historial con las consultas realizadas por cada usuario (denotadas por el número de teléfono utilizado para interactuar con la aplicación).

La primera de las bases de datos de la aplicación, a la que llamaremos de ahora en adelante base de datos de noticias, es dinámica, por lo que dependiendo de las noticias que contenga el *feed* seleccionado se crearan más tablas o menos tablas. Esta base de datos de noticias se compone de un número dinámico de tablas, en las que al menos existirá una. El nombre de la base de datos será el de la fuente de donde se han obtenido las noticias, o el nombre del tema popular que se ha obtenido. Estas tablas contienen los siguientes campos:

- **Id:** Se trata de un campo numérico que asigna un número único a cada una de las entradas de la tabla, este campo se utiliza como clave primaria.

- **Title:** Se trata de un campo de tipo texto, en el que se almacenan los titulares de las diferentes noticias
- **Description:** Se trata de otro campo de tipo texto, en el que se guarda una breve descripción de la noticia.

La Figura 16 muestra una tabla de la base de datos con diferentes noticias de deportes extraídas de ficheros RSS del diario ABC. En la Figura 17 vemos una tabla con las noticias que hacen referencia al tema popular "Jorge Valdano".

	id	title	description
	1	Diarra se marcha al Monaco	El Real Madrid y el Monaco han acordado el traspas...
	2	29.000 euros de multa por alinear suplentes	La Premier League ha sancionado al Blackpool con u...
	3	¿Como sera el Ferrari F150?	A pocas horas del lanzamiento oficial del Ferrari ...
	4	«¿La hora del relevo? De ninguna manera»	Roger Federer, eliminado del Open de Australia tra...
	5	Horarios de los partidos de vuelta de las semifina...	Los dos partidos de vuelta de la semifinales de la...
	6	«Yo al indio no le vendo mis acciones, por si acas...	El Gobierno de Cantabria asistira este viernes en ...
	7	Valdano : «Mis funciones van a seguir siendo las m...	Jorge Valdano, director general del Real Madrid, n...
	8	Iniesta, protagonista del ultimo anuncio de Nike	El azulgrana Andres Iniesta se ha convertido en el...
	9	Adebayor : «No he venido a echar a Benzema»	Emmanuel Adebayor, nuevo delantero del Real Madrid...
	10	Djokovic arruina a Federer	Algo se mueve en Australia, atonita la grada ya qu...
	11	Recogepelotas a la fuga	Sucedio tras el pitido final del arbitro Undiano M...
	12	Busquets renueva con el Bari§a hasta junio de 2015	El centrocampista Sergio Busquets (Sabadell, 1988)...
	13	Guardiola : «La experiencia de Sevilla esta demasi...	El entrenador del Barcelona, Pep Guardiola, ha adv...
	14	Raul Albiol : «La saque en la linea»	El defensa del Real Madrid Raul Albiol nego que el...
	15	Manzano : «El gol, es gol, no hay tutia»	El tecnico del Sevilla, Gregorio Manzano, lamento ...
	16	Cristiano : «Que sigan con esas cosas de pintarse ...	El jugador del Real Madrid Cristiano Ronaldo afirm...
	17	El arbitro refleja en el acta el lanzamiento «de u...	El arbitro Undiano Mallenco, del colegio navarro, ...
	18	Karanka : «El equipo ha ganado, que es lo importan...	El segundo entrenador del Real Madrid, Aitor Karan...
	19	Clijsters regresa a la final, que disputara con Li	Siete años despues, la belga Kim Clijsters alcanzo...
	20	Valdano : «El club esta dispuesto a darle a Mourin...	El director general y adjunto a la presidencia del...

Figura 16. Ejemplo de tabla de noticias

	id	title	description
	7	Valdano : «Mis funciones van a seguir siendo las m...	Jorge Valdano, director general del Real Madrid, n...
	20	Valdano : «El club esta dispuesto a darle a Mourin...	El director general y adjunto a la presidencia del...

Figura 17. Ejemplo de tabla de noticias para un tema popular

Los contenidos de esta base de datos se incluyen de forma dinámica de acuerdo a las selecciones llevadas a cabo por el usuario en la interacción actual con el sistema.

La segunda de las bases de datos, a la que llamaremos de log o de historial, se compone de dos tablas. La primera de ellas es la tabla de enlaces, en la que se

almacenan los enlaces a los *feeds* de noticias de las diferentes fuentes que están contenidas en la aplicación. Esta tabla no se modifica durante la ejecución de la aplicación, sólo se tendría que modificar en el caso de querer añadir otra fuente de noticas, para guardar así los enlaces de la misma. Los diferentes campos que componen esta tabla son los siguientes:

- **Id:** Se trata de un campo numérico que da un número único a cada una de las entradas de la tabla, este campo se utiliza como clave primaria.
- **Fuente:** Se trata de un campo de tipo texto, en el que se guarda el nombre de la fuente de noticias.
- **Sección:** Se trata de un campo de tipo texto, en el que se guarda el nombre de la sección a la que corresponde el link.
- **URL:** Se trata de otro campo de tipo texto, en el que se guarda el enlace al *feed*, que corresponde con la fuente y la sección de noticias correspondiente.

La Figura 18 muestra un ejemplo de esta tabla.

	id	fuente	seccion	url
	44	abc	economia	http://abc.es/rss/feeds/abc_Economia.xml
	43	abc	deportes	http://abc.es/rss/feeds/abc_Deportes.xml
	42	abc	internacional	http://abc.es/rss/feeds/abc_Internacional.xml
	41	abc	nacional	http://abc.es/rss/feeds/abc_EspanaEspana.xml
	40	elmundo	cultura	http://www.estaticos.elmundo.es/elmundo/rss/cultur...
	39	elmundo	economia	http://www.estaticos.elmundo.es/mundodinero/rss/po...
	38	elmundo	deportes	http://www.elmundo.es/elmundodeporte/rss/portada.x...
	37	elmundo	internacional	http://www.elmundo.feedsportal.com/elmundo/rss/int...
	36	elmundo	nacional	http://www.elmundo.feedsportal.com/elmundo/rss/esp...
	35	cope	cultura	http://cope.es/rss.8.false
	34	cope	economia	http://cope.es/rss.4.false
	33	cope	deportes	http://cope.es/rss.6.false
	32	cope	internacional	http://cope.es/rss.2.false
	31	cope	nacional	http://cope.es/rss.1.false
	30	elpais	cultura	http://elpais.com/rss/feed.html?feedId=1008
	29	elpais	economia	http://elpais.com/rss/feed.html?feedId=1006
	28	elpais	internacional	http://elpais.com/rss/feed.html?feedId=1001
	27	elpais	deportes	http://elpais.com/rss/feed.html?feedId=1007
	26	elpais	nacional	http://elpais.com/rss/feed.html?feedId=1002
	45	abc	cultura	http://abc.es/rss/feeds/abc_CulturaCultura.xml
	46	ser	nacional	http://cadenaser.com/rss/feed.html?feedId=101
	47	ser	internacional	http://cadenaser.com/rss/feed.html?feedId=102
	48	ser	deportes	http://cadenaser.com/rss/feed.html?feedId=103
	49	ser	economia	http://cadenaser.com/rss/feed.html?feedId=107
	50	ser	cultura	http://cadenaser.com/rss/feed.html?feedId=108

Figura 18. Ejemplo de contenidos de la tabla de enlaces

La siguiente tabla en esta base de datos es la tabla de historial. En ella se almacena un pequeño historial sobre las llamadas que se han realizado a la aplicación, incluyendo la fuente y la categoría de noticias que se han consultado por cada número de teléfono con el que se ha interactuado con a la aplicación. Esta tabla se utiliza para posibilitar el reconocimiento de usuarios frecuentes y adaptación de la aplicación teniendo en cuenta sus preferencias y consultas previamente realizadas, dándoles a elegir antes de preguntarles por la fuente, sus fuentes y categorías de noticias favoritas, basándose en sus consultas recientes. Los campos que componen esta tabla son los siguientes:

- **Llamada:** Se trata de un campo numérico, con la finalidad de dar un id a la entrada, es único y por lo tanto se usa como clave primaria.
- **Caller:** Se trata del número de donde ha provenido la llamada, es de tipo texto, debido a que ciertos identificadores de llamada pueden contener caracteres alfanuméricos.
- **Fuente:** Se trata de un campo de tipo texto, en el que se guarda la fuente de noticias que ha consultado esa llamada.
- **Sección:** Se trata de un campo de tipo texto, en el que se guarda la sección de noticias que ha consultado esa llamada.

La Figura 19 muestra un ejemplo de este tipo de tabla.

←T→	llamada	caller	fuente	seccion
	6	3853f340-15f1-400d-bbdb-dded93c923d7	abc	deportes
	5	3853f340-15f1-400d-bbdb-dded93c923d7	elpais	nacional
	7	3853f340-15f1-400d-bbdb-dded93c923d7	elpais	nacional
	8	3853f340-15f1-400d-bbdb-dded93c923d7	abc	deportes
	9	3853f340-15f1-400d-bbdb-dded93c923d7	elpais	deportes
	10	0062e050-6d88-44d1-8786-118928b71cea	cope	internacional
	11	0062e050-6d88-44d1-8786-118928b71cea	elpais	deportes
	12	0062e050-6d88-44d1-8786-118928b71cea	elpais	nacional
	13	0062e050-6d88-44d1-8786-118928b71cea	elpais	deportes
	14	0062e050-6d88-44d1-8786-118928b71cea	elpais	deportes
	15	0062e050-6d88-44d1-8786-118928b71cea	elpais	nacional
	16	0062e050-6d88-44d1-8786-118928b71cea	elpais	deportes
	17	0062e050-6d88-44d1-8786-118928b71cea	cope	nacional
	18	+34[redacted]	abc	deportes
	19	+34[redacted]	elpais	deportes

Figura 19. Ejemplo de contenidos de la tabla de historial

Como vemos en la figura se han ocultado algunos números para preservar la confidencialidad del llamante. En el caso de que la aplicación fuera comercializada en el futuro, se debería informar al usuario de que su número de teléfono va a ser almacenado en una base de datos, teniendo así en cuenta los requerimientos de Ley Orgánica de Protección de Datos (www.boe.es/boe/dias/1999/12/14/pdfs/A43088-43099.pdf).

Capítulo 4

Descripción detallada de los módulos del sistema

En el presente capítulo se describe con detalle cada uno de los módulos que componen la aplicación desarrollada. La aplicación está compuesta por un total de cinco módulos que comprenden diferentes páginas PHP, dos páginas más auxiliares, con la misma extensión en las que almacenan diferentes funciones, y tres archivos de gramáticas, además de la base de datos descrita en el capítulo anterior.

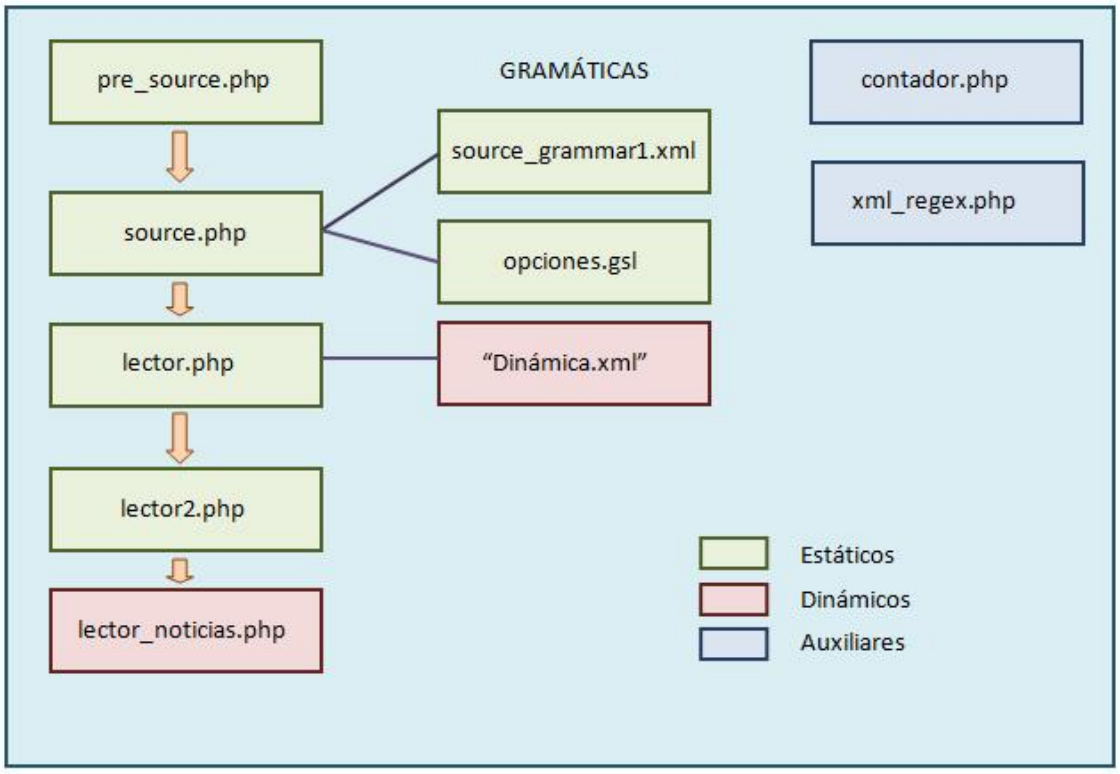


Figura 20. Esquema de Los diferentes módulos que componen la aplicación

A continuación se describe detalladamente cada uno de componentes de la aplicación.

4.1 Módulo 1: Identificación de usuario

La composición de este módulo es muy sencilla, ya que su función principal es conocer el número de donde procede la llamada y transmitirlo a la siguiente página, que se encarga de almacenarlo en una variable de PHP (Figura 21).

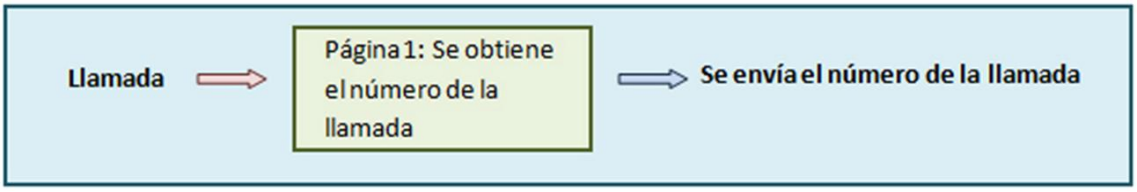


Figura 21. Esquema del módulo 1

La utilización de esta página es debido a que cuando disponemos de un dato almacenado en una variable de VoiceXML y necesitamos guardarlo en una variable de PHP para, por ejemplo, almacenarlo seguidamente en la base de datos, el método recomendado por la plataforma Voxeo consiste en remitirlo a otra página mediante el método <*submit*> y seguidamente almacenarlo en esta página utilizando una variable de PHP, tal y como muestra la Figura 22.

```
<submit next="source.php"  method="get" namelist="preid"/>

$id = $_REQUEST["preid"];
```

Figura 22. Ejemplo de transmisión de variables en PHP y VoiceXML

Debido a que sólo existe este medio de pasar variables desde VoiceXML a PHP, éste ha sido una de las motivaciones por las que la aplicación se ha dividido en diferentes páginas. Como veremos más adelante, este método de tratamiento de variables se repite más veces durante la ejecución de la aplicación.

4.2 Módulo 2: Selección de la fuente y categoría de noticias

Se trata de una de los módulos más importantes de la aplicación, en él se requiere al usuario sobre la fuente de donde obtener las noticias y la categoría que desea consultar, con la excepción de que si el usuario ya ha utilizado anteriormente la aplicación, se le ofrece adicionalmente la posibilidad de acceder directamente a la sección anotada por el sistema como favorita (Figura 23).

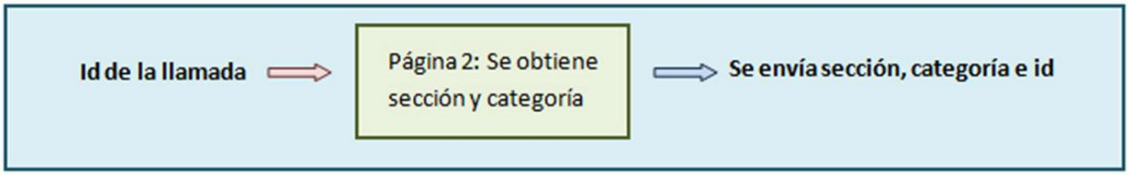


Figura 23. Esquema de funcionamiento del módulo 2

De este modo, la primera función que realiza este módulo es verificar si el usuario ha accedido recientemente a la aplicación, para lo que utilizamos el número que hemos obtenido en la página anterior y consultamos la tabla "LOG" de la base de datos. En caso de que así sea, comprobamos si el usuario ha accedido a una misma fuente y sección más de dos veces.

Si estas comprobaciones resultan positivas, deducimos que el usuario ya ha accedido varias veces a la aplicación y que posee además una sección favorita, que será la que la aplicación le sugiera antes de preguntarle por una en concreto. Si se diera el caso de que aparecieran dos secciones con el mismo número de visitas, se seleccionaría aquella que se ha consultado más recientemente. Tanto en el caso de que el usuario seleccione la sección detectada como favorita como si se transfiere al usuario a la página de elección de fuente, se almacenan los datos de identificación de la llamada, fuente y sección seleccionada.

Si por el contrario se diera el caso de que fuera la primera vez que el usuario utiliza la aplicación, directamente se transfiere al menú de elección de fuente, seguidamente al del elección de sección, y por último se envían los datos de fuente, sección e identificador de la llamada a la siguiente página.

La estructura de los menús de elección de fuente y sección consta de los dos campos correspondientes en un formulario VoiceXML, cada uno de ellos con una gramática asociada. En dichas gramáticas se recogen las posibles opciones que serán reconocidas por el reconocedor automático del habla, en este caso, las diferentes fuentes de noticias y secciones definidas en la aplicación. La Figura 24 resume el funcionamiento de este módulo.

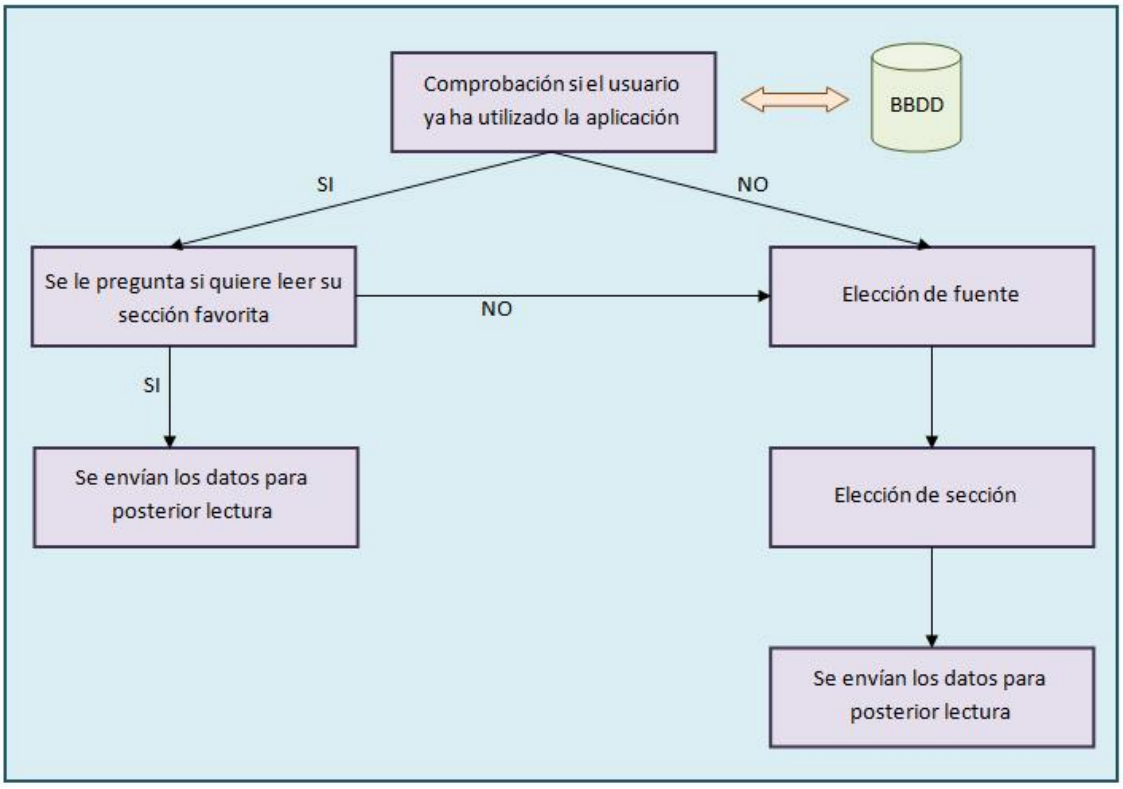


Figura 24. Diagrama de funcionamiento del módulo 2

4.3 Módulo 3: Extracción de noticias y temas populares

Durante la ejecución de este módulo se extraen las noticias de la fuente y categoría seleccionadas, para guardarlas seguidamente en la base de datos. Además, se realiza la extracción de los temas populares de esa sección, utilizando un algoritmo que se detallará a continuación. Una vez obtenidos estos temas, se genera dinámicamente la gramática correspondiente y se requiere al usuario el tema que desea escuchar (Figura 25). Con esta información se envían los datos de: identificador de llamada, fuente, categoría y tema a la siguiente página para la posterior creación de la página de lectura.

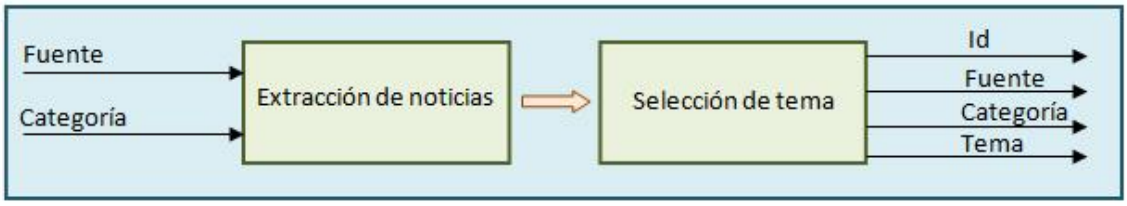


Figura 25. Funcionamiento del módulo 3

La ejecución de este módulo comienza con la captura de las variables obtenidas en el módulo anterior, fuente y categoría, que se utilizan para la posterior extracción de noticias del *feed* correspondiente. Para obtener la URL del *feed* se realiza una llamada a la función *obten_url()*, tomando como parámetros las variables fuente y categoría, con lo que la función simplemente se conecta a la base de datos y solicita de la tabla "URL", la dirección correspondiente a la fuente y sección solicitada.

Una vez que se conoce la URL, se realiza una llamada a la función *extraer()*. Esta función es la más compleja de la aplicación, dado que se encarga de extraer las noticias y almacenarlas en la base de datos, así como de obtener dinámicamente los temas populares de las noticias extraídas. Esta función se explica de manera ampliada al final de este apartado.

Después de finalizar el proceso de extracción y procesamiento de las noticias, se solicita sobre qué tema desea escuchar las noticias, teniendo en cuenta que siempre está disponible la opción del tema "general" que contendrá todas las noticias de la categoría, así como la posibilidad de seleccionar uno de los temas que contendrá las noticias relacionadas con ese tema. Para obtener la elección del usuario sobre los temas es necesaria la creación, dinámicamente, de la gramática correspondiente. Para ello, se utiliza la función *crea_gramatica()* que se encarga, utilizando como entrada el vector de palabras de la función *extraer()*, de generar la gramática de los temas populares. A cada archivo de gramáticas se le asigna el nombre correspondiente a la fuente y la categoría de las mismas, con el fin de ser fácilmente identificables, por ejemplo, "elpaisnacional.xml".

Nada más se conoce la elección del usuario sobre el tema de noticias que desea escuchar, se envía esta información a la siguiente página junto con el identificador de la llamada, la fuente y la categoría de las noticias. La Figura 26 resume el funcionamiento de este módulo.

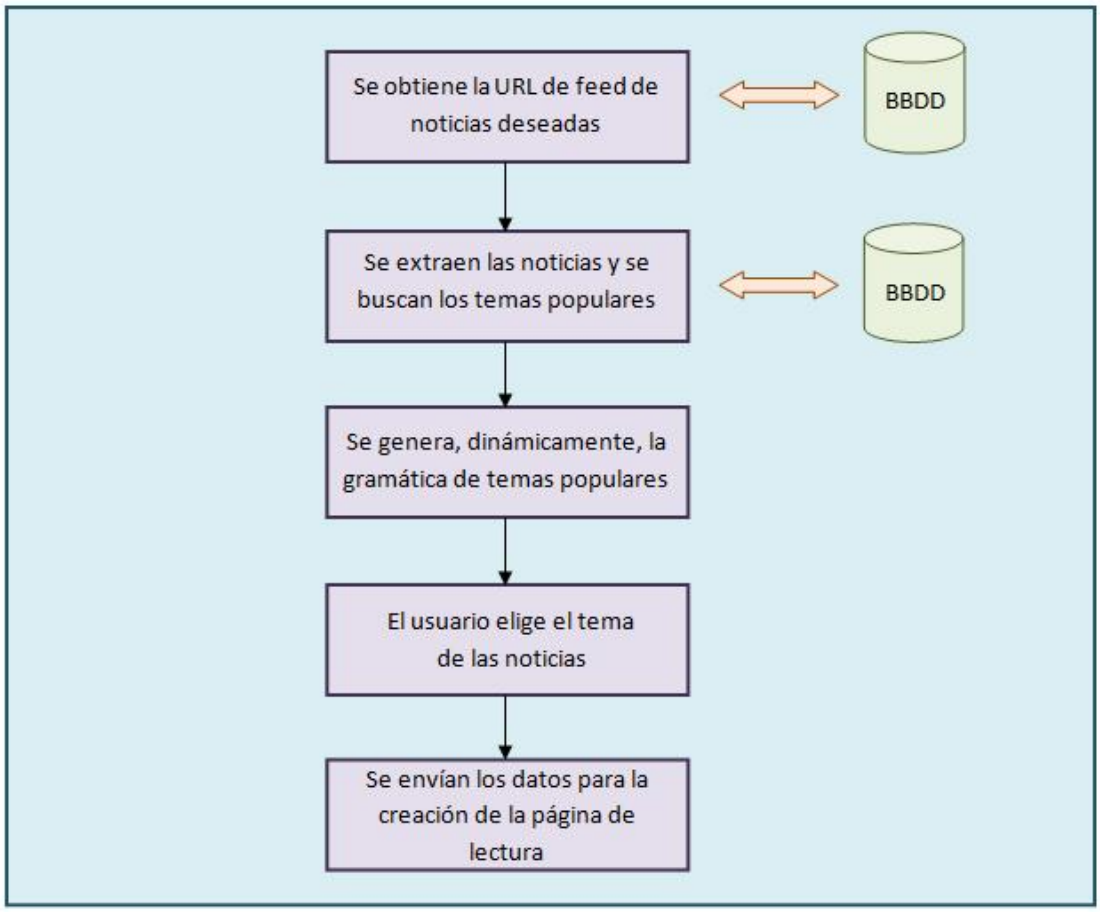


Figura 26. Diagrama de funcionamiento del módulo 3

4.3.1 Función Extraer()

Esta función realiza la conexión a la base de datos y crea la tabla que almacenará todas las noticias de la categoría seleccionada. Las noticias, tal y como se detalló al describir la base de datos en el capítulo 3, se componen de un título y una descripción. A continuación, se van extrayendo los títulos y las descripciones de las diferentes noticias, filtrando caracteres que pudieran provocar un error al realizar la lectura.

Una vez que ya tenemos las noticias, con su título y descripción introducidos en la base de datos, se realiza la obtención de los temas populares de las mismas. Para ello, se consultan los títulos de las noticias guardadas en la base de datos, y se sigue el siguiente algoritmo:

i) Se buscan las palabras que comienzan con mayúsculas.

ii) De este grupo se buscan las que se repiten más de dos veces, contando todos los títulos.

iii) De este último grupo se eliminan las palabras de longitud menor de tres caracteres.

Con este pequeño algoritmo se consigue que los temas populares sean nombres propios, ya sean personas o instituciones, que estos sean populares ya que se repiten más de dos veces en diferentes noticias de la misma sección, y que al tener una longitud mayor a tres caracteres, queden fuera conjunciones y preposiciones.

Una vez obtenidos estos temas populares, se almacenan en un vector generado como salida de la función, y se crea una tabla en la base de datos por cada uno de los temas. En estas tablas se introducirán las noticias referidas a dichos temas. De este modo, dispondremos de una tabla que contendrá todas las noticias de la categoría y una tabla por cada tema popular, que contendrá las diferentes noticias referidas a los distintos temas.

4.4 Módulo 4: Historial del usuario

Durante la ejecución de este módulo se almacena el historial del usuario que ha efectuado la llamada y, además, se genera dinámicamente la siguiente página, conocida como "lector_noticias.php", que será la utilizada para la lectura de las noticias de la base de datos (Figura 27).

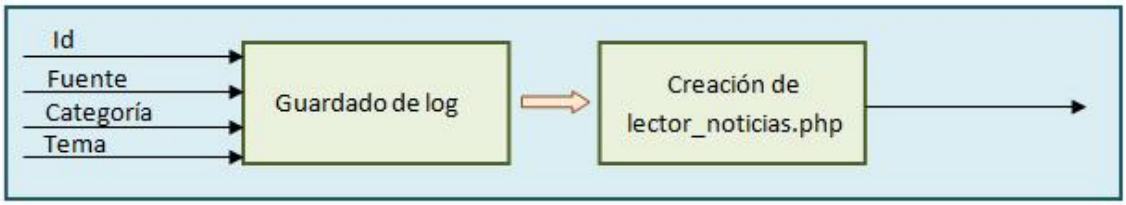


Figura 27. Esquema del módulo 4

Esta página recibe las variables enviadas por la página anterior (identificador, fuente, categoría y tema). Estas variables se utilizan para el almacenamiento del historial y para la generación de la página de lectura.

A continuación, se realiza una llamada a la función *guarda_log()*, que tal y como su nombre indica, se encarga de almacenar el historial de la llamada. Para ello, la función se conecta a la base de datos, y en concreto a la tabla de "log", para almacenar el identificador de la llamada, la fuente y sección que ha consultado, para su posterior uso cuando el mismo usuario realice una nueva llamada al sistema.

Antes de generar el archivo de lectura, nos aseguramos de que no existe ninguno anterior en el servidor, utilizando para ello la función *borrar_lector()*, que detecta si existe algún archivo anterior para así proceder a eliminarlo. Posteriormente, la aplicación se conecta a la base de datos de nuevo para obtener el número de noticias que existen del tema seleccionado, utilizando este número en la función *generador_lector()* para saber el número de noticias que deben leerse.

Seguidamente se realiza una llamada a la función *borrar_lector()*, que se encargará de borrar la página de lectura de noticias si existiera (Figura 28).

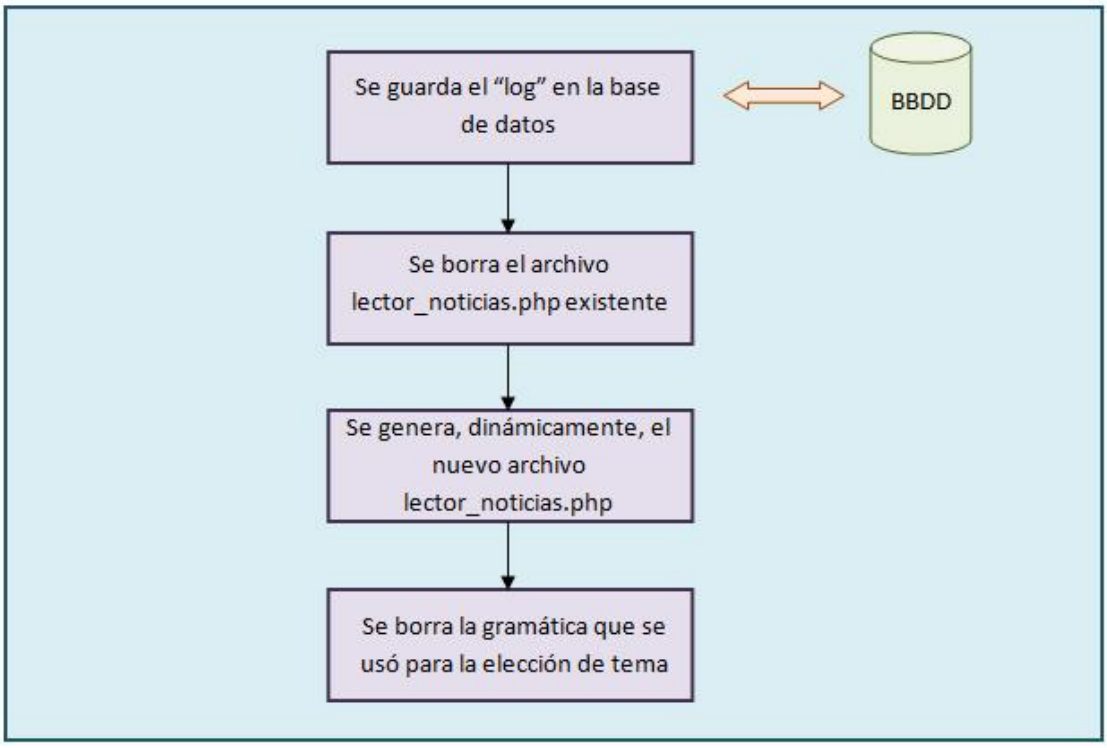


Figura 28. Diagrama de funcionamiento del módulo 4

A continuación se procede a la creación de la página de lectura de noticias. Para ello, se llama a la función *generador_lector()* con los parámetros de número de noticias, fuente y categoría. Está función creará una página nueva llamada "lector_noticias.php", que contendrá el código necesario para la lectura de las noticias y que describiremos con un ejemplo en el siguiente apartado.

Después de crear esta página, la aplicación realiza una llamada a la función *borrar_gramatica()*. Dado que ya se ha dado por terminado el proceso de selección de tema por parte del usuario, podemos eliminar esta gramática y así eliminar el problema de que se dupliquen gramáticas o se sobrescriban. Por último, se redirige la aplicación a la página recientemente creada, lector_noticias.php.

4.5 Módulo 5: Lectura de las noticias

La creación de la página "lector_noticias.php" es dinámica, por lo que en cada ejecución de la aplicación esta página será distinta. Para su explicación tomaremos un ejemplo obtenido de una de las ejecuciones de prueba, a partir del cual se describe la estructura de la página. En una primera parte se definen las acciones a realizar con los controles avanzados de reproducción: siguiente, anterior, repetir, ampliar y volver. En la segunda parte se incluyen los formularios necesarios para la reproducción de las noticias.

Para poder desarrollar la primera parte de la página se utiliza la etiqueta *<link>,* como vemos en la Figura 29. Gracias a esta etiqueta podemos definir las palabras que activarán cada una de las opciones, si el usuario así las pronuncia en cualquier momento durante la ejecución de la página, y la acción que queremos que se lleve a cabo para cada una de ellas, en nuestro caso, la etiqueta *<next>* para ir al formulario correspondiente.

```
9  <link next="source.php"><grammar type="text/gsl">[volver]</grammar></link>
10 <link next="#siguiente"><grammar type="text/gsl">[siguiente]</grammar></link>
11 <link next="#anterior"><grammar type="text/gsl">[anterior]</grammar></link>
12 <link next="#repetir"><grammar type="text/gsl">[repetir]</grammar></link>
13 <link next="#ampliar"><grammar type="text/gsl">[ampliar]</grammar></link>
```

Figura 29. Ejemplo de etiquetas <*link*>

Como veremos posteriormente, cada noticia de las existentes en la base de datos se lee en un formulario diferente. Estos formularios están numerados para

poder generarlos dinámicamente, ya que inicialmente no sabemos cuántas noticias van a existir sobre el tema seleccionado. Además, se utiliza la variable ”*cont*” de VoixeXML como contador, para poder deducir en qué formulario se encuentra actualmente la aplicación. Por este motivo, en las opciones siguiente, anterior y repetir, se utilizan formularios con sentencias de *<if>* anidados y *<goto>* en su interior, para utilizar el valor de la variable “*cont*” para acceder al formulario que sea necesario.

Como vemos en la Figura 30, en el ejemplo sólo existen dos noticias en la base de datos sobre el tema seleccionado, por lo que se generarán dos formularios para la lectura de cada una de éstas noticias.

```
15 <form id="repetir"><block>
16 <if cond="cont == 1"><goto next="#1"/>
17  <elseif cond="cont == 2"/><goto next="#2"/>
18 </if></block></form>
19 <form id="siguiente"><block><assign name="document.cont" expr="sumar(cont)"/>
20 <if cond="cont == 1"><goto next="#1"/>
21  <elseif cond="cont == 2"/><goto next="#2"/>
22 </if></block></form>
23 <form id="anterior"><block><assign name="document.cont" expr="restar(cont)"/>
24 <if cond="cont == 1"><goto next="#1"/>
25  <elseif cond="cont == 2"/><goto next="#2"/>
26 </if></block></form>
```

Figura 30. Ejemplo de comandos de reproducción

Para definir la opción “ampliar”, se ha diseñado un formulario más complejo. En él, debemos de establecer una conexión con la base de datos para poder extraer la parte de “descripción” de las noticias, e introducirlas en un vector para su posterior lectura. En función de la noticia que se estuviera leyendo en el momento en el que el usuario pronunció el comando ampliar, se leerá la descripción correspondiente. Para esta selección se utiliza la variable “*cont*” que comentamos anteriormente, que en nuestro ejemplo, dará como resultado la elección de la noticia uno o dos (Figura 31).

```
<form id="ampliar">
<field name="dummy0"><property name="timeout" value="1s"/><grammar type="text/gsl">[poppaoomowmow]</grammar>
 <filled><prompt>no way this will ever happen.</prompt></filled>
<noinput>
 <goto nextitem="announce0"/>
 </noinput></field>
<?$username="";$password="";$database="victorgb_1";mysql_connect("localhost",$username,$password);
 @mysql_select_db($database) or die ("No se pudo conectar con la base de datos");
 $i=1;$k=0;
while ($i<=2){
 $query = "SELECT * FROM zapatero";
 $result=mysql_query($query);
 $description[$i]=mysql_result($result,$k,"description");$i++;$k++;}?>
<block name="announce0">
<if cond="cont == 1"><prompt bargein="true"><?echo $description[1];mysql_close();?></prompt><goto next="#2"/>
 <elseif cond="cont == 2"/><prompt bargein="true"><?echo $description[2];mysql_close();?></prompt><goto next="#3"/>
 </if>
 </block>
 </form>
```

Figura 31. Ejemplo del comando ampliar información

Podemos apreciar como en el ejemplo del comando “ampliar”, introducimos un campo con el nombre de “*dummy0*”. Este campo no va a tener un uso específico en la aplicación, pero es necesario para que se puedan ejecutar la lectura de las noticias. Por ello, le asignamos una gramática irreconocible para que siempre se active el evento “*noinput*”. Esta práctica la repetiremos para los formularios de lectura de noticias normales, como se describe más adelante

Una vez que se ha leído la ampliación o descripción de la noticia deseada, se seguirá a continuación con la siguiente noticia que se estaba leyendo en el momento de seleccionar estas opciones.

Para la lectura de las noticias normales generaremos formularios independientes para cada una de ellas. Se ha elegido esta técnica porque resultó ser la más sencilla para posibilitar el paso entre cada una de las noticias durante su lectura.

Como vemos en la Figura 32, los formularios consisten en una conexión con la base de datos para la extracción de la noticia y su posterior lectura. Una vez leída continuamos con el siguiente formulario y la siguiente noticia.

```
<form id="1">
<field name="dummy1"><property name="timeout" value="1s"/><grammar type="text/gsl">[poppaoomowmow]</grammar>
 <filled><prompt>no way this will ever happen.</prompt></filled>
<noinput>
 <goto nextitem="announce1"/>
</noinput></field>
<?$username="";$password="";$database="victorgb_1";
 mysql_connect("localhost",$username,$password);@mysql_select_db($database) or die ("No se pudo conectar con la base de datos");
 $query = "SELECT * FROM zapatero";
 $result=mysql_query($query);
 $title=mysql_result($result,0,"title");
 echo '<block name="announce1">';
 echo "<assign name='document.cont' expr='sumar(cont)'/>";
 echo '<prompt bargein="true">';
 echo $title;
 echo "</prompt>";
 echo '<goto next="#2"/>';
 echo "</block>";
mysql_close();?>
</form>
<form id="2">
<field name="dummy2"><property name="timeout" value="1s"/><grammar type="text/gsl">[poppaoomowmow]</grammar>
 <filled><prompt>no way this will ever happen.</prompt></filled>
<noinput>
 <goto nextitem="announce2"/>
</noinput></field>
<?$username="";$password="";$database="victorgb_1";
 mysql_connect("localhost",$username,$password);@mysql_select_db($database) or die ("No se pudo conectar con la base de datos");
 $query = "SELECT * FROM zapatero";
 $result=mysql_query($query);
 $title=mysql_result($result,1,"title");
 echo '<block name="announce2">';
 echo "<assign name='document.cont' expr='sumar(cont)'/>";
 echo '<prompt bargein="true">';
 echo $title;
 echo "</prompt>";
 echo '<goto next="#fin"/>';
 echo "</block>";
mysql_close();?>
</form>
```

Figura 32. Ejemplo de formularios para la lectura de noticias

4.6 Páginas contador.php y xml_regex.php

La principal función de estas páginas es la de contener funciones que posteriormente serán consultadas en el resto de páginas durante la ejecución de la aplicación.

Las funciones contenidas en “contador.php” son las que se han comentado anteriormente durante la explicación de las otras páginas, en concreto:

- Extraer()
- Crea_gramática()
- Generador_lector()
- Borrar_gramática()
- Borrar_db()

- Borrar_lector()
- Guarda_log()
- Obten_url()

Las funciones contenidas en la página "xml_regex.php" fueron obtenidas de Internet (http://www.bobulous.org.uk/coding/php-xml-feeds.html) y realizadas por el autor "Bobulous". Se utilizan para la extracción de información de documentos XML, en nuestro de caso de los *feeds* RSS. Sólo se utilizan las funciones *value_in*() y *element_set*(), aunque en total se incluyen las siguientes:

- ***Value_in*():** Devuelve el valor de los elementos cuyos nombre coinciden con el parámetro introducido y en el *feed* buscado.
- ***Element_set*():** Devuelve un vector de elementos cuyos nombres coinciden con el parámetro introducido y en el *feed* buscado.
- ***Element_attributes*():** Devuelve los atributos del primer elemento que coincide con el parámetro introducido y en el *feed* buscado.
- ***Make_safe*():** Elimina algunos caracteres conflictivos de la cadena introducida.

4.7 Evaluación de la aplicación desarrollada

En esta sección se describe la evaluación preliminar que se ha llevado a cabo de la aplicación. Esta evaluación ha consistido en la elaboración de un cuestionario de 11 preguntas para medir la valoración subjetiva de los usuarios tras interactuar con la aplicación, así como su experiencia previa y manejo de este tipo de tecnologías. A continuación mostramos la selección de preguntas incluidas en el test:

1. De 1 al 5, ¿Cómo calificaría sus conocimientos previos sobre las tecnologías utilizadas en la aplicación?
2. De 1 al 5, ¿Cómo calificaría su experiencia previa en el uso de interfaces por voz y tecnologías similares?
3. ¿Cuántas veces ha utilizado interfaces por voz previamente? Nunca, muy pocas, algunas veces, de vez en cuando, muchas veces.
4. ¿Qué tal entendió los mensajes generados por el sistema? Muy mal, mal, regular, bien, muy bien.

5. En su opinión, la interacción con el sistema fue… muy lenta, lenta, adecuada, rápida, muy rápida.
6. Califique la dificultad de utilización del sistema. Muy fácil, fácil, regular, difícil, muy difícil.
7. ¿Le fue fácil obtener la información que solicitaba al sistema? No, fue imposible; sí, pero con gran dificultad; sí, pero con algún dificultad; sí, fue fácil; sí, fue muy fácil.
8. ¿Quedó satisfecho con el uso de la aplicación? Absolutamente insatisfecho, no muy satisfecho, indiferente, satisfecho, muy satisfecho.
9. ¿Sabía lo que tenía que responder al sistema en cada momento? No, nunca; no, casi nunca; a veces; sí, casi siempre; sí, siempre.
10. ¿Cree que el comportamiento del sistema es parecido al que podría desempeñar un ser humano? no, nunca; no, casi nunca; n veces; sí, casi siempre; sí, siempre.
11. Valore globalmente la aplicación. No me ha gustado nada, muy poco, más o menos, buena, realmente muy buena.

Adicionalmente se ha evaluado un conjunto de parámetros estadísticos que incluye el número de interacciones con éxito, la media de turnos de diálogo, el número medio de errores por diálogo, la duración media de las llamadas y el número medio de repeticiones del turno del sistema (*reprompts*). Un total de 25 usuarios participó en esta evaluación preliminar, llevando a cabo cada uno de ellos una interacción con el sistema sin la indicación de escenarios predefinidos.

4.7.1 Resultados globales de la evaluación

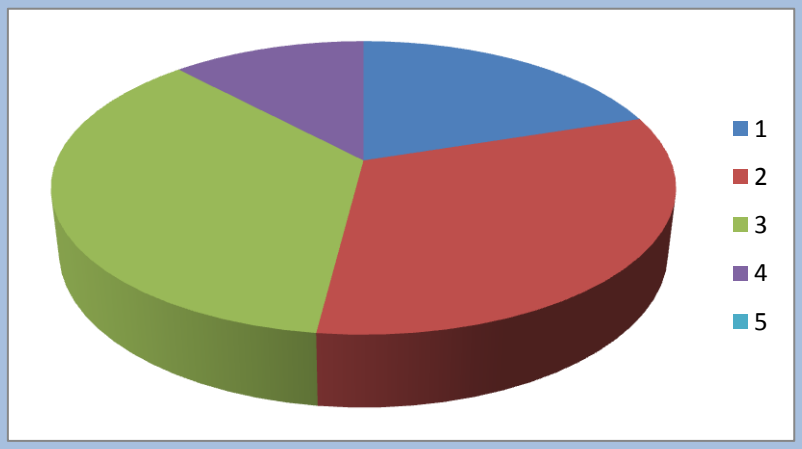
La Tabla 4 muestra los resultados globales de los parámetros estadísticos definidos para le evaluación del sistema. Como podemos apreciar si nos fijamos en los resultados, la tasa de iteraciones con éxito es elevada, produciéndose errores tan sólo en una de ellas. Además, podemos observar que la tasa de repetición de frases no es excesivamente elevada, por lo que podemos deducir que los usuarios entendieron bien las locuciones del sistema y el sistema entendió bien las órdenes de los usuarios.

RESULTADOS GLOBALES	
Número de usuarios	25
Número de interacciones	25
Número de interacciones con éxito	24
Número medio de errores por diálogo	0.04
Media de turnos de diálogo	16
Duración media de la llamada	6 minutos y 24 segundos
Número medio de repetición de frases (*Reprompts*)	1.6

Tabla 4. Resultados globales de la medidas estadísticas

A continuación mostramos las estadísticas globales obtenidas del cuestionario de evaluación subjetiva.

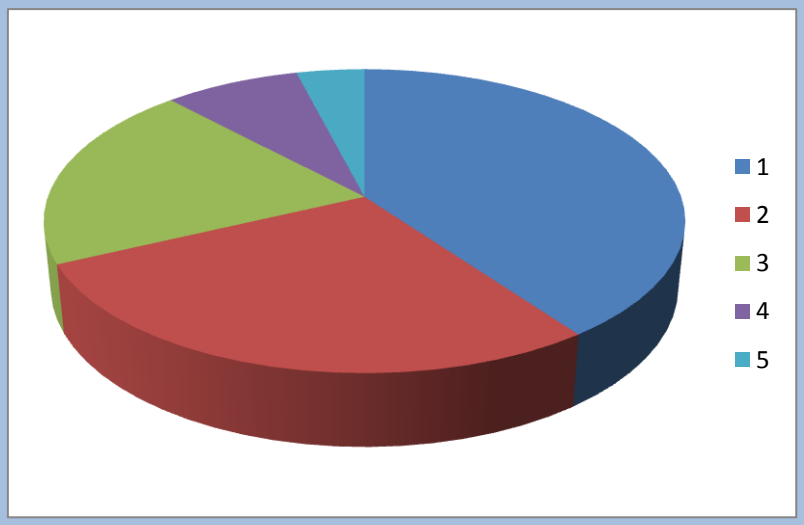
Pregunta 1: De 1 al 5, ¿Cómo calificaría sus conocimientos previos sobre las tecnologías utilizadas en la aplicación?



Valor máximo	Valor mínimo	Valor medio	Desviación típica
4	1	2,40	0,94

Tabla 5. Resultados globales de la Pregunta 1

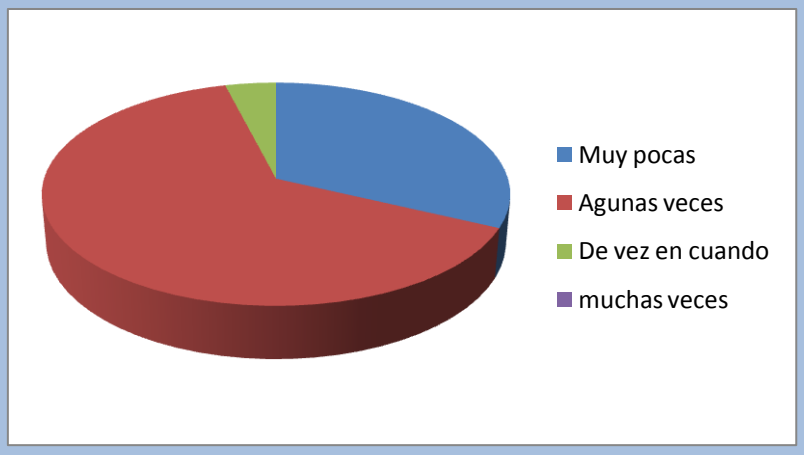
Pregunta 2: De 1 al 5, ¿Cómo calificaría su experiencia previa en el uso de interfaces por voz y similares?



Valor máximo	Valor mínimo	Valor medio	Desviación típica
5	1	2,08	1,13

Tabla 6. Resultados globales de la Pregunta 2

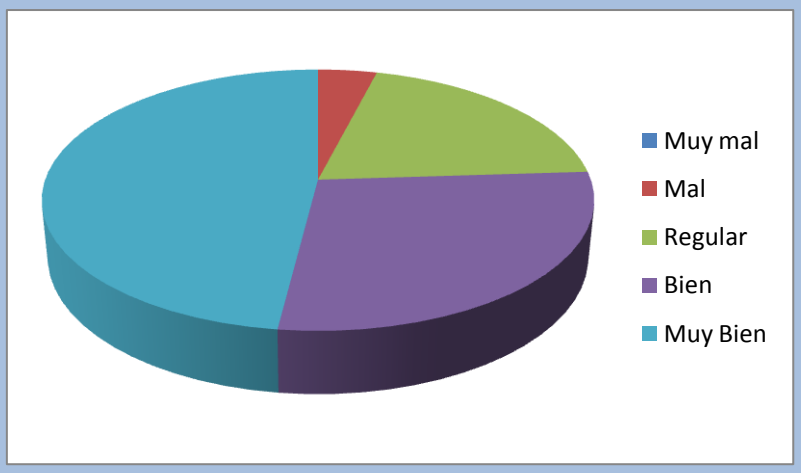
Pregunta 3: ¿Cuántas veces ha utilizado interfaces por voz previamente?



Valor máximo	Valor mínimo	Valor medio	Desviación típica
3	2	2,72	0,53

Tabla 7. Resultados globales de la Pregunta 3

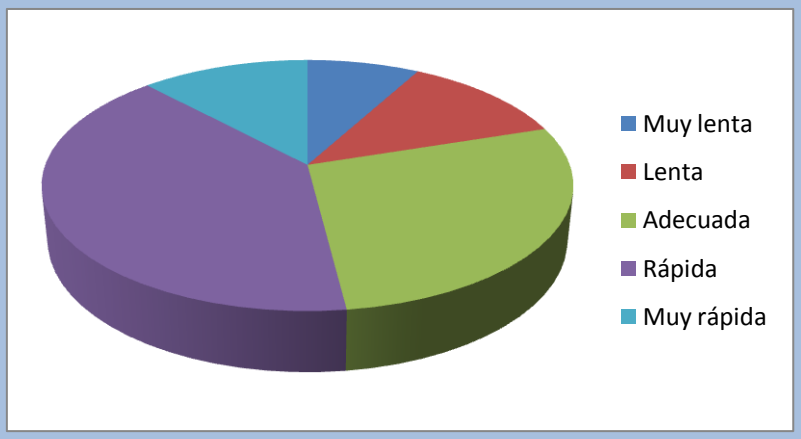
Pregunta 4: ¿Qué tal entendió los mensajes generados por el sistema?



Valor máximo	Valor mínimo	Valor medio	Desviación típica
5	2	4,20	0,89

Tabla 8. Resultados globales de la Pregunta 4

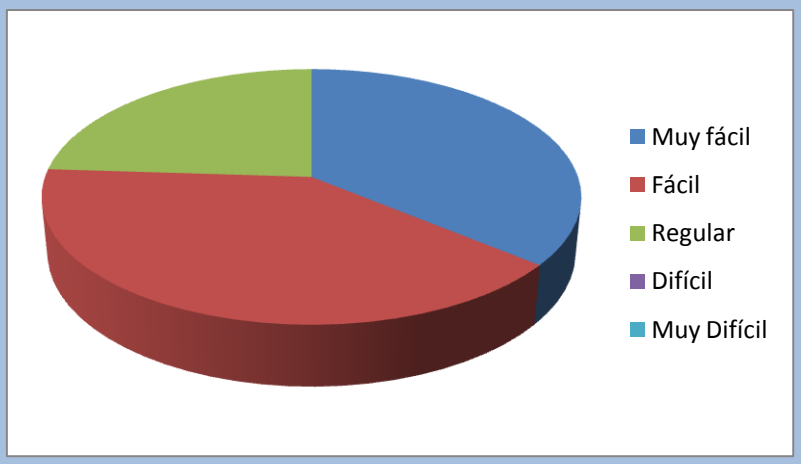
Pregunta 5: En su opinión, la interacción con el sistema fue… muy lenta, lenta, adecuada, rápida, muy rápida.



Valor máximo	Valor mínimo	Valor medio	Desviación típica
5	1	3,36	1,09

Tabla 9. Resultados globales de la Pregunta 5

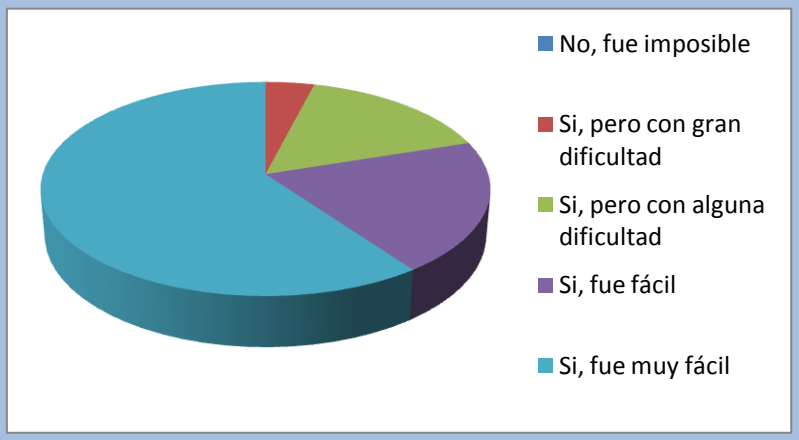
Pregunta 6: Califique la dificultad de utilización del sistema.



Valor máximo	Valor mínimo	Valor medio	Desviación típica
3	1	1,88	0,77

Tabla 10. Resultados globales de la Pregunta 6

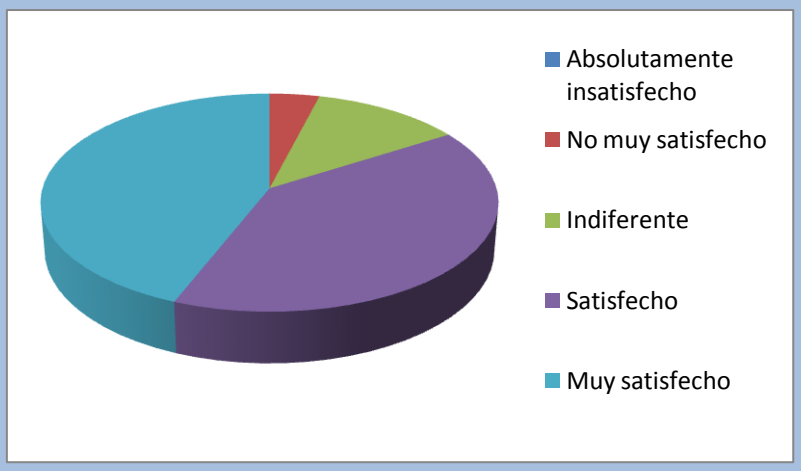
Pregunta 7: ¿Le fue fácil obtener la información que solicitaba al sistema?



Valor máximo	Valor mínimo	Valor medio	Desviación típica
5	2	4,36	0,89

Tabla 11. Resultados globales de la Pregunta 7

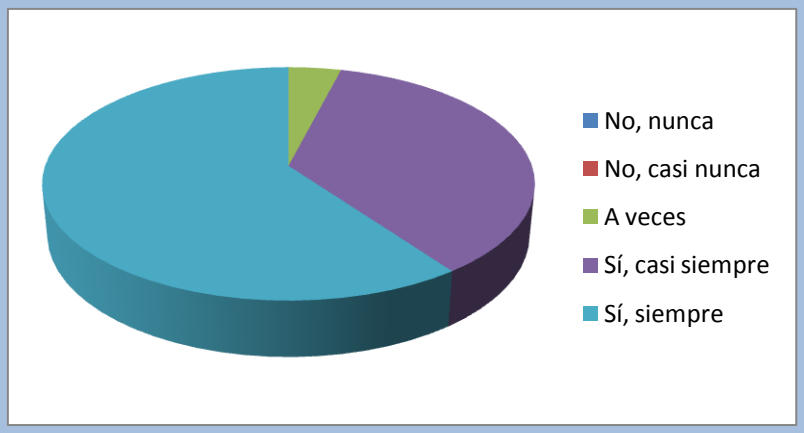
Pregunta 8: ¿Quedó satisfecho con el uso de la aplicación?



Valor máximo	Valor mínimo	Valor medio	Desviación típica
5	2	4,24	0,81

Tabla 12. Resultados globales de la Pregunta 8

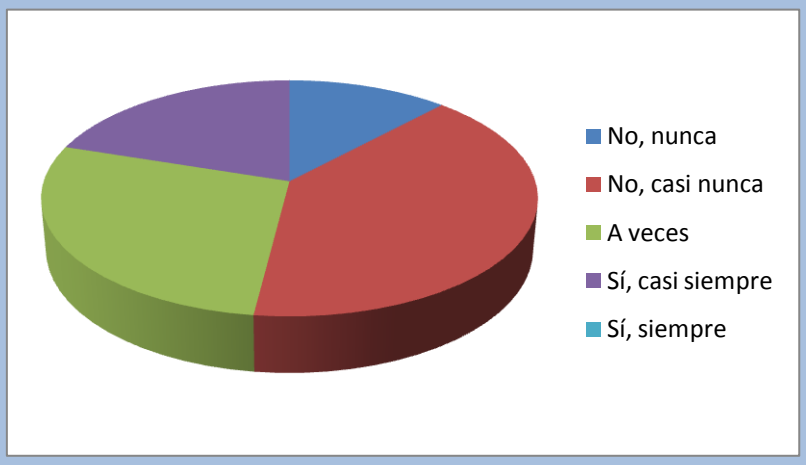
Pregunta 9: ¿Sabía lo que tenía que responder al sistema en todo momento?



Valor máximo	Valor mínimo	Valor medio	Desviación típica
5	3	4,56	0,57

Tabla 13. Resultados globales de la Pregunta 9

Pregunta 10: 1. ¿Cree que el comportamiento del sistema es parecido al que podría desempeñar un ser humano?



Valor máximo	Valor mínimo	Valor medio	Desviación típica
4	1	2,56	1,64

Tabla 14. Resultados globales de la Pregunta 10

Pregunta 11: Valore globalmente la aplicación.

Valor máximo	Valor mínimo	Valor medio	Desviación típica
5	3	4,20	0,75

Tabla 15. Resultados globales de la Pregunta 11

4.7.2 Prueba de evaluación 1

En la prueba número 1 se muestran los resultados de una prueba realizada por un usuario de 59 años, con conocimientos informáticos limitados, y que no ha utilizado la aplicación previamente. El resultado fue satisfactorio según el usuario. Se consultaron las siguientes secciones:

- El País > Nacional > Madrid (Tema)
- El Mundo > Internacional > General

Los pasos seguidos una vez respondida la llamada por el sistema fueron:

1. Nueva llamada.
2. Elección de fuente de noticias, se eligió el periódico “El País”.
3. Elección de la sección, se eligió “Nacional”.
4. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Batasuna, Madrid. Se eligió “Madrid”.
5. Una vez terminada la escucha de las noticias se volvió al menú principal para la elección de otro periódico.
6. Elección de fuente de noticias, se eligió el periódico “El Mundo”.
7. Elección de la sección, se eligió “Internacional”.
8. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Mubarak, Obama. Se eligió “General”.
9. Escucha de las noticias.
10. Fin de la llamada.

La aplicación funcionó perfectamente para las peticiones del usuario, simplemente se encontró cierta dificultad para descubrir la existencia de los controles avanzados. De esta prueba se puede deducir que se podría desarrollar como línea futura una nueva manera de que el usuario aprendiera los controles avanzados.

Estadísticas de la prueba:

- ¿Éxito?: Sí
- Nº de errores: 0
- Turnos de diálogo: 12 (sumando entre usuario y sistema)
- Duración de la llamada: 6 minutos y 35 segundos.
- Nº repetición de frases (Reprompts): 2

4.7.3 Prueba de evaluación 2

En la prueba número 2 se muestran los resultados del uso realizado por un usuario de 61 años, con conocimientos medios de Informática, y que no ha utilizado la aplicación previamente. El usuario consultó las noticias de uno de los periódicos en concreto su uso fue este:

- ABC > Nacional > General.
- El País > Economía > General.

Los pasos seguidos una vez respondida la llamada fueron los siguientes:

1. Nueva llamada.
2. Elección de fuente de noticias, se eligió el periódico “ABC”.
3. Elección de la sección, se eligió “Nacional”.
4. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Zapatero. Se eligió “General”.
5. Una vez terminada la escucha de las noticias se volvió al menú principal para la elección de otro periódico.
6. Elección de fuente de noticias, se eligió el periódico “El País”.
7. Elección de la sección, se eligió “Economía”.
8. Elección de unos de los temas populares de entre los siguientes que se generaron: General, España. Se eligió “General”.
9. Escucha de las noticias.
10. Se amplió una de las noticias con la descripción de la misma.
11. Escucha de las noticias restantes.
12. Fin de la llamada.

La interacción según palabras del usuario fue buena. Destacó que se podría mejorar la sensibilidad del reconocimiento de voz, posteriormente se hablará de ello en líneas futuras.

Estadísticas de la prueba:

- ¿Éxito?: Sí
- Nº de errores: 0
- Turnos de diálogo: 13 (sumando entre usuario y sistema)
- Duración de la llamada: 7 minutos y 06 segundos.
- Nº repetición de frases (Reprompts): 1

4.7.4 Prueba de evaluación 3

En la prueba número 3 se muestran los resultados del uso realizado por un usuario de 23 años, estudiante de Ingeniería, por lo que cuenta con amplios conocimientos de Informática. Se realizaron varias llamadas con el fin de probar la funcionalidad de "usuarios frecuentes". El resumen de las llamadas realizadas por el usuario fue el siguiente:

- Cadena Ser > Deportes > General.
- Cadena Ser > Deportes > Madrid.
- Cadena Cope > Deportes > General.

Los pasos seguidos por el usuario durante las pruebas fueron los siguientes:

1. Nueva llamada.
2. Elección de fuente de noticias, se eligió la radio "Cadena Ser"
3. Elección de la sección, se eligió "Deportes".
4. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Madrid, Messi. Se eligió "General".
5. Escucha de las noticias.
6. Terminó la llamada.
7. Nueva llamada.
8. Elección de fuente de noticias, se eligió la radio "Cadena Ser"

9. Elección de la sección, se eligió “Deportes”.
10. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Madrid, Messi. Se eligió “Madrid”.
11. Escucha de las noticias.
12. Terminó la llamada.
13. Nueva llamada.
14. Se le siguiere al usuario su sección favorita, en este caso Cadena Ser y Deportes. El usuario dice “No” y se pasa a elección de fuentes.
15. Elección de fuente de noticias, se eligió la radio “Cadena Cope”
16. Elección de la sección, se eligió “Deportes”.
17. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Villa. Se eligió “General”.
18. Escucha de las noticias.
19. Fin de la llamada.

Estadísticas de la prueba:

- ¿Éxito?: Sí
- Nº de errores: 0
- Turnos de diálogo: 20 (entre usuario y sistema)
- Duración de la llamada: 8 minutos y 35 segundos.
- Nº repetición de frases (Reprompts): 3

4.7.5 Prueba de evaluación 4

En la prueba de evaluación número 4 se muestran los resultados del uso de la aplicación realizado por un usuario de 22 años, con conocimientos a nivel medio-alto de Informática. Se pidió al usuario que tuviera especial atención en utilizar los controles de reproducción avanzados. El resumen de la llamada realiza por el usuario fue el siguiente:

- ABC > Cultura > Hollywood
- Cadena Ser > Internacional > General
- El Pais > Nacional > Rajoy

Los pasos seguidos una vez realizada la llamada fueron los siguientes:

1. Nueva llamada.
2. Se utilizó el comando “Help” para conocer los controles avanzados.
3. Elección de fuente de noticias, se eligió el periódico “ABC”.
4. Elección de la sección, se eligió “Cultura”.
5. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Hollywood. Se eligió “Hollywood”.
6. Escucha de las noticias.
7. Se utilizó “volver” para volver al menú principal, antes de que se terminará de leer todas las noticias.
8. Elección de fuente de noticias, se eligió el periódico “Cadena Ser”.
9. Elección de la sección, se eligió “Internacional”.
10. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Hollywood. Se eligió “General”.
11. Escucha de las noticias.
12. El usuario utiliza “siguiente” para pasar a la siguiente noticia rápidamente.
13. El usuario utiliza “ampliar” para ampliar una determinada noticia.
14. Elección de fuente de noticias, se eligió el periódico “El País”.
15. Elección de la sección, se eligió “Nacional”.
16. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Rajoy. Se eligió “Rajoy”.
17. Escucha de las noticias.
18. El usuario utiliza “repetir” para repetir la noticia que escuchaba en ese momento.
19. Fin de la llamada.

Estadísticas de la prueba:

- ¿Éxito?: Sí
- Nº de errores: 0
- Turnos de diálogo: 23 (entre usuario y sistema)
- Duración de la llamada: 10 minutos y 9 segundos.
- Nº repetición de frases (Reprompts): 2

4.7.6 Prueba de evaluación 5

En la prueba de evaluación número 5 se muestran los resultados del uso de la aplicación realizado por una usuaria de 30 años, con conocimientos medios de Informática. El resumen de la llamada realizada por la usuaria fue el siguiente:

- El Mundo > Economía > General
- Cadena Cope > Cultura > Cope

Los pasos seguidos una vez realizada la llamada fueron los siguientes:

1. Nueva llamada.
2. Elección de fuente de noticias, se eligió el periódico “El Mundo”.
3. Elección de la sección, se eligió “Economía”.
4. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Merkel, España. Se eligió “General”.
5. Escucha de las noticias.
6. Elección de fuente de noticias, se eligió el periódico “Cadena Cope”.
7. Elección de la sección, se eligió “Cultura”.
8. Elección de unos de los temas populares de entre los siguientes que se generaron: General, Cope. Se eligió “Cope”.
9. Escucha de las noticias.
10. Fin de la llamada.

Estadísticas de la prueba:

- ¿Éxito?: Sí
- Nº de errores: 0
- Turnos de diálogo: 12 (entre usuario y sistema)
- Duración de la llamada: 3 minutos y 35 segundos.
- Nº repetición de frases (Reprompts): 0

Capítulo 5

Conclusiones y Líneas Futuras

En este capítulo se comentan las conclusiones principales del trabajo realizado para el Proyecto de Fin de Carrera, así como las principales líneas de trabajo futuro propuestas.

5.1 Conclusiones

Una de las principales características que pueden diferenciar a la aplicación desarrollada, de otras aplicaciones para la consulta de noticias, es la completa funcionalidad que nos ofrece poder acceder a ella simplemente con una llamada telefónica, ya que se consigue que la aplicación pueda ser utilizada en cualquier parte, en cualquier lugar y a cualquier hora. Esta posibilidad la convierte en un método idóneo de información cuando, por ejemplo, vamos de camino al trabajo o a casa.

Otra de las principales características de la aplicación es su acceso mediante la voz, lo que incrementa notablemente su accesibilidad, especialmente para personas que posean discapacidades visuales o motoras. Además, esta modalidad de acceso facilita que el uso de la aplicación sea realmente sencillo.

Por lo tanto, se podría definir a la aplicación como móvil y accesible, características que aumentan la posibilidad de su comercialización.

Una vez que se ha finalizado el proyecto podemos llegar a la conclusión de que se ha cumplido el principal objetivo para el que se desarrolló el mismo, consistente en realizar un profundo estudio sobre el lenguaje VoiceXML y las posibilidades que este estándar ofrece, desarrollando adicionalmente una aplicación que combina la utilización de este lenguaje con lenguajes y tecnologías de amplio uso en Internet.

Finalizado el Proyecto de Fin de Carrera puedo concluir que mis conocimientos sobre este lenguaje son muy altos y extensos.

Respecto a la experiencia vivida durante el desarrollo de este trabajo, puedo decir que me ha ayudado a reforzar y ampliar mis conocimientos adquiridos durante los años de carrera. Además he conseguido aprender lenguajes y tecnologías nuevas, como PHP, MySQL, VoiceXML o JavaScript.

Además, algunas de las asignaturas cursadas durante la carrera (como, por ejemplo, Diseño de Bases de Datos, Gestión de Proyectos, Ingeniería del Software o Sistemas Hipermedia) me han sido de gran utilidad durante las distintas fases del proyecto.

Gracias a la realización de este proyecto he podido darme cuenta de todo lo que conlleva el desarrollo de cualquier aplicación o producto software, desde su diseño y planificación hasta su desarrollo, implementación y evaluación, y cómo van apareciendo situaciones inesperadas que debemos resolver sobre la marcha. Sé que todavía se aleja de lo que sería desarrollar un proyecto en una empresa del mundo laboral, pero me ha ayudado a comprender mejor lo que sería trabajar en el desarrollo de un proyecto comercial y a acercarme un visión del mismo distinta de la que tenía cuando se desarrollaban las prácticas de las diferentes asignaturas cursadas durante la carrera.

5.2 Líneas futuras

Respecto a las líneas de trabajo futuras existen numerosas formas de mejorar el rendimiento y las funcionalidades de la aplicación. Una de las más comentadas por los usuarios que probaron la aplicación fue la de mejorar las prestaciones del reconocedor automático de voz. Esta tarea es una de las más complicadas y en las que más se está trabajando actualmente en el desarrollo de sistemas de diálogo. La posibilidad de que el reconocimiento vocal sea casi perfecto es una idea, que en un futuro esperamos que sea posible pero que de momento hay que conformarse con los resultados que nos ofrecen las tecnologías actuales.

En cuanto a las posibilidades de mejora de las funcionalidades de la aplicación desarrollada, se podría estudiar la posibilidad de aumentar la personalización de las opciones y funcionalidades del sistema, así cada usuario sería reconocido como único cuando accede a la aplicación y podría guardarse las fuentes y secciones que desea consultar en sus llamadas, con lo que se ahorraría el paso por los menús de selección. Respecto a esta personalización cabe destacar además la posibilidad de poder añadir nuevos idiomas a la aplicación.

La ampliación de las fuentes y categorías que se pueden consultar sería otra de los puntos a mejorar en el futuro para la aplicación. La arquitectura de la aplicación nos ofrece un abanico casi infinito para añadir fuentes de noticias para consultar, el único requisito es que se cuente con un *feed* RSS, del que poder obtener las noticias. Debido a este hecho, la aplicación podría llegar a derivar en un lector de contenidos RSS por voz, extendiendo la extensión actual de únicamente noticias y llegando a poder leer desde actualizaciones de un blog, hasta noticias de cualquier tema específico que sea de interés para el usuario.

En lo referente a las líneas de investigación futuras para los sistemas de diálogo, los entornos en los que el usuario no puede disponer de las manos para el manejo del dispositivo en cuestión tienen una gran importancia, como es el caso de aplicaciones en sistemas GPS, manos libres para el automóvil e incluso, pensando en un futuro más lejano, la posibilidad de realizar la conducción de un automóvil por voz. Además de esto se están realizando avances significativos en campos como, por ejemplo, la Domótica, llevándose a cabo la aplicación de los interfaces orales para el desarrollo de sistemas que sean capaces de controlar, por ejemplo, la temperatura de la calefacción del hogar, el aire acondicionado, persianas, luz, etc. Este tipo de mejoras están consideradas como el futuro de los denominados hogares inteligentes.

Como hemos visto las posibilidades de futuro para los sistemas basados en diálogo son infinitas, por lo que en los próximos años este tipo de sistemas darán mucho de qué hablar, valga la redundancia.

Glosario

Gestión del diálogo: Módulo o proceso que se encuentra dentro de cualquier sistema de dialogo, y en el que el sistema se encargará de dar sentido a la secuencia de sonidos detectada y producir una contestación o respuesta que satisfaga al usuario o el objetivo de la aplicación.

Gramática: Base estructurada de palabras que será capaz de reconocer el sistema de diálogo, esta base debe de estar en un formato determinado y contendrá todas las palabras o secuencias de palabras que reconocerá el sistema.

Script: Un *script* (cuya traducción literal es 'guion') o archivo de órdenes o archivo de procesamiento por lotes es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los script son casi siempre interpretados, pero no todo programa interpretado es considerado un script. El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

PHP: Hypertext Pre-processor.

Prompt: Secuencia de sonidos y palabras que utiliza el sistema de diálogo para comunicarse con el usuario

Reconocimiento Automático del Habla: Proceso por el cual un sistema o aplicación es capaz de reconocer expresiones completas en diálogos y generar respuestas en concordancia con estas expresiones.

RSS: Son las siglas de **Really Simple Syndication**, un formato XML para sindicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. (http://tools.ietf.org/id/draft-nottingham-rss-media-type-00.txt)

Sintetizador de Voz: Se trata de uno de los módulos que componen un sistema de diálogo, es la parte que se encarga de pasar las secuencias de palabras o voz que ha

generado el sistema y convertirlas en sonido audible por el usuario, también es usado para leer los “prompts” al usuario.

URL: Un localizador uniforme de recursos, más comúnmente denominado URL (sigla en inglés de *Uniform Resource Locator*), es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones digitales, etc.

URI: Un URI es una cadena corta de caracteres que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, etc.). Normalmente estos recursos son accesibles en una red o sistema. (http://tools.ietf.org/html/rfc3986).

VoiceXML: Lenguaje de programación basado en etiquetas, que proporciona una interfaz de programación de alto nivel de recursos de voz y telefonía, para desarrolladores de aplicaciones, proveedores de aplicaciones y fabricantes de equipamientos.

W3C: World Wide Web Consortium.

XML: Extensible Markup Language.

Bibliografía

Aist, G., Dowding, J., Hockey, B.A., Rayner, M., Hieronymus, J., Bohus, D., Boven, B., Blaylock, N., Campana, E., Early, S., Gorrell, G., Phan, S.: Talking through procedures: An intelligent Space Station procedure assistant. Proc. of Demo Session at EACL-2003. 2-3. Budapest, Hungría. 2003.

Allen J. y Perault C.R.: Analyzing intentions in dialogues. E Artificial Intelligence, volumen 15(3), pp. 143-178.(1980)

Appelt, D.E.: Planning English Sentences. (Cambridge University Press, 1985, 1ª Edición)

Bohus D., Rudnicky A.I.: The RavenClaw dialog management framework: Architecture and systems, Computer Speech & Language 23(3), pp. 332-361. (2009)

Broekstra J.,Kampman A., Van Hamelen F., Condori: A generic Architecture for Storing and Querying RDF and RDF Schema. Seminario Técnicas de bases de datos para la web. 4-7. (2002).

Callejas Z.: Desarrollo de sistemas de diálogo oral adaptativos y portables. Reconocimiento de emociones, adaptación al idioma y evaluación del campo. Tesis Doctoral, Universidad de Granada, 2008.

Cohen, P.R. y Levesque H.J.: Rational interaction as the basis for communication.SRI International. 433, pp. 1-5. (1988).

Dybkjær L.,Bernsen N.: The DISC Project. The Maersk Mc-Kinney Moller Institute for Production Technology. 1997.

Farrús M., Anguita J., Hernándo J., Cerdà R.: Fusión de sistemas de reconocimiento basados en características de alto y bajo nivel. Universitat Politécnica de Catalunya. Universitat de Barcelona.2005.

Gauvain J., Lamel L., Adda G.: Lightly supervised and unsupervised acoustic model training. Computer Speech & Language 16, 1. pp 115-129. (2002).

Griol D.: Desarrollo y evaluación de diferentes metodologías para la gestión automática del diálogo. Tesis Doctoral Universidad Politécnica de Valencia, 2007.

Hirschman L.: Overview of the DARPA Speech and natural language workshop. Unisys Defense Systems. 1989.

Hubal R., Guinn C., Frank G.: Avatalk. Virtual humans for training with computer generated forces. Research Triangle Institute. 2000.

Hurtado L., Blat F., García F., Grau S., Griol D., Sanchis E., Segarra E., Torres F.: Sistema de diálogo para el proyecto DIHANA. Procesamiento del Lenguaje Natural. 35, pp. 453-454. (2005).

Jordán A.:, La lengua española y las nuevas tecnologías. Carnegie Mellon University. Instituto Cervantes. 1992.

Klein, Lemon, Oka: VERMOBIL. The Vermobil protype system- a software engineering perspective. Journal of Natural Language Engineering. 5, pp. 95-112. (1999).

Delgado R., Araki M.: Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment. (J. Wiley & Sons. 1ª Edición. 2005.)

McTear M.: Spoken dialogue technology: toward the conversational user interface. Computational Linguistics. 31, 3, pp. 404-405. (2004).

Minker W.: Stochastic versus rule-based speech understanding for information retrieval. Speech Communication, 25(4), pp 223-247. (1998).

Oh A., Rudnicky A.: Stochastic language generation for spoken dialogue systems. En: Proc. Of ANLP/NAACL 2000 Workshop on Conversational systems. Seattle, Estados Unidos, 2000.

O'Neill I., Hanna P., Liu X., McTear M.: The Queen's Communicator: An Object-Oriented Dialogue Manage. EuroSpeech '03. Geneva. 2003.

Peckham J., SUNDIAL. A new generation of spoken dialogue systems. Results and lessons from SUNDIAL project. EuroSpeech'93. Berlin, Alemania. 1993.

Guha R.: Innovators of the net. Netscape Communications Corporation. 1999.

RSS Advisory Board, Really Simple Syndication specifications. 1999. 2009.

Segarra E., Sanchis E., García F., y Hurtado L.F.: Extracting semantic information through automatic learning techniques. En International Journal of Pattern Recognition and Arti_cial Intelligence, volumen 16(3), Salt Lake City (Estados Unidos). 2002.

Stallard D.: Talk'n'travel: a conversational system for air travel planning**.** BBN Tecnologies. 70 pp. 68-75 (2000).

Traum D., Bos J., Cooper R., Larsson S., Lewin I., Matheson C., y Poesio M.: A model of dialogue moves and information state revision. (Trindi, 1ª Edición, 1999).

W3C, Semantic Interpretation for speech recognition.2001.2007.

W3C, XHTML+Voice estándar 1.0. 2001.

W3C, Speech Recognition Grammar Specification version 1.0. 2004.

W3C, Voice Extensible Markup Language (VoiceXML).2001. Última revisión 2004.