



Universidad  
Carlos III de Madrid

Ingeniería Técnica en Informática de Gestión

## **PROYECTO FIN DE CARRERA**

# **DESARROLLO DE UN ENTORNO PARA LA INTERACCIÓN MULTIMODAL CON DIFERENTES APLICACIONES EN XHTML+VOICE**

Autor: Víctor Corrales Muñoz

Tutor/Director: Dr. David Griol Barres

Leganés, diciembre de 2010



**Título:** Desarrollo de un Entorno para la Interacción Multimodal con Diferentes Aplicaciones en XHTML+Voice

**Autor:** Víctor Corrales Muñoz

**Director:** Dr. 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

Un profundo y sincero gracias a mi familia por no permitirme caer y empujarme siempre a seguir luchando. A ti por hacerme entender sin lugar a dudas que soy y seré informático. Y a David Griol, por sus incansables palabras de aliento, aún cuando el reto parecía imposible. A todos, gracias.

# Resumen

A lo largo del viaje a través de estas páginas me haré cargo de descubrirles la aplicación que me he esforzado por, no sólo diseñar, sino que sea lo más perfecta posible. Un entorno Web formado por cinco aplicaciones útiles para el usuario capaz de ser manejado completamente por la voz, no sólo mediante el método tradicional de teclado y ratón. Comenzando por el estado del arte y los objetivos que han permitido su creación, pasando por la descripción detallada de todo el entorno y su utilización, hasta llegar a las conclusiones. Una vez inmersos en la descripción del entorno, me permitiré desglosar la aplicación en las cinco aplicaciones de las que se compone, detallando el funcionamiento, diseño, arquitectura, y todas las pruebas usadas de cada una de ellas que han ido limitando los errores hasta reducirlos prácticamente a cero.

El presente Proyecto Final de Carrera centrará su atención en el estudio y desarrollo de una aplicación basada en la tecnología que permite la interacción multimodal entre usuario y máquina. La multimodalidad empleada está centrada esencialmente en la interacción oral. Las tecnologías utilizadas están basadas en Sistemas de Diálogo Orales integrados en programas tales como el navegador web Opera, y desarrollados mediante aplicaciones diseñadas mediante lenguajes de programación como XHTML+Voice.

La aplicación descrita en este manual se denomina *Voice Applications*, y compuesta a su vez de cinco módulos: *Voice Dictionary*, que permite al usuario búsquedas de contenidos en plataformas como Wikipedia; *Voice Pronunciations*, que favorece y ayuda a ejercitar la pronunciación en lengua inglesa a través de divertidos juegos de palabras e imágenes; *Voice Library*, con acceso a una base de datos de trescientos libros que pueden ser consultados por el usuario; *Voice Videoclub*, una completa aplicación de búsqueda de películas en una base de datos con más de novecientas películas; y *Voice Google*, como un completo, rápido y efectivo buscador web. Todos y cada uno de estos módulos pertenecientes a la aplicación pueden ser manejados por el usuario a través del teclado y ratón, o a través de su propia voz, obteniendo también los resultados mediante la voz.

La interacción oral de la aplicación con el usuario permite el acceso a usuarios que no puedan utilizar los tradicionales teclado y ratón, ya sea por comodidad, por obligación, o por necesidad, favoreciendo en este último caso la integración de persona con discapacidades motoras o visuales al mundo de internet, con el objetivo siempre en mente de ayudar a la eliminación las barreras arquitectónicas que lastran el uso de las nuevas tecnologías a determinados sectores de la población.

**Palabras clave:** Entorno Web Multimodal, Interacción Oral. VoiceXML, XHTML+Voice, Sistemas de diálogo.

# Abstract

We will travel through these pages to describe the application that I have designed, and I made the effort to obtain that the application was most perfect possible. This consists of a web environment with five useful applications designed for the interaction with the user by means of the voice, and not only by means of the traditional interfaces like the keyboard or mouse. We begin by the state-of-the-art and the objectives that have made this creation possible, followed by a detailed description of the web environment and its uses, until finishing this document with the conclusions. Once immersed in the environment's description, we will detail the five applications that make it up, detailing the main functionalities, how they work, their design, architecture, and the tests that have been used to limit the errors until practically reducing them to zero.

This report is then focused on the study and development of an application based on the technology that allows a multimodal interaction between user and machine. The multimodality used in the application is centered in the oral interaction. Technologies used in the application are based on Dialog Systems and the Opera Web Browser, and developed using programming languages like XHTML+Voice.

The application described in this report is called *Voice Applications*, and it is composed by five modules: *Voice Dictionary*, which allows users to orally search contents in platforms like Wikipedia; *Voice Pronunciation*, which facilitates exercising the articulation in English through funny games based on words completing and images; *Voice Library*, which access a database of three hundred books that can be consulted by the user; *Voice Videoclub*, a complete browser of films using a database with more than nine hundred films; and *Voice Google*, like a complete, fast and effective Oral Web Browser. All these modules can be interacted with the keyboard and mouse, or by means of the voice, also presenting orally the results. Oral interaction allows the access to Internet for users with motor or visual disabilities, eliminating architectonic barriers that ballast avoids the use of new technologies.

**Keywords:** Multimodal Web Systems, Multimodality, Oral Interaction, VoiceXML, XHTML+Voice, Dialogue Systems.





# Índice general

<b>1. INTRODUCCIÓN Y OBJETIVOS .....</b>	<b>2</b>
1.1 Introducción .....	2
1.2 Objetivos .....	11
1.2.1 <i>Objetivos del Proyecto Final de Carrera</i> .....	12
1.2.2 <i>Objetivos particulares de la aplicación</i> .....	13
1.2.3 <i>Otros objetivos</i> .....	16
1.3 Fases del desarrollo .....	18
1.3.1 <i>Planificación y Documentación</i> .....	18
1.3.2 <i>Análisis y Diseño Inicial</i> .....	19
1.3.3 <i>Diseño de Interfaz</i> .....	23
1.3.4 <i>Desarrollo Técnico</i> .....	24
1.4 Medios empleados.....	26
1.4.1 <i>Lenguajes de Programación</i> .....	26
1.4.2 <i>Programas empleados</i> .....	27
1.4.3 <i>Manuales</i> .....	28
1.5 Planificación Temporal .....	28
1.6 Estructura de la memoria .....	29
<b>2. ESTADO DEL ARTE.....</b>	<b>1</b>
2.1 Introducción .....	1
2.2 Sistemas de Diálogo Oral.....	2

2.2.1 Reconocimiento Automático del Habla .....	3
2.2.2 Procesamiento del Lenguaje Natural .....	4
2.2.3 Gestión del Diálogo.....	5
2.2.4 Generación del Lenguaje Natural .....	5
2.2.5 Síntesis de Texto y Voz.....	6
2.3 Historia de los Sistema de Diálogo .....	7
2.4 Estándares.....	12
2.4.1 VoiceXML.....	13
2.4.2 XHTML.....	20
2.4.3 XML Events.....	25
2.4.4 DOM .....	27
2.4.5 Gramáticas SRGF y JSRGF.....	31
2.5 Ejemplos de SDOs.....	37
<b>3. ENTORNO X+V: VOICE APPLICATIONS .....</b>	<b>2</b>
3.1 Introducción .....	2
3.2 Voice Dictionary .....	7
3.2.1 Funcionamiento .....	8
3.2.2 Diseño.....	12
3.2.3 Arquitectura.....	13
3.2.4 Pruebas y limitación de errores .....	19
3.3 Voice Pronunciation .....	20
3.3.1 Funcionamiento .....	21
3.3.2 Diseño.....	22
3.3.3 Arquitectura.....	28
3.3.4 Pruebas y limitación de errores .....	31
3.4 Voice Library & Voice Videoclub .....	31
3.4.1 Funcionamiento .....	33
3.4.2 Diseño.....	34
3.4.3 Arquitectura.....	35
3.4.4 Pruebas y limitación de errores .....	37
3.4.5 Gestión privada de bases de datos .....	56
3.5 Voice Google.....	58
3.5.1 Funcionamiento .....	59
3.5.2 Diseño.....	60
3.5.3 Arquitectura.....	62
3.5.4 Pruebas y limitación de errores .....	64

<b>4. CONCLUSIONES .....</b>	<b>75</b>
4.1 Resumen .....	75
4.2 Trabajo Futuro.....	79
4.3 Conclusiones Personales .....	81

# Índice de figuras

<i>Figura 1. Multimodal Flight Query</i> .....	6
<i>Figura 2. Diagrama de acceso USUARIO - APLICACIÓN</i> .....	15
<i>Figura 3. Web Accessibility Initiative</i> .....	17
<i>Figura 4. CAST (Center for Applied Special Technology)</i> .....	17
<i>Figura 5. Base de datos de aplicación Voice Library – tabla books</i> .....	20
<i>Figura 6. Base de datos de aplicación Voice Videoclub – tabla films</i> .....	20
<i>Figura 7. Base de datos de aplicación Voice Pronunciation – tabla words</i> .....	21
<i>Figura 8. Base de datos de aplicación Voice Pronunciation – tabla picture</i> .....	21
<i>Figura 8. Diagrama de Gantt: planificación temporal del proyecto</i> .....	29
<i>Figura 10. Arquitectura Modular de los Sistemas de Diálogo Oral</i> .....	2
<i>Figura 11. Arquitectura del modelo VoiceXML</i> .....	14
<i>Figura 12. Diagrama dl flujo de eventos en DOM2</i> .....	25
<i>Figura 13. Presentación de Voice Applications</i> .....	5
<i>Figura 14. Presentación de aplicación Voice Dictionary</i> .....	7
<i>Figura 15. Aplicación VD con las instrucciones delplegadas</i> .....	9
<i>Figura 16. Flujo de información de VD</i> .....	14
<i>Figura 17. Código de información de cabecera.</i> .....	15
<i>Figura 18. Código de información de cabecera.</i> .....	16
<i>Figura 19. Diseño de pantalla inicial de VP.</i> .....	23

<i>Figura 20. Diseño de sección Instructions de VP.</i>	25
<i>Figura 21. Diseño de juego Words de VP.</i>	26
<i>Figura 22. Diseño de juego Pictures de VP.</i>	27
<i>Figura 23. Flujo de información de la aplicación VP.</i>	28
<i>Figura 24. Página principal de VL.</i>	32
<i>Figura 25. Página principal de VV.</i>	32
<i>Figura 26. Flujo de información de las aplicaciones VL y VV.</i>	36
<i>Figura 27. Resultados de PRUEBA #1 sobre VL.</i>	40
<i>Figura 28. Resultados de PRUEBA #2 sobre VL.</i>	43
<i>Figura 29. Resultados de PRUEBA #3 sobre VL.</i>	44
<i>Figura 30. Resultados de PRUEBA #4 sobre VV.</i>	49
<i>Figura 31. Resultados de PRUEBA #5 sobre VV.</i>	53
<i>Figura 32. Resultados de PRUEBA #6 sobre VV.</i>	55
<i>Figura 33. Menú de acceso al sistema gestor de BBDD.</i>	56
<i>Figura 34. Menú de gestión de la BBDD Videoclub.</i>	57
<i>Figura 35. Página principal de la aplicación VG.</i>	61
<i>Figura 36. Flujo de información de la aplicación VG.</i>	63
<i>Figura 37. Resultados de la Prueba #1 sobre VG.</i>	67
<i>Figura 38. Resultados de la Prueba #2 sobre VG.</i>	70
<i>Figura 39. Resultados de la Prueba #3 sobre VG.</i>	73



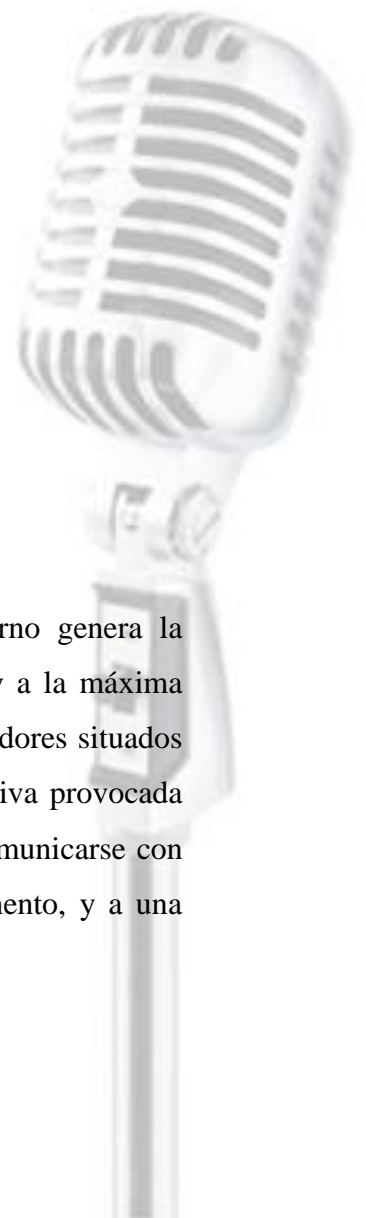


# Capítulo 1

## *Introducción y objetivos*

### 1.1 Introducción

El auge de las tecnologías de la información en el mundo moderno genera la necesidad cada vez mayor de acceder a los datos desde cualquier sitio y a la máxima velocidad posible. Hace tiempo era impensable poder conectar dos ordenadores situados en diferentes lugares del planeta, pero hoy en día se genera una expectativa provocada por el desarrollo de estas tecnologías que permite acceder a Internet o comunicarse con cualquier persona prácticamente desde cualquier sitio, en cualquier momento, y a una velocidad casi instantánea.





Adaptar estos accesos y comunicaciones a dispositivos provistos de interfaces que permiten su control a través de la voz, o dispositivos con acceso a aplicaciones controladas por voz, provoca un gran avance en la industria de las tecnologías wireless, y en el entorno de Internet. Esta interacción mediante la voz se consigue a través de la tecnología X+V, que permite el desarrollo de entornos de usuario multimodales.

X+V [\[1.1-A\]](#) son la iniciales de XHTML+VoiceXML, dos lenguajes de programación independientes, cohesionados para dar forma a un nuevo sistema de comunicación con el entorno Web. XHTML [\[1.1-B\]](#) a su vez nace de la interacción de las mayores ventajas de XML sobre el popular HTML 4.0, de manera que podemos contemplar como el tradicional lenguaje de marcado o etiquetas (*tags*) HTML se ve impregnado de las especificaciones más estrictas de XML. Esta combinación de lenguajes genera una gran sencillez de compilación gracias a la eliminación de las heurísticas que se utilizaban para saber qué quiso escribir el programador en el código cuando se encuentra incompleto, por lo que se genera un código mucho más ágil. Por otro lado, VoiceXML [\[1.1-C\]](#) es un lenguaje de marcas desarrollado por el W3C [\[1.1-D\]](#), creado para transformar el habla en texto, de manera que, mediante mecanismos implementados en navegadores como Opera [\[1.1-E\]](#) (mecanismos como SRGS [\[1.1-F\]](#)), permite crear diálogos con los que se puede interactuar, tanto narrando contenidos al usuario, como recibiendo comandos de navegación o ejecución por voz. Ambos lenguajes crean la posibilidad de diseñar un entorno agradable al usuario que le permita navegar a través de su dispositivo móvil con la voz mientras conduce, o que le permite controlar su dispositivo sin necesidad de ver los elementos gráficos y visuales mostrados para su manejo.

Los avances tecnológicos han favorecido la creación de tecnologías cada vez con tamaños más reducidos, capaces de ejecutar aplicaciones y acceder a los datos mediante conexiones inalámbricas, como por ejemplo las PDAs y los teléfonos Smartphone o teléfonos inteligentes. En cambio, mientras que los dispositivos de comunicación se hacen cada vez más pequeños y útiles, la fisiología de las personas permanece inalterada, las manos y dedos de las personas siguen siendo iguales. Para facilitar su uso las interfaces diseñadas tienden a la combinación de entradas a través del tradicional teclado y ratón con tecnologías interactivas de entradas por voz. A este tipo de interacción, en la que el usuario tiene más de una forma de acceso a los datos utilizando modos humanos

(la voz o habla, gestos, movimientos, ojos...), se llama multimodal. El mundo de la tecnología wireless (inalámbricas) está convirtiendo estas formas de interacción en la norma a una velocidad asombrosa, favoreciendo a su vez un modo de comunicación entre el usuario y la máquina mucho más natural para el usuario. Atrás quedan los tiempos en los que el usuario debía adaptarse sin condiciones a los requisitos impuestos por las limitaciones del desarrollo tecnológico, obligados a aprender complicados manuales para poder manejar aplicaciones de tareas relativamente sencillas. Usuarios adaptados a las máquinas. Ahora, gracias al floreciente desarrollo de la tecnología multimodal, las máquinas cada vez están más adaptadas a las condiciones de interacción humana.

Antiguamente era necesario sentarse delante de un ordenador conectado a un módem para poder acceder a su correo electrónico, ver las noticias, o comprobar su cuenta bancaria. Actualmente esa tarea puede ser realizada desde el banco de un parque, o mientras camina desde la oficina al coche. Incorporar estos nuevos dispositivos a las nuevas maneras de comunicación, a nuevos ambientes de conexión requiere nuevas formas de acceder a los datos. Facilitar al usuario la posibilidad de cambiar entre los diferentes modos de interacción, principalmente el uso de audio únicamente (con manos libres y sin conexión visual con el dispositivo) favorece la auténtica movilidad del dispositivo. No sólo realizando tareas multimodales para acceder a los recursos de Internet, sino integrando esta multimodalidad en la interfaz del dispositivo. Con la introducción de una interfaz oral y la consiguiente comunicación con la aplicación se persiguen tres objetivos fundamentales:

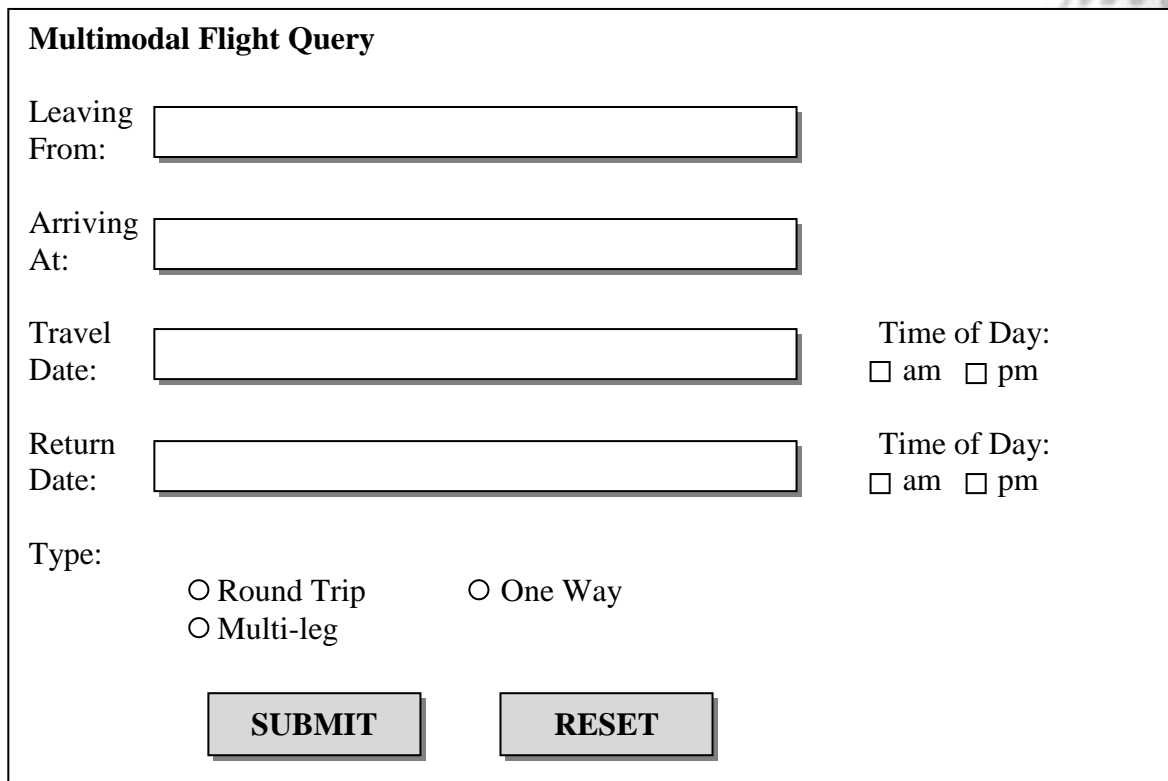
- 1) Conseguir que la comunicación con el entorno sea lo más natural posible, es decir, conseguir que la interacción o diálogo sea lo más parecido posible al diálogo que podría establecerse entre dos personas.
- 2) Permitir el uso de aplicaciones en entornos en los que no podrían ser utilizadas debido al uso exclusivo del tradicional sistema de teclado y ratón. Este entorno podría ser por ejemplo un automóvil.
- 3) Facilitar el acceso a las aplicaciones a personas con discapacidades visuales o motoras, favoreciendo la eliminación de barreras arquitectónicas de Internet y el desarrollo para la integración de personas con discapacidades.

Basado en estándares testados, el lenguaje X+V permite al desarrollador Web implementar en su entorno tanto elementos visuales como elementos de voz. Esta fácil integración de ambos elementos logra brindar de gran versatilidad al campo de desarrollo de interfaces multimodales. La entrada paralela, tanto por voz como por teclado, dota de velocidad al entorno al permitir accesos más rápidos, además de añadir valor a la experiencia general de la interacción. Esta dualidad reduce el esfuerzo excesivo que puede resultar de la interacción con sistemas de modalidad única.

El lenguaje HTML era el estándar usado para desarrollar sistemas Web, pero durante estos últimos años, ha ido siendo suplantado por el lenguaje favorito del W3C: el lenguaje XHTML, que da la posibilidad de inserción de etiquetas XML en el código de HTML. Estas etiquetas integradas son, por ejemplo, las utilizadas por VoiceXML. De esta manera podemos diseñar cualquier elemento de XHTML (ya sean campos de formularios como *inputs* de entrada, *check-box*, avisos de texto, y así sucesivamente) para que se muestre gráfica y visualmente, y dotarle de voz gracias a VoiceXML, favoreciendo la combinación con estos elementos visuales mediante la voz. Así podríamos rellenar el formulario Web mediante el teclado y el ratón, o con nuestra voz sin ningún problema. Incluso completar interfaces más sofisticadas realizando tareas de comprobación de campos de entrada mediante scripting, como es el caso del lenguaje JavaScript.

El estándar del subconjunto del lenguaje de marcas VoiceXML ha sido desarrollado por los expertos en la industria de la voz y mejorado durante años. Dicho estándar ha superado las pruebas realizadas en situaciones reales y complejas, como por ejemplo en centros de atención telefónica, testadas por los desarrolladores encargados de programar y probar el lenguaje. VoiceXML ha sido dado a conocer como un lenguaje capaz de permitir el desarrollo de una amplia gama de aplicaciones, que junto con la experiencia X+V rompe las barreras de acceso y se asienta en un mundo sin límites a la hora de idear nuevos algoritmos y aplicaciones.

En la Figura 1 podemos observar un típico formulario, obtenido del manual del programador de XHTML+Voice diseñado por IBM. Este formulario se compone de varios campos de entrada y check-box, de manera que se pueda completar la información de un vuelo correctamente. La multimodalidad de este formulario permite, no sólo completar los campos a golpe de tecla y ratón, sino completarlo respondiendo con el habla a los avisos de voz que el propio formulario pide al usuario para ingresar los datos en los campos a medida que los va necesitando. El algoritmo reconoce los datos suministrados mediante voz y los inserta en función del momento de diálogo en el que se encuentre, o en función de los datos aportados. A este tipo de interacción, en la que la máquina pide un dato, el usuario lo aporta, y la máquina pasa al siguiente dato del formulario, se le denomina “*directed dialog*”.



**Multimodal Flight Query**

Leaving From:

Arriving At:

Travel Date:  Time of Day:  am  pm

Return Date:  Time of Day:  am  pm

Type:

Round Trip       One Way

Multi-leg

Figura 1. Multimodal Flight Query  
(XHTML+Voice Programmer's Guide. IBM)

Una interacción más avanzada permite introducir los datos de manera más natural, sin menos interacciones de la máquina. De manera que, mediante algoritmos que identificasen los datos, el usuario podría decir directamente en la misma frase a dónde se dirige, del lugar del que va a partir, las fechas, horas y resto de datos que completan el formulario, siendo la aplicación la que se encargase de seleccionar los datos importantes y distribuirlos entre los campos del formulario. A este tipo de interacción se le denomina “*iniciativa mixta*” [\[1.1-G\]](#).

Observando el código de una aplicación programada en X+V, diferenciamos entre dos clases de etiquetas, las etiquetas visuales (XHTML) y las etiquetas de voz (VoiceXML). El desarrollador Web utiliza las etiquetas visuales para comunicarle a la aplicación cómo debe comportarse cuando recibe los datos a través de teclado, mostrando al usuario nueva información o realizando tareas internas, entre otros comportamientos. Para esta clase de comportamientos, la aplicación utiliza el recurso del buscador llamado motor gráfico. Las etiquetas de voz se utilizan de igual manera para dirigir a la aplicación en caso de recibir información a través del canal de audio, de manera que una orden o información a través de la voz del usuario provoque una respuesta auditiva, visual o de comportamiento interno en la aplicación, utilizando el motor de discurso del navegador Web. De la misma forma que las etiquetas visuales invocan interfaz visual, las etiquetas de voz invocan la interfaz de voz, diferenciando así entre los múltiples comportamientos de las entradas multimodales. Ambos interfaces deben realizar rupturas y tareas de modularidad, de manera que al final únicamente quedan los componentes básicos (como por ejemplo en la Figura 1 los campos dedicados a las horas o los box de “am” y “pm”). Después de diseñar cada componente en su interfaz hay que enlazar dichos componentes para que queden finalmente asociados, y los campos puedan ser completados tanto por voz como por teclado.

El manual del programador de IBM [1.1-H] realiza las siguientes preguntas al programador para realizar una buena labor con el código de su aplicación:

- ¿Qué palabras debe decir o resumir el Gestor del Diálogo?
- ¿Qué palabras o frases va a recibir el Gestor del Diálogo?
- ¿Qué debe hacer la aplicación si el Reconocedor Automático del Habla no reconoce una palabra o una frase?
- ¿Cuál debe ser el resultado del Gestor del Diálogo cuando reconozca una palabra o una frase?

Al responder a estas cuestiones, el desarrollador de X+V tiene una llave muy valiosa para comenzar a programar. Una aplicación con una interfaz de voz tiene que hablar al usuario, realizarle una petición y predecir qué es lo que va a recibir del usuario. Cuando lo recibe, debe prever que la respuesta puede ser correcta o no, y actuar en consecuencia para ambos casos. Así pues, con las etiquetas de voz y las respuestas a las preguntas programadas como discursos e interpretadas como recortes de voz, ya se tiene prácticamente completada la capa de presentación de la aplicación Web multimodal. Únicamente sería necesario sincronizar esos campos y etiquetas de voz con los elementos visuales correspondientes (ya que el Gestor de Diálogo sólo puede mantener activa una conversación o recorte a la vez), y saber en qué momento del diálogo debe activarse cada discurso.

Para facilitar las tareas de programación, el entorno de la aplicación Web se encuentra dirigido por eventos, de modo que X+V incorpora el marco *eventing* del DOM (*Document Object Model*) usado en el estándar de eventos del lenguaje XML. Usando este marco, X+V define los tipos de familias de eventos tales como “*on mouse-over*” o “*on input focus*” para especificar funciones o tareas a ejecutar cuando se produzcan estos eventos, y para crear la relación entre elementos visuales y etiquetas de voz. Esta incorporación de los eventos de XML a X+V crea una base estándar y uniforme de modelos *eventing* que permiten la integración de eventos entre los diferentes lenguajes de XML, así como una línea que una y permita comunicarse a los lenguajes de base XML.

Hasta este momento hemos concretado que para crear una aplicación Web multimodal basada en X+V es necesario:

- Contenido de etiquetas o marcas visuales.
- Colección de discursos y etiquetas de voz para cada elemento del interfaz de usuario.
- Las etiquetas de eventos que le digan a la aplicación qué discursos utilizar y cuándo hacerlo.

Para las etiquetas visuales, X+V utiliza el lenguaje XHTML. Para los discursos y etiquetas de voz, utiliza un subconjunto de VoiceXML definido por las formas de construcción de VoiceXML. Para asociar este subconjunto y la interfaz visual creada con XHTML, X+V utiliza el estándar de eventos de XML. Todas estas relaciones se basan en los estándares oficiales para el desarrollo Web denominado IETF (*Internet Engineering Task Force*), el cual gobierna los estándares Web.

Así pues, tomando el ejemplo de la Figura 1, la etiqueta de VoiceXML para el campo de la ciudad de salida tenemos los siguientes elementos:

- Un aviso mediante diálogo que solicita al usuario la ciudad de salida.
- Una gramática dentro del código VoiceXML que enumera todas las posibles ciudades que puede introducir el usuario.
- Un algoritmo que dice al motor de voz dónde debe poner el resultado para que se muestre visualmente al usuario.
- Algoritmos de reconocimiento en caso de no introducir un resultado esperado, es decir, una respuesta que no se encuentre dentro de nuestra gramática de ciudades.

Las gramáticas a las que se hace referencia, son elementos que utilizan los desarrolladores de aplicaciones multimodales para decirle al motor del discurso qué palabras o frases debe reconocer como respuesta del usuario. Este sistema es a la vez una ventaja y una desventaja a la hora de programar. La ventaja es que puedes limitar los contenidos, especificando qué es lo que debe permitir como valor en las variables de resultado, aquello con lo que el discurso debe continuar, omitiendo y generando como

error todo aquello que no se encuentre dentro de la colección de palabras/frases aportadas en la gramática. La desventaja es esto mismo, que el número de elementos dentro de la gramática es limitado, y esto a la vez limita al desarrollador a la hora de dar vida a sus aplicaciones. Esta desventaja ha sido eliminada (como veremos más adelante) gracias a la combinación XHTML+VoiceXML. Este elemento llamado gramática puede verse implícito en las etiquetas del subconjunto de VoiceXML, o ser un archivo externo al que se hace referencia desde X+V.

Pensando en la arquitectura de X+V podemos decir que el documento XHTML es un contenedor, un recipiente que guarda los elementos visuales por un lado (formularios, campos, check-boxes, textos), almacena los elementos de diálogo por otro lado (campos y formularios VoiceXML), y por último los elementos de XML *Events*. De esta manera, contiene e interconecta todos los elementos necesarios para hacer funcionar una aplicación basada en X+V. Estos eventos por un lado identifican el formulario de voz que debe activarse, y por otro lado señalan las condiciones o acontecimientos visuales que activarán dicho formulario.

No obstante, también tenemos la posibilidad de extraer el código del subconjunto de VoiceXML e implementarlo en un archivo separado al de XHTML. Separando las etiquetas de voz de las etiquetas visuales obtenemos más flexibilidad en el desarrollo de la aplicación, pudiendo ejecutar el código de VoiceXML por separado de los elementos visuales, y hacer que se interconecten más adelante.

Otra ventaja de mantener en archivos separados los elementos visuales y los elementos de voz, es la fácil reutilización de las unidades de voz creadas para diferentes páginas de XHTML. De esta forma, en lugar de tener el mismo código de voz implementado en todas y cada una de las páginas, tenemos un único código en un archivo externo del que se aprovechan los elementos visuales del entorno Web multimodal. Este hecho brinda de agilidad al desarrollador a la hora de hacer cambios y modificaciones en la interfaz de voz, ya que en lugar de cambiar todos los archivos en los que hace acto de presencia el mismo código de voz, sólo sería necesario modificar el módulo de voz del archivo externo.



El lenguaje X+V es el último en entrar en la familia de lenguajes basados en XML. Considerando que XHTML fue desarrollado para soportar interfaces visuales y VoiceXML se centra en la voz únicamente, X+V es un híbrido cuyo objetivo es la interconexión de ambos para un óptimo desarrollo multimodal. Los desarrolladores de este lenguaje hacen frente a una demanda cada vez mayor de usuarios que utilizan entradas y salidas multimodales. Los estándares cada vez son más fuertes y flexibles, admitiendo todo un futuro de avances en este campo. Al tener esta base, X+V se presenta como un lenguaje de un grado lingüístico de menor dificultad, pues sólo se necesitan conocimientos en XML y sus estándares para comenzar a programar una buena aplicación Web de interfaz multimodal.



## 1.2 Objetivos

Entrando en los objetivos perseguidos al realizar este proyecto, podemos diferenciarlos entre los objetivos principales del Proyecto Final de Carrera y los que hacen referencia a lo que se persigue al afrontarlo, es decir, los objetivos particulares de la aplicación o lo que se consigue con la innovación de esta nueva aplicación en el mundo de Internet, y el resto de los objetivos que se vislumbran con el desarrollo de esta tecnología.

## **1.2.1 Objetivos del Proyecto Final de Carrera**

Este Proyecto Final de Carrera tiene como principal finalidad el estudio y desarrollo de sistemas Web basados en el estándar de X+V para la creación de entornos multimodales.

El objetivo principal, como se ha visto en el apartado anterior, es la implementación y cohesión de los lenguajes XHTML y VoiceXML, utilizando el primero como base e interfaz visual para el subconjunto del lenguaje VoiceXML (encargado de la interfaz de oral). Ambos lenguajes unidos mediante eventos del estándar XML Events. Esta estructura permite una comunicación oral entre el usuario y el entorno Web.

El objetivo que he visto esencial para mi carrera profesional es el aprendizaje de los lenguajes necesarios para la implementación de todo el entorno Web. Al querer desarrollarlo lo más completo posible, no he escatimado a la hora de afrontarme a lenguajes totalmente desconocidos para mí, investigar formas más eficientes de desarrollo, y solucionar errores de alta complejidad. Todo ello ha sido, no sólo estudiado, sino implementado (tanto la interfaz visual, como la interfaz oral) en inglés, favoreciendo el uso de esta aplicación a nivel internacional, dado que la problemática en cuestión a solventar afecta a usuarios de todo el mundo, y favoreciendo mi crecimiento académico.

## 1.2.2 Objetivos particulares de la aplicación

La aplicación diseñada se trata de un portal que integra a su vez cinco aplicaciones, que hemos llamado *Voice Applications*. Inicialmente se trató la idea de una aplicación que presentase un formulario, narrado al usuario, y completado mediante la voz. Una vez completado el formulario debía realizar un proceso de búsqueda y mostrar/narrar al usuario el contenido del resultado, dándole la opción de volver a realizar una nueva búsqueda o navegar mediante la voz por un sistema de hipervínculos. Sin embargo, en el afán de ir más allá, las cinco aplicaciones abarcan el máximo uso y potencial extraído de X+V.

La primera de las aplicaciones desarrolladas es la llamada *Voice Dictionary*. Recibe del usuario el contenido de la búsqueda, envía esa información a *Wikipedia*, recoge el contenido del resultado, lo procesa y se lo muestra/narra al usuario, permitiéndole realizar una nueva búsqueda, o seleccionar mediante voz en cualquiera de los links que aparecen en el resultado, favoreciendo mediante el sistema de hipertextos el desarrollo de una aplicación viva, que se autogenera con cada nueva búsqueda.

Dos aplicaciones, *Voice Library* y *Voice Videoclub*, mediante búsquedas multimodales permiten al usuario encontrar en ambas bases de datos los libros o películas introducidos en la búsqueda, mostrando los datos de cada uno de ellos y reproduciéndolos mediante voz, permitiéndole igualmente cualquier nueva búsqueda que al usuario le pudiese surgir.

Diseñando sistemas de búsqueda no podía dejar escapar la oportunidad de integrar en un entorno como *Google* una completa interfaz oral con un potente motor del discurso, tanto para búsquedas como para tratar los resultados y el acceso a los resultados. Esta aplicación, *Voice Google*, realiza las búsquedas enviando la información al buscador, procesando la información resultante y mostrándosela y narrándosela al usuario, permitiéndole también el acceso a los links generados.

La última de las aplicaciones está dividida en dos. Se trata de la resolución de una dificultad surgida a lo largo del proceso de desarrollo de este proyecto en inglés: la pronunciación. La aplicación se denomina *Voice Pronunciations*, y contiene dos divertidos juegos de pronunciación. El primero muestra una palabra y narra su descripción, de manera que hay que pronunciarla correctamente. Si no se pronuncia correctamente no se avanza, y en caso de gran dificultad se proporciona la opción de acceder mediante voz o ratón a la solución. Cuando se pronuncia correctamente se pasa a la siguiente palabra y se obtienen diez puntos. El segundo juego en lugar de mostrar una palabra, muestra una imagen y narra una descripción. Con una base de datos de cerca de setecientas imágenes el usuario o jugador se puede ver sorprendido con sencillas imágenes de fácil pronunciación, o difíciles viñetas que incitan a la solución. Cuando se pronuncia correctamente, se muestra la siguiente imagen dando al usuario/jugador diez puntos.

De esta manera se abarcan al máximo las posibilidades ofrecidas por X+V, tanto para accesos a Webs externas como a bases de datos internas, dándole al usuario la posibilidad de navegar por el entorno y utilizar todas y cada una de las aplicaciones sin utilizar, aunque también podría, el ratón o el teclado. Mediante el uso de la voz, en este entorno, se puede sustituir perfectamente al tradicional teclado y ratón, objetivo esencial de la aplicación.

La Figura 2 presenta el flujo de la información a través de todo el sistema diseñado, desde que sale del usuario, hasta que retorna a él. Entra en el sistema y se distribuye a través de cualquiera de las aplicaciones que lo conforman, accediendo a bases de datos y recibiendo información de estas, o realizando el mismo proceso con Internet.

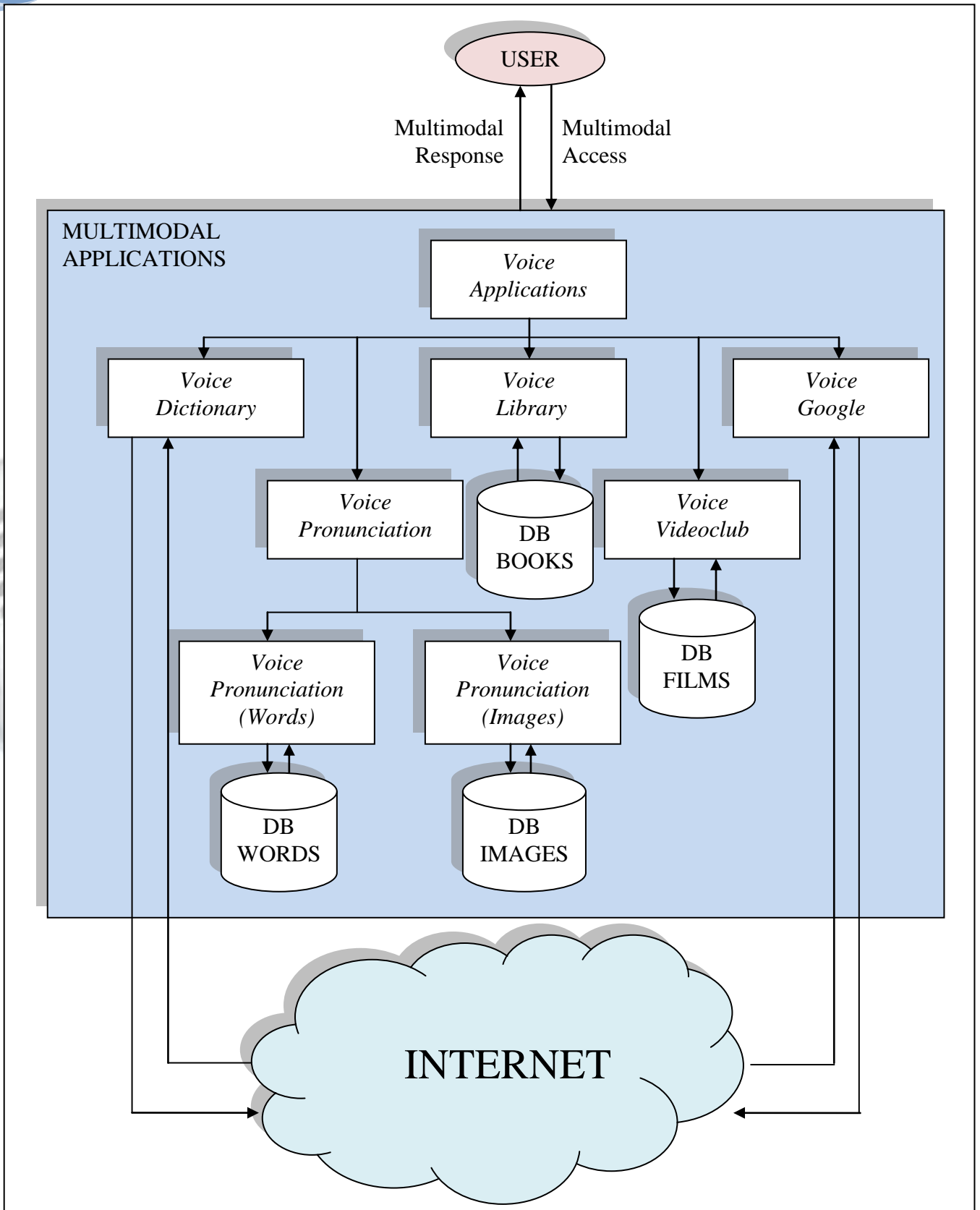


Figura 2. Diagrama de acceso USUARIO - APLICACIÓN

### 1.2.3 Otros objetivos

Es importante destacar como objetivo de este proyecto, no sólo el de facilitar el uso de dispositivos y aplicaciones Web a los usuarios, sino el de eliminar una barrera arquitectónica de Internet para personas discapacitadas. Gracias a X+V eliminamos una barrera muy importante y difícil de eliminar, y favorecemos la integración y acercamiento de personas invidentes y con dificultades motoras al universo de Internet y las nuevas tecnologías.

El desarrollo de las ciudades permite cada vez más edificios accesibles y más vías transitables para discapacitados. Poco a poco se aprueban leyes que exigen esta integración en las estructuras de los urinarios, ascensores, puertas, rampas y otros elementos. Pero el mundo de Internet, a pesar de avanzar vertiginosamente rápido, no crece de la misma manera para personas con discapacidades. Sin embargo existen Webs dedicadas a este propósito. Por ejemplo, la página *Web Accessibility Initiative* [\[1.2.3-A\]](#), mostrada en la Figura 3, aporta guías y documentación para que los desarrolladores Web implementen mejoras en sus entornos que favorezcan una mayor accesibilidad. Estas mejoras se describen de la siguiente forma:

*“Más específicamente, la accesibilidad Web significa que personas con discapacidades puedan percibir, entender, navegar e interactuar con la Web, y que esto pueda contribuir al entorno Web. La accesibilidad Web también beneficia a personas mayores con capacidades disminuidas debido al envejecimiento.”*

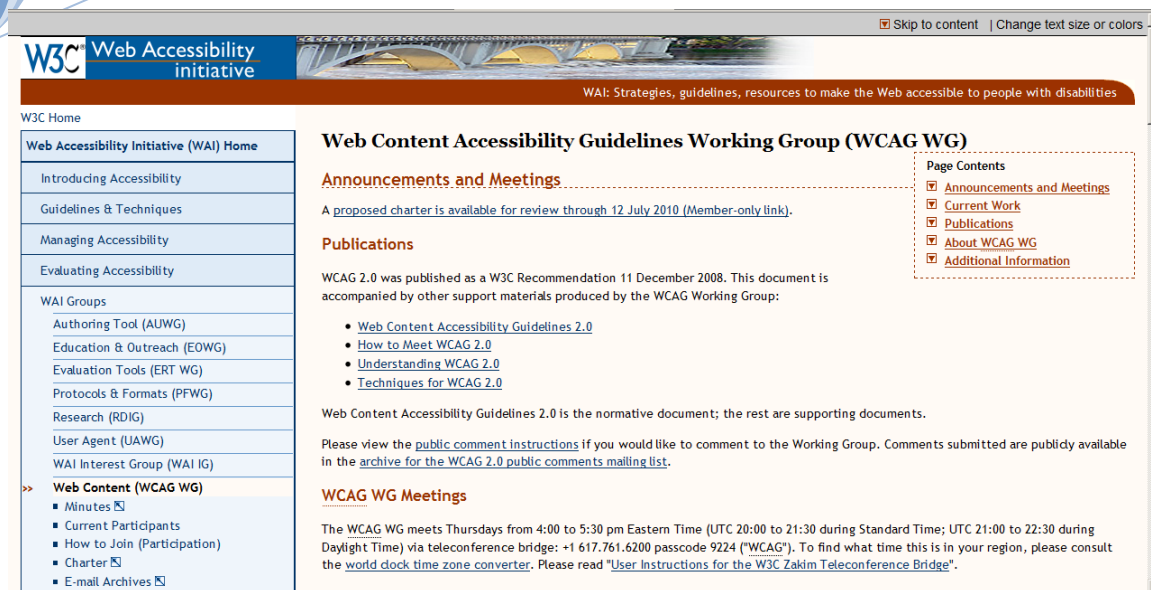


Figura 3. Web Accessibility Initiative  
(<http://www.w3.org/WAI/GL/>)

Otro ejemplo, incluso más práctico si cabe, es el de la Web CAST [\[1.2.3-B\]](#), representado en la Figura 4. Se trata de una organización sin ánimo de lucro dedicada al desarrollo con tecnologías para todas las personas, incluso las personas con discapacidades. Mantienen un sistema llamado “Bobby”, cuyo objetivo es escanear cualquier Web de Internet introducida por el usuario y comprobar el grado de accesibilidad que contiene esa Web. Esa calificación de accesibilidad, si es apta, nos permitirá insertar su logo de “Bobby Approved” en el entorno comprobado.

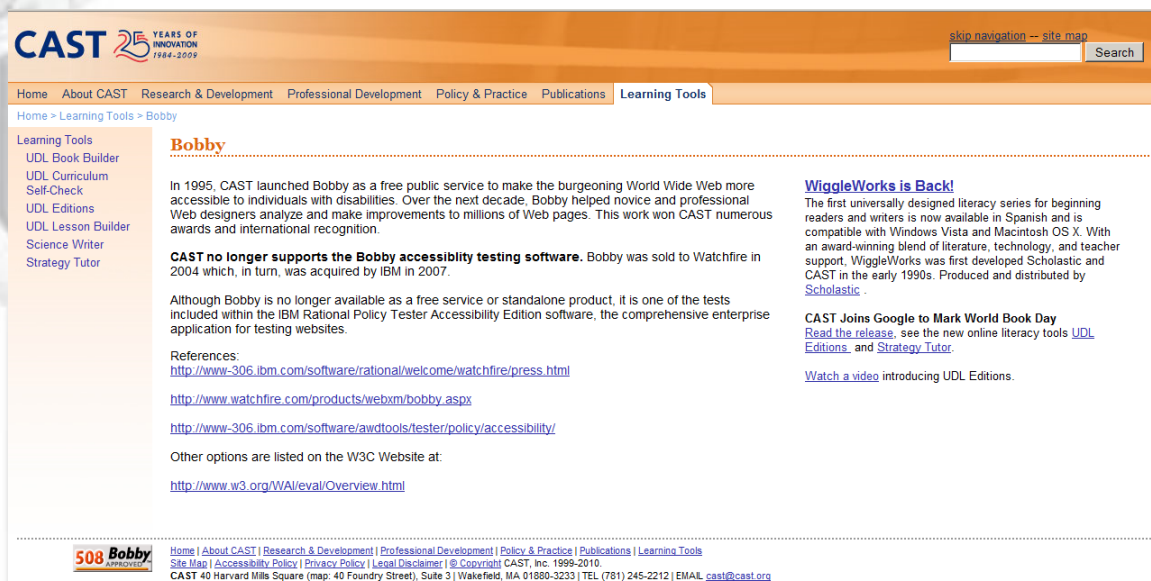


Figura 4. CAST (Center for Applied Special Technology)  
(<http://www.cast.org/learningtools/Bobby/index.html>)

## 1.3 Fases del desarrollo

El trabajo presentado ha sido dividido en tres fases principales, atendiendo a las fases de desarrollo de los sistemas de software del mercado: fase de planificación, en la que atendemos a la planificación y documentación del proyecto como cimientos para un buen desarrollo; fase de construcción, subdividida a su vez en análisis y diseño inicial, y en diseño de interfaz, atendiendo en el primer diseño a un estudio más analítico, en el segundo diseño a generar una interfaz amigable y profesional; y la fase de implementación, referido como desarrollo técnico, dónde se describe el código utilizado y las pruebas realizadas.

### 1.3.1 Planificación y Documentación

Durante las primeras reuniones en las que se definieron los objetivos principales de este proyecto, comenzaron a gestarse unas ideas básicas de la estructura del prototipo y su diseño, sin embargo, este objetivo no llegó a materializarse hasta no haber completado una buena planificación y una buena recopilación de documentos que apoyasen el proyecto.

Comenzamos la planificación por reunir la documentación necesaria para dar forma a las ideas y apoyar sobre una base sólida la estructura y el diseño. Desde congresos sobre Procesamiento de Lenguaje Natural y Tecnologías del Habla, pasando por una cuantiosa colección de artículos en inglés sobre X+V en la plataforma de Opera, hasta una variedad de manuales sobre los lenguajes y foros de profesionales en la materia. Continuamos con la estructuración y redacción de la memoria del proyecto para dejar constancia del desarrollo en el momento de realizarlo. Definimos unos objetivos claros y concisos que



favorecen en gran medida un punto al que mirar para no perderse en el camino del desarrollo, concretando el mensaje a transmitir, el público al que se dirige, y el contenido principal del proyecto. Marcando en el horizonte un plazo de tiempo realista no superior a seis meses.

### 1.3.2 Análisis y Diseño Inicial

Nos introducimos en esta fase de desarrollo de la aplicación realizando un primer análisis del entorno, visualizando los objetivos principales y acometiéndolos para generar un diseño acorde. Para ello, nos centramos en dos diseños básicos: el diseño de las bases de datos necesarias para el buen funcionamiento de las aplicaciones, debido al gran volumen de datos que se prevé van a utilizar; y el diseño de la estructura del sitio Web, definiendo el aspecto y distribución de los elementos principales que la compondrán.

#### *Diseño de Bases de Datos*

Algunas de las aplicaciones, por su abundante información han obligado a la creación de bases de datos, como es el caso de las aplicaciones de *Voice Library*, *Voice Videoclub*, y *Voice Pronunciation*, ésta última por partida doble, para el juego de las palabras y para el de las imágenes.

Debido a la simpleza de las bases de datos no ha sido necesario el diseño del modelo entidad/relación, ni del modelo relacional, pues se ha generado simplemente, en el caso de los libros por ejemplo, una tupla por libro con su correspondiente información. En cambio, sí ha sido necesaria la creación de un sistema de gestión de estas bases de datos camuflado en las aplicaciones. En cada una de las aplicaciones con acceso a una base de datos hay un botón que muestra un sistema de acceso por nombre de usuario y contraseña que permite entrar a un menú. A través de este menú podemos crear, modificar o borrar los objetos de las

bases de datos. Esta zona oculta al usuario ha sido descartada de la integración de interfaz de voz debido al uso único y exclusivo del desarrollador del entorno.

En las Figuras 5, 6, 7 y 8 podemos observar la descripción detallada mostrada por la consola de comandos de MySQL, de las tablas utilizadas en las bases de datos principales de la aplicación.

```

mysql
phpmyadmin
pronunciation
test
videoclub
-----+
9 rows in set (0.66 sec)

mysql> use library
Database changed
mysql> describe book;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| isbn       | varchar(20)   | NO   | PRI |          |       |
| title      | varchar(60)   | YES  |     | NULL    |       |
| author     | varchar(25)   | YES  |     | NULL    |       |
| editorial  | varchar(50)   | YES  |     | NULL    |       |
| description| varchar(500)  | YES  |     | NULL    |       |
| edition    | varchar(10)   | YES  |     | NULL    |       |
| link       | varchar(200)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.23 sec)

mysql>
    
```

Figura 5. Base de datos de aplicación Voice Library – tabla books

```

Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.51b-community-nt-log MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> USE videoclub;
Database changed
mysql> DESCRIBE films;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| original_title | varchar(100)  | NO   | PRI |          |       |
| translated_title| varchar(100)  | YES  |     | NULL    |       |
| year           | varchar(5)    | YES  |     | NULL    |       |
| director       | varchar(100)  | YES  |     | NULL    |       |
| languages      | varchar(100)  | YES  |     | NULL    |       |
| category       | varchar(100)  | YES  |     | NULL    |       |
| description    | varchar(3000) | YES  |     | NULL    |       |
| picture        | varchar(50)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.13 sec)

mysql>
    
```

Figura 6. Base de datos de aplicación Voice Videoclub – tabla films

```

mysql> use pronunciation;
Database changed
mysql> describe words;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| number     | varchar(20)   | NO   | PRI |          |       |
| word       | varchar(20)   | YES  |     | NULL    |       |
| description | varchar(1000) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.10 sec)

mysql>
    
```

Figura 7. Base de datos de aplicación Voice Pronunciation – tabla words

```

mysql> use pronunciation;
Database changed
mysql> describe picture;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| number     | varchar(10)   | NO   | PRI |          |       |
| name       | varchar(20)   | YES  |     | NULL    |       |
| type       | varchar(30)   | YES  |     | NULL    |       |
| description | varchar(50)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.06 sec)

mysql>
    
```

Figura 8. Base de datos de aplicación Voice Pronunciation – tabla picture

### Diseño de la estructura del sitio Web

Como vimos en la Figura 2, el entorno Web está distribuido de tal manera que el usuario al acceder a la página principal *index.xhtml* recibe una bienvenida a modo de introducción oral y una presentación visual agradable y acorde con el entorno. Se le muestra una breve descripción de todas y cada una de las aplicaciones de las que se compone el entorno, de tal manera que si pasa el ratón

por encima de cualquiera de los textos, éstos son leídos al usuario por el motor del diálogo.

El usuario dice el nombre de la aplicación que quiere visitar, o hace click sobre ella y accede a dicha aplicación. Por ejemplo, la aplicación *Voice Dictionary* nos muestra una barra de búsqueda y unas breves instrucciones. El usuario puede insertar el contenido de la búsqueda o seguir navegando por el entorno. Para buscar algo debe deletrearlo haciendo corresponder la letra a insertar con el país que se proporciona en las instrucciones. Esta manera de introducir los datos resulta menos natural de lo habitual, pero eliminamos la limitación de una gramática restringida. Una vez introducido el entorno envía la información a *Wikipedia* y recoge la información resultante, procediendo a procesarla y a mostrarla en el entorno.

Cada una de las aplicaciones cumple un fin específico y diferente del resto, de manera que se completa una amplia gama de aplicaciones de acceso multimodal. Sin embargo la estructura es siempre parecida:

- Presentación y búsqueda.
- Motor de la búsqueda.
- Captura de resultados.
- Presentación de los resultados.


A excepción de la aplicación *Voice Pronunciation*, en la que ayudamos a la mejora en aspectos de pronunciación mediante divertidos juegos. En este caso, en ambos juegos, la estructura se presenta de la siguiente manera:

- Presentación y elección del juego.
- Objeto a pronunciar y descripción (Repetición Aleatoria).
- Fin del juego y vuelta a presentación inicial (Usuario pulsa/dice “Exit”).

### 1.3.3 Diseño de Interfaz

En esta segunda descripción del diseño, nos centramos en desarrollar el prototipo imaginario y el prototipo real. En el prototipo imaginario se busca fundamentalmente la descripción de los detalles de visualización, mientras que en el prototipo real se busca una descripción más detallada de la estructura empleada.

#### *Prototipo imaginario*



Con una visualización del contenido y la estructura a desarrollar, la implementación de las interfaces, visual y oral, fue una tarea de diseño centrada en darle un aspecto a la vez profesional y cercano al usuario, que favoreciese el acercamiento e invitase a su utilización. Realizando un desarrollo de la plantilla CSS [\[1.3.3-A\]](#) centrada en unos colores específicos, y acordes con el entorno, únicamente diferenciando con detalles mínimos las aplicaciones entre sí, con unas fuentes y estilos desarrollados para no destacar y aparecer agradables a la vista, conseguimos ese objetivo de interfaz visual buscado. Para desarrollar la interfaz oral fue más necesario un estudio de diálogo, con presentaciones, descripciones e instrucciones acordes con el desarrollo del entorno, mostrando esa profesionalidad y acercamiento al usuario buscado.

#### *Prototipo real*

Este estilo se presenta con un contenido centrado en la página, con márgenes sombreados que lo delimitan y fondo blanco. En la parte superior se encuentra la imagen representativa de la aplicación, lo podríamos llamar la imagen corporativa.

Bajo esta imagen encontramos el menú de cinco aplicaciones, en tonos naranjas y blancos. Situar el ratón encima de alguno de los botones que componen el menú hace aparecer una imagen de fondo dándole un toque de dinamismo, y acceder a alguna de estas aplicaciones muestra una imagen sobre ese botón concreto que nos informa del lugar en el que nos encontramos del entorno.

Tras el menú tenemos el cuerpo de la página, el contenido. Este contenido se muestra con letras de color negro, tamaños en función de la representación (títulos, subtítulo, lista, texto, anotación...), y fuentes comprendidas en la familia Verdana, Arial, Helvetica, y Sans-Serif. Las barras de búsqueda mostradas en el contenido se ajustan a la gama de estilo con sombras de azules claros.

Al final de la página encontramos el pie de página, que contiene una imagen del W3C de comprobación de hoja de estilos correcta, y los enlaces a páginas tales como políticas de privacidad, información legal, información de contacto y copyright. Estas Webs han sido cuidadas tanto en su forma y contenido como en su estilo acorde con el entorno.

### **1.3.4 Desarrollo Técnico**

Centrados ya en la fase de construcción, a la que hemos llamado desarrollo técnico, entramos en la descripción del desarrollo funcional, donde se muestra la implicación de cada uno de los lenguajes de programación utilizados en el proyecto. A continuación, se describen las pruebas de evaluación realizadas a rasgos generales, tanto pruebas unitarias, como pruebas de integración y de sistema.

## ***Desarrollo funcional***

Cada una de las páginas diseñadas ha llevado un proceso similar, en el que se creaba la estructura principal que iba a contener la información y las etiquetas de desarrollo de interfaz oral. Esta estructura principal se ha realizado en XHTML acorde con el estándar X+V. Se ha programado una única plantilla CSS para albergar los estilos del entorno, una única estructura en JavaScript para crear las funciones que daban acceso a objetos y procesos calculados, y favorecer un entorno más desarrollado y dinámico. La estructura XHTML también alberga el subconjunto de VoiceXML necesario para desarrollar un completo motor del diálogo e interfaz oral.

Los accesos a páginas externas al entorno, recogida de información de otras Webs, y procesos más desarrollados los han llevado a cabo páginas independientes pertenecientes al entorno, programadas en PHP [\[1.3.4-A\]](#) sin interfaz de usuario. Las páginas de proceso en PHP favorecen el acceso a las bases de datos programadas en MySQL [\[1.3.4-B\]](#) y la creación a su vez de las páginas de resultado desarrolladas en X+V, debido a que el subconjunto de VoiceXML no permite interfaces orales en Webs montadas directamente en PHP.

## ***Pruebas de evaluación***

En esta etapa del desarrollo de la aplicación se ha tenido especial consideración en las pruebas unitarias, pruebas de integración, y pruebas de sistema realizadas tanto en los accesos a Webs externas como a los accesos a bases de datos. Sin embargo, el grueso de las pruebas de evaluación realizadas a la aplicación se ha centrado en la interfaz oral del entorno. Pruebas de navegabilidad y reconocimiento vocal, de inserción de caracteres y búsqueda de contenidos. Al realizar las pruebas orales se vio la posibilidad de encontrarse con usuarios cuya pronunciación fuese más irregular, por lo que se decidió realizar la aplicación *Voice Pronunciation* que favorece el desarrollo y perfeccionamiento de la pronunciación de inglés con sencillos y divertidos juegos.

## 1.4 Medios empleados

La sección Medios Empleados nos aporta la información necesaria sobre los lenguajes de programación, los programas, y los manuales necesarios para desarrollar el proyecto. De esta forma abarcamos todos los recursos que han sido empleados para la creación de la aplicación.

### 1.4.1 Lenguajes de Programación

El proyecto ha estado dirigido por la batuta del lenguaje X+V, marcando los tiempos en forma de multimodalidad, y guiando la estructura desde las etiquetas hasta los mínimos detalles de interfaz. Sin embargo, este proyecto no habría sido desarrollado sin el uso de JavaScript, plantillas CSS, y en su gran mayoría PHP. Este último lenguaje ha sido el principal motor de proceso de todas y cada una de las aplicaciones, permitiendo las consultas y accesos a las bases de datos, realizando consultas a páginas externas al entorno, y recogiendo la información necesaria para después procesarla y generar el código de la página de respuesta, teniendo así una página de resultado siempre vacía a espera de que el proceso correspondiente de PHP registre el contenido de las búsquedas de los usuarios y quede en perfectas condiciones para su visualización/escucha/narración.

También ha sido muy relevante el uso de MySQL para la creación y gestión de las bases de datos necesarias para la inmensa cantidad de datos de los que hacen uso determinadas aplicaciones. De esta forma los lenguajes utilizados han sido X+V (XHTML, VoiceXML, XML Events), CSS, JavaScript, PHP y MySQL.



## 1.4.2 Programas empleados

El principal programa utilizado es el único navegador que tiene actualmente la capacidad de diálogo oral implementada y que permite el uso de interfaces orales en el desarrollo de entornos Web, me refiero al navegador y suite de Internet Opera. Su motor de reconocimiento de voz permite la ejecución de páginas desarrolladas para entradas y salidas multimodales.

Para generar todo el código, de cada uno de los lenguajes utilizados en este proyecto se han usado los editores de texto con formatos de programación: [PSPad Editor](#), y [Notepad++](#). Ambos editores permiten un manejo más creativo y útil del código, dejando de lado las herramientas de construcción y edición Web WYSIWYG [\[1.4.2-A\]](#) y evitando el diseño a golpe de ratón debido al exceso de código inútil generado por muchos de ellos.

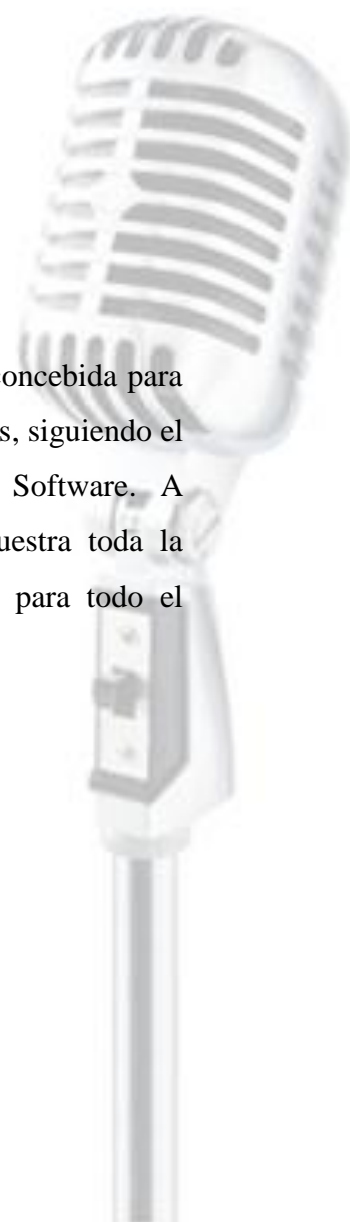
El paquete Appserv [\[1.4.2-B\]](#), que incluye entre otras herramientas el servidor gratuito Apache [\[1.4.2-C\]](#), y el gestor MySQL, han contribuido de igual manera para el desarrollo de las aplicaciones. Apache es un servidor Web multiplataforma de código abierto escrito en C que, aun careciendo de interfaz visual que lleve a una fácil configuración, ha permitido la ejecución de los script de PHP de este proyecto, y ha favorecido el rápido avance en el desarrollo y aprendizaje de este lenguaje. MySQL, siendo un sistema gestor de bases de datos relacionales, multihilo y multiusuario desarrollado en código libre, es el motor de desarrollo de todas y cada una de las bases de datos usadas por las aplicaciones diseñadas, favoreciendo su gestión a través de procesos privados del entorno Web.

### 1.4.3 Manuales

La documentación examinada consta de artículos extraídos de entornos dedicados a la programación X+V o entornos de interacción multimodal, de congresos y libros consultados sobre lenguaje de computación oral e interfaces de diálogo. Sin embargo, los manuales principalmente utilizados han sido *XHTML+Voice Programmer's Guide* en su versión 1.0 de IBM, y el manual ofrecido por el W3C para programación VoiceXML.

## 1.5 Planificación Temporal

El desarrollo de todo el proyecto ha ido siguiendo una planificación concebida para depurar y optimizar al máximo el tiempo empleado en cada una de las tareas, siguiendo el Método Larman [\[1.5-A\]](#) estudiado en la asignatura Ingeniería del Software. A continuación se presenta el diagrama de Gantt en la Figura 8 que muestra toda la planificación temporal del proyecto, haciendo una estimación temporal para todo el desarrollo del proyecto de 170 días:



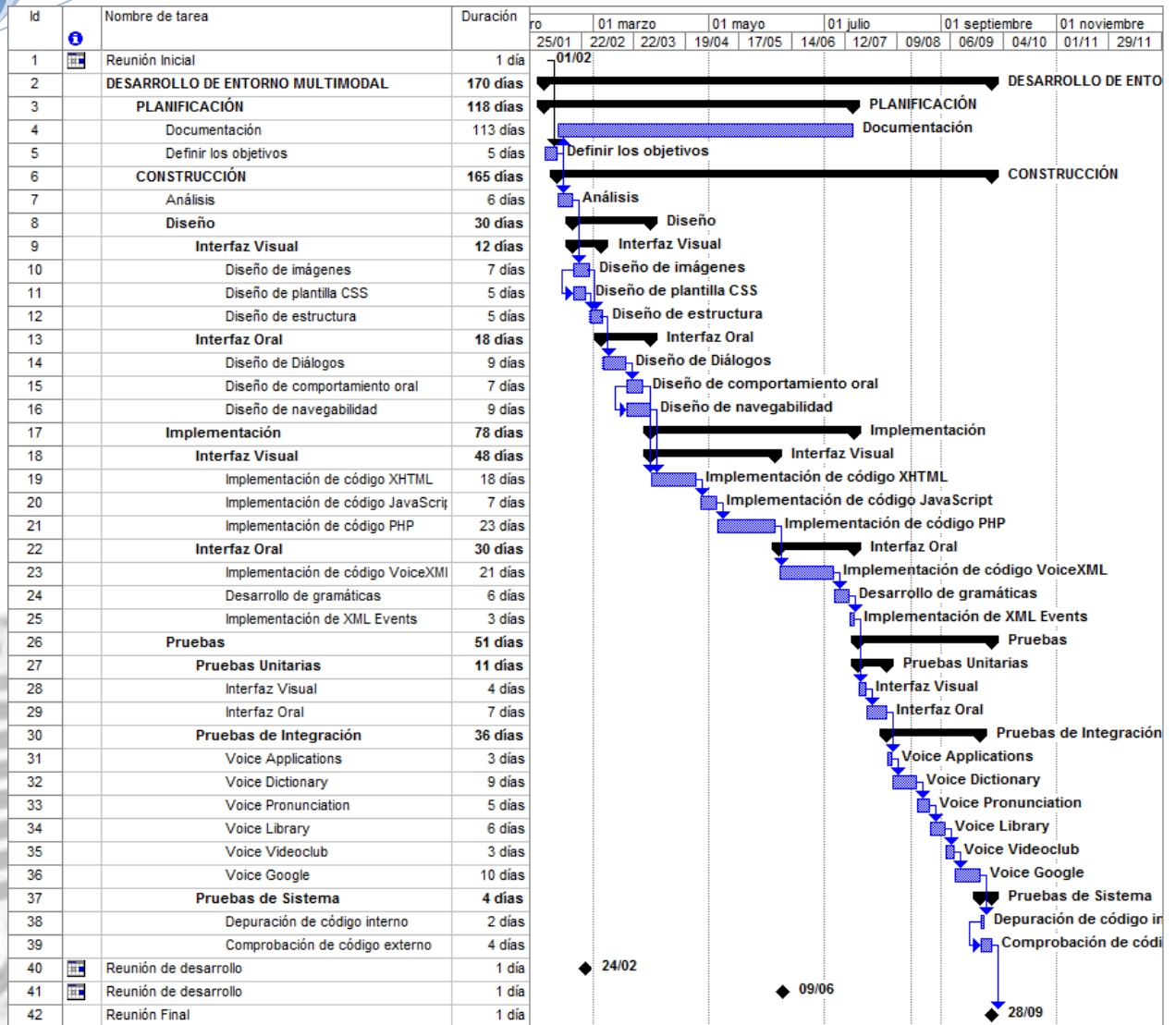


Figura 9. Diagrama de Gantt: planificación temporal del proyecto

## 1.6 Estructura de la memoria

Para facilitar la lectura de la memoria, se incluye a continuación un breve resumen de cada capítulo:

**Capítulo 1: Introducción.** Con la lectura de este capítulo entramos en la concepción de la idea general que rodea a la creación de un entorno de usuario multimodal. Se introduce al lector en las necesidades creadas por una era tecnológica cada vez más desarrollada, favoreciendo la integración del usuario y la interacción a través de diversos

modos humanos con las máquinas, de manera que cualquier persona pueda acceder a los recursos ofrecidos por la tecnología actual y futura. Se presentan todos los objetivos perseguidos por el presente proyecto, medios necesarios, y planificación temporal. Además de una amplia descripción de las fases de desarrollos por las que ha ido pasando el entorno.

**Capítulo 2: Estado del Arte.** Esta sección de la memoria del proyecto se centra en el estado actual de los Sistemas de Diálogo Orales y la tecnología X+V, además de la evolución histórica de los sistemas de diálogo, sus estándares y aplicaciones.

**Capítulo 3: Entorno X+V: *Voice Applications*.** Descripción detallada de todas y cada una de las aplicaciones, presentando las aplicaciones y su funcionamiento, la arquitectura que rodea a las aplicaciones, y las pruebas y limitación de errores que generan la robustez que caracteriza al entorno.

**Capítulo 4: Presupuesto.** Incluye un pequeño resumen del proyecto, una división en fases y subfases con el correspondiente diagrama de Gantt y un desglose de costes de personal, costes del material y costes totales.

**Capítulo 5: Conclusiones.** Conjunto de experiencias y conclusiones obtenidas a lo largo de todo el desarrollo del proyecto, además de apuntes sobre las posibles líneas de investigación acometidas en el futuro.

Tras estos capítulos podemos encontrar la información detallada sobre el Glosario de términos, Referencias y Bibliografía, y los anexos correspondientes para completar las especificaciones.

# Capítulo 2

## *Estado del Arte*

### 2.1 Introducción

A lo largo del presente capítulo se detalla el resultado obtenido del proceso de investigación sobre la tecnología utilizada durante la creación de la aplicación y entorno multimodal. La base principal de este capítulo está apoyada en el excelente trabajo de investigación realizado por Zoraida Callejas sobre los Sistemas de Diálogo Orales [\[2.1-A\]](#), aportando un amplio resumen completado con toda la información necesaria para desarrollarlo paralelamente al proyecto realizado. Deteniéndonos en primer lugar en los Sistemas de Diálogo Oral y su estructura, completándolo con la descripción detallada de cada uno de los módulos que componen la arquitectura de los Sistemas de Diálogo Orales. Posteriormente se presenta una línea temporal con la historia de la tecnología que nos ocupa, y un análisis detallado de los estándares principales de este proyecto.

## 2.2 Sistemas de Diálogo Oral

El reconocimiento oral tiene una larga trayectoria en la historia, debido principalmente a la gran dificultad que arrastra dentro de la ciencia de la computación y la inteligencia artificial. En esta ardua tarea se produjo el nacimiento de lo que hoy llamamos Sistemas de Diálogo Orales (SDOs), definidos por los expertos como software que acepta lenguaje natural como entrada y produce una salida también oral en lenguaje natural, estableciendo de esta manera una conversación con el usuario. Esta interacción se realiza gracias a todos los procesos y módulos que componen la arquitectura de los SDOs, como podemos ver en la Figura 10. Estos procesos son el Reconocimiento Automático del Habla, Procesamiento del Lenguaje Natural, Gestión del Diálogo, Generación del Lenguaje Natural y Conversión Texto-Habla. Cada una de estas tareas representa un módulo distinto en la arquitectura clásica de estos sistemas.

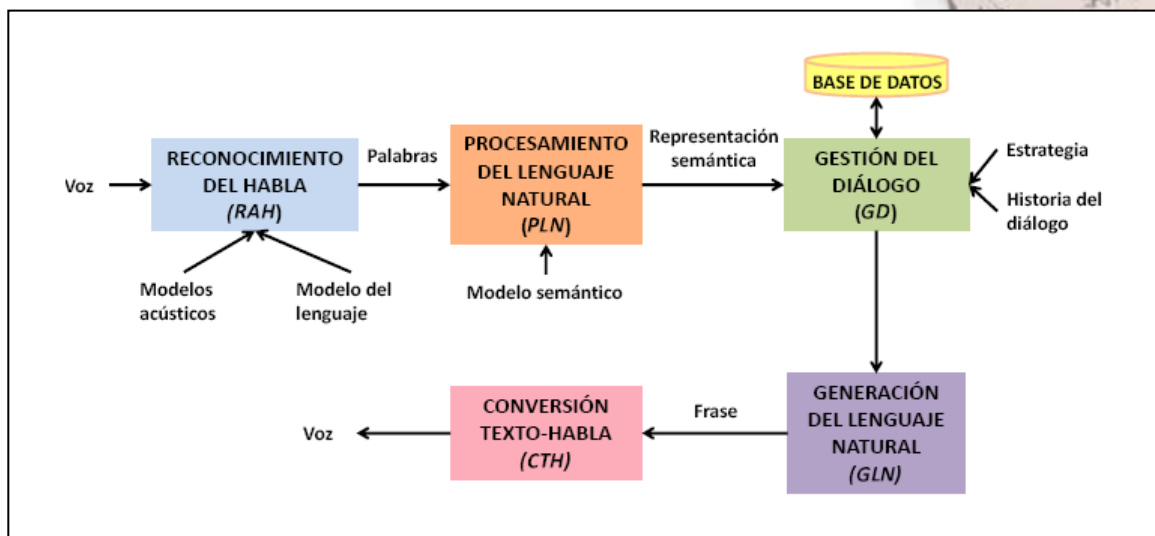


Figura 10. Arquitectura Modular de los Sistemas de Diálogo Oral

[\[2.1-A\]](#)

### 2.2.1 Reconocimiento Automático del Habla

Podemos datar los primeros los Reconocedores Automáticos del Habla (RAH) junto a los primeros espectrógrafos debido al tratamiento de la señal acústica y su similitud. El RAH es el proceso encargado de obtener el texto escrito correspondiente a una entrada oral pronunciada por el usuario. Este sistema se compone de una etapa de preproceso (modelado de la señal), una etapa Acústico-Fonética (modelado acústico de unidades subléxicas y/o léxicas), y una etapa Sintáctico-Semántica (modelado del lenguaje). Las dos últimas etapas pueden combinarse de forma secuencial o integrarse en único módulo.

El objetivo principal del RAH es combinar un conjunto de informaciones que provienen de diversas fuentes del conocimiento (Acústica, Fonética, Fonológica, Léxica, Sintáctica, Semántica y Pragmática). Este conjunto de informaciones puede estar incompleto o hallarse en presencia de ambigüedades, incertidumbres y errores inevitables. El RAH obtiene una interpretación aceptable del mensaje acústico recibido del usuario, principalmente obtiene una serie de patrones a partir de la aplicación de técnicas de proceso de señal de voz, cribando las frases de texto más probables asignadas a una señal acústica pronunciada por el usuario.

Debido a su complejidad, en el desarrollo de este módulo intervienen diversas disciplinas (Fisiología, Acústica, Procesamiento de Señales, Inteligencia Artificial, y Ciencia de la Computación) en trabajo conjunto para llegar a minimizar todos los errores producidos de las diferencias lingüísticas y de pronunciación entre diferentes locutores. Este tipo de variabilidad, incluso en el mismo locutor, se produce frecuentemente debido a factores fisiológicos y psicológicos, convirtiendo los errores de comunicación en impredecibles. La forma de hablar, tono de voz, edad, sexo o nacionalidad intervienen directamente en la variabilidad. Los expertos se han visto obligados a someter al sistema a una robustez tal que consiga eliminar errores procedentes, no sólo de la variabilidad antes comentada, sino también de los canales o dispositivos de comunicación y ambientales.

## 2.2.2 Procesamiento del Lenguaje Natural

En el momento en el que el sistema de diálogo reconoce la información pronunciada por el usuario, se necesita comprender su significado. El Procesamiento del Lenguaje Natural (PLN) se ocupa de proveer métodos y técnicas que automáticamente faciliten la comunicación entre los usuarios y las máquinas por medio de lenguajes naturales. La formulación e investigación para que estos mecanismos sean eficaces computacionalmente (que se puedan realizar por medio de programas que ejecuten o simulen la comunicación) permite obtener la semántica asociada a una cadena de texto.

El PLN se realiza mediante los siguientes análisis fundamentales: un primer análisis morfológico-léxico; un posterior análisis sintáctico; análisis semántico; y análisis pragmático. El conocimiento del sistema sobre el aspecto léxico y morfológico permite dividir las palabras en lexemas (parte de las palabras que indica su semántica) y en morfemas (diferentes infijos y sufijos mediante los cuales se obtienen las distintas clases de palabras). Mediante el análisis sintáctico se obtiene la estructura jerárquica de las frases, y se extrae el significado de la estructura sintáctica compleja obtenida a partir del significado de sus constituyentes. A través del análisis semántico se obtiene un procesamiento del significado, sin embargo las técnicas en representación de significados no han obtenido los resultados deseados. El análisis pragmático añade información al análisis semántico de la frase en función del contexto en el que aparece.

Además, se pueden añadir otros niveles del conocimiento como es la información fonológica (relación de las palabras y su sonido al pronunciarlas), análisis del discurso (estudio de información previa que puede interferir en el significado de la información actual), y lo que se denomina como “conocimiento del mundo” (información esencial sobre la estructura del mundo necesaria para mantener una conversación).



### 2.2.3 Gestión del Diálogo

Debido a que no existe una definición global de las tareas a realizar por el módulo de Gestión del Diálogo, nos apoyamos en los conceptos aportados por Traum y Larsson [\[2.1.3-A\]](#). Según estos autores, un Gestor del Diálogo debe cumplir las funciones siguientes:

- i. Actualizar el contexto del diálogo.
- ii. Proveer de contexto en el que basar las interpretaciones.
- iii. Coordinar el resto de módulos del sistema.
- iv. Decidir qué información dar al usuario, y cuándo hacerlo.

De esta forma entendemos que el Gestor del Diálogo debe tratar con distintas fuentes de información tales como el resultado de procesar el lenguaje natural, de consultar las bases de datos, y conocimientos acerca del dominio de la aplicación, las intervenciones de los usuarios y la historia previa del diálogo. A partir de todas estas fuentes de información, el gestor del diálogo selecciona la próxima acción que llevará a cabo el sistema. De este modo, se puede considerar al Gestor del Diálogo como el “cerebro” del sistema de diálogo.

### 2.2.4 Generación del Lenguaje Natural

A través del proceso de Generación del Lenguaje Natural (GLN) obtenemos la construcción de expresiones o partes de ellas de cualquier tamaño tanto habladas como escritas en lenguaje natural a partir de una representación no lingüística. Así podemos decir, que este modulo trata de traducir la acción seleccionada por el gestor de diálogo en una frase en lenguaje natural que será comunicada al usuario.

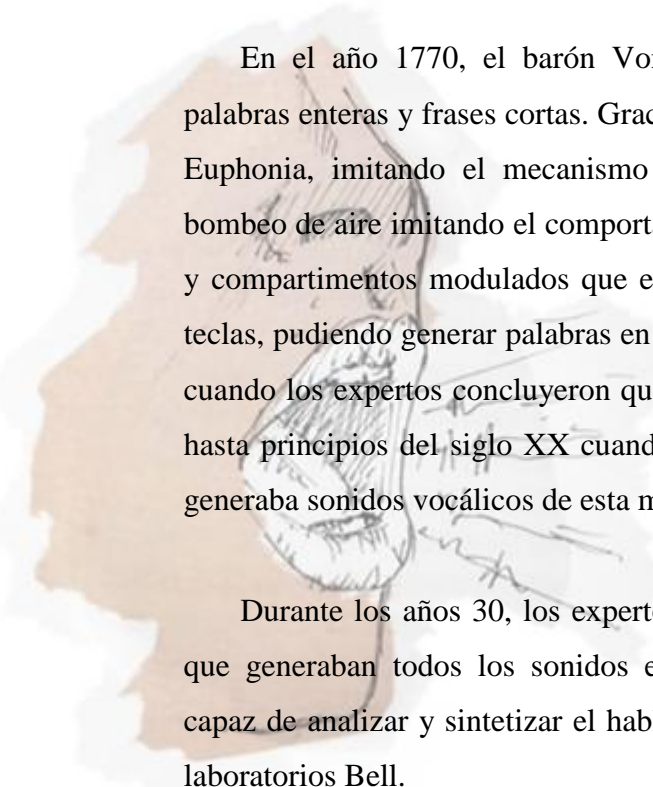
El sistema debe escoger la información de las bases del conocimiento, decidir cómo organizarla, y determinar cómo producir el texto en lenguaje natural, incluyendo la decisión de la entrada léxica y la estructura sintáctica. Este proceso se lleva a cabo mediante la organización del contenido, distribución del contenido en fases, la lexicalización del contenido, generación de expresiones referenciales y realización lingüística. Es importante optimizar el texto para obtener mensajes legibles, adaptando el vocabulario y la complejidad de las estructuras sintácticas a la destreza lingüística del usuario.

### 2.2.5 Síntesis de Texto y Voz

Un sistema de síntesis oral se encarga de transformar un texto escrito en una señal acústica. Los sintetizadores de texto a voz se componen de dos partes fundamentales, una encargada de convertir el texto plano, compuesto por símbolos y abreviaturas, en sus palabras asociadas, o lo que es lo mismo, se encargada de normalizar el texto y de asignar transcripciones fonéticas a las palabras dividiendo y marcando el texto en frases, cláusulas y oraciones; y la otra es la encargada de convertir la representación lingüística simbólica en sonidos, utilizando diversas técnicas para simular los sonidos producidos por las cuerdas vocales humanas. La primera parte de los sintetizadores de texto a voz se llama “*front-end*”, y la segunda parte se llama “*back-end*” [\[2.1-A\]](#).

## 2.3 Historia de los Sistema de Diálogo

Desde la construcción y desarrollo de las ideas que el hombre tenía de lo que podrían ser las máquinas, siempre se ha deseado establecer una comunicación con ellas. Los primeros diseños serios que apoyan esta comunicación se remontan a los siglos XVIII y XIX en los que podemos encontrar los primeros autómatas que imitan la conducta humana. Se trataba de máquinas imitando a pequeños animales o muñecos que podían producir sonidos. Estos primeros ejemplares de máquinas capaces de emitir sonidos fueron construidas por los maestros relojeros de la época con los materiales y conocimientos de que disponían.



En el año 1770, el barón Von Kempelen desarrolló un autómata que generaba palabras enteras y frases cortas. Gracias a esto, Josef Faber construyó en 1857 la máquina Euphonia, imitando el mecanismo de producción del habla mediante un fuelle con bombeo de aire imitando el comportamiento de un pulmón humano, y una serie de placas y compartimentos modulados que empleaban un teclado parecido al de un piano de 16 teclas, pudiendo generar palabras en varios idiomas europeos. Fue a finales del siglo XIX cuando los expertos concluyeron que la voz podía ser generada eléctricamente, y no fue hasta principios del siglo XX cuando J. Q. Stewart [\[2.2-A\]](#) construyó una máquina que generaba sonidos vocálicos de esta manera.

Durante los años 30, los expertos se volcaron en el diseño y desarrollo de sistemas que generaban todos los sonidos eléctricamente, como fue VOCODER, un proyecto capaz de analizar y sintetizar el habla humana mediante un teclado, desarrollado por los laboratorios Bell.

En los años 40 aparecen las primeras computadoras, y con ellas aparece el “Test de Turing” [\[2.2-B\]](#) de las manos de Allan Turing. Este test medía la capacidad de inteligencia de las máquinas mediante un juez humano que entabla una conversación en lenguaje natural con la máquina y debe decidir si ha hablado con una máquina o con otra persona. En caso de que la persona no fuese capaz de diferenciarlo, la máquina pasa la prueba. Este punto de partida dio origen a la investigación que en los años 60 propició los primeros agentes conversacionales, como ELIZA [\[2.2-C\]](#), además de dar origen a la idea principal de la novela de Philip K. Dick “¿Sueñan los androides con ovejas eléctricas?” escrita en 1968, en la que un investigador tiene que averiguar si la persona que tiene delante es una máquina o una persona a través de una serie de preguntas y problemas de situación (novela en la que se inspiró Ridley Scott en 1982 para dar forma a la película *Blade Runner*). Ciencia ficción aparte, el agente conversacional ELIZA es un sistema que utiliza la localización de palabras clave para completar una serie de plantillas predefinidas. Aunque ELIZA superaba el Test de Turing, no interpretaba semánticamente la entrada de los usuarios.

La investigación en lingüística computacional realizada durante los años 70 surgió para tratar el reto de poder realizar estas interpretaciones semánticas partiendo de los trabajos teóricos desarrollados desde los 50 por Chomsky, Montague y Wood. En esta década aparecen los primeros sintetizadores de voz basados en reglas, además de los primeros reconocedores continuos del habla, creados después de décadas de investigación sobre las conversaciones y diálogos en los que se mantienen pausas largas.

La evolución tecnológica y los progresos obtenidos en el campo del reconocimiento oral favorecen las primeras iniciativas de investigación a principios de los años 80. Por un lado tenemos el programa DARPA (*Defense Advanced Research Projects Agency*) Spoken Language Systems [\[2.2-D\]](#) en los Estados Unidos, dirigido al estudio y desarrollo de tecnologías relacionadas con el reconocimiento automático del habla. El procesamiento del lenguaje natural fue llevado a cabo a través de *Air Travel Information Services* (ATIS), sistema enfocado a la reserva de vuelos vía telefónica.

Por otro lado tenemos el proyecto Esprit SUNDIAL (*Speech UNderstanding and DIALog*) [2.2-K] en Europa, tratando la información relacionada con los horarios de avión o tren en cuatro idiomas europeos. La investigación realizada en SUNDIAL fue el origen de numerosos proyectos financiados por la Comunidad Europea relativos principalmente al modelado del diálogo. Según algunos autores [1.1-G] el proyecto DARPA ATIS pertenece a una generación previa de sistemas de diálogo orales debido a que está restringido a un dominio cerrado.

Durante la década de los 90 se prestó mayor atención a la mejora en número de éxito de los diversos módulos de los sistemas de diálogo. La mayor preocupación era la robustez, investigando la degradación del funcionamiento que experimentaban los sistemas debido a cambios de menor importancia como variaciones en los micrófonos o en los canales de comunicación. La tecnología de la época no era capaz de ofrecer soluciones aceptables.

DARPA Communicator destaca como uno de los programas de investigación, con participación Europea y Estadounidense, más importantes de los años 90. El proyecto albergaba la capacidad de interactuar con los usuarios a nivel multidominio, haciendo posible que el usuario y el sistema iniciasen una conversación, cambiaran de tema o se interrumpiesen. Este programa de investigación se centró en el desarrollo de nuevas tecnologías del habla, que a su vez pudiesen emplear como entrada no sólo la voz sino también otras modalidades.

A finales de los 90, los expertos tuvieron que evolucionar para hacer frente a las diferencias existentes entre el canal telefónico fijo y los novedosos y cada vez más populares teléfonos móviles. Tales diferencias, como anchos de banda más estrechos y bajas relaciones señal-ruido, además de una mayor variedad de entornos desde los que los usuarios podían interactuar con los sistemas que exigían una mayor robustez para poder gestionar la comunicación en ambientes muy ruidosos [2.2-E], traían de cabeza a los investigadores.

Los expertos en el campo del PLN llevaron sus líneas de investigación durante los años 90 entorno a la generación de vocabularios más ricos facilitando desde el reconocimiento de palabras hasta el habla espontánea. En relación con la Gestión de Diálogo, sus esfuerzos se orientaron hacia la consecución de diálogos menos restrictivos en los cuales los usuarios pudiesen llevar la iniciativa de la comunicación.

A finales de 1999, el W3C Voice Browser Working Group [\[2.2-L\]](#) presentó los primeros estudios de requisitos para navegadores web, definiendo de esta manera una serie de lenguajes estándar para el desarrollo de Sistemas de Diálogo Orales. Estos estudios sentaron las bases de lenguajes de etiquetas tales como VoiceXML, del cual hablaremos más adelante.

El desarrollo de subdiálogos de uso frecuente favoreció la obtención de componentes más portables y reutilizables. Hasta finales de la década de los 90 no aparecieron las primeras arquitecturas que permitían el desarrollo y uso de dispositivos cuya configuración de controlador se encontraba integrada en el sistema operativo (los componentes conocidos como “*plug-and-play*” o “*enchufar-y-usar*”). Se desarrolla en esta década la adopción de aproximaciones estocásticas para obtener métodos no supervisados que aumentasen la capacidad de los módulos de RAH y PLN [\[2.2-F\]](#).

Las líneas de investigación futura apuntan hacia metas cada vez más complejas, con sistemas de diálogo cada vez más inteligentes, adaptables, dinámicos, portables y multimodales. Durante los 90 la tendencia principal se centraba en la robustez de los diversos módulos. Sin embargo, a partir del 2003 los expertos proponen objetivos de más alto nivel: sistemas de razonamiento avanzado, con capacidad de resolución de problemas, facultad de adaptación, proactividad, inteligencia afectiva, multimodalidad y multilingüismo.

Uno de estos objetivos de alto nivel son los sistemas proactivos. Estos sistemas (más sensibles al contexto y con mayor capacidad de resolución de problemas) influyen en la forma de ver a las máquinas, pasando de ser simples herramientas a convertirse en verdaderas entidades conversacionales. Un sistema proactivo puede entablar una conversación con el usuario incluso cuando este no haya solicitado explícitamente la intervención del sistema. El sistema se comporta de manera pasiva observando el diálogo entre usuarios humanos y capturando el contexto conversacional relevante, procesándolo para evaluar cuando intervenir.

Los investigadores no cesan en el empeño de obtener conversaciones naturales, tan ricas y dinámicas como las entabladas entre seres humanos. Esta investigación lleva al desarrollo de interfaces multimodales, dejando de lado el tradicional sistema de entrada a través del ratón y el teclado, o incluso SDOs unimodales (la voz como único modo humano). Se avanza en la investigación buscando combinar diferentes modalidades de entradas y/o salidas del sistema como la voz, las expresiones faciales o los gestos humanos.

A mediados de los años 90 aparecieron los primeros sistemas multimodales, que combinaban el habla con mapas gráficos [\[2.2-G\]](#) o con medios escritos [\[2.2-H\]](#). Las últimas investigaciones se han desarrollado entorno a la plena multimodalidad, dado que la mayoría de los sistemas sólo son multimodales en la entrada o la salida. El proyecto SMARTKOM (integra y sincroniza modalidades, y trata los fenómenos del diálogo asociados a la multimodalidad) proporciona lo que se ha denominado como “multimodalidad simétrica total” en un sistema de diálogo con iniciativa mixta [\[2.2-I\]](#).

La adaptabilidad, otro de los objetivos de alto nivel propuesto por los expertos, proporciona la ayuda al sistema para responder a diferentes necesidades en el momento en el que usuarios diferentes desean interacciones diferentes con el sistema. En las interacciones multimodales, persona-ordenador, los usuarios disponen de diversas formas de comunicación, lo que crea necesidades diferentes, a las cuales el sistema debe adaptarse, ya sean diferentes interfaces como diferentes lenguas. Un sistema de usos multilingües reconoce simultáneamente varios idiomas y comparte los modelos acústicos y/o de lenguaje.

Por último, la inteligencia emocional o afectiva integrada en los sistemas proporciona una aproximación para ajustar el comportamiento del diálogo al estado emocional del usuario. Este tipo de sistemas reconocen el estado emocional del usuario y ejecutan procesos en función de ese estado emocional para interactuar de diferente manera con el usuario.

Veamos ahora las diferentes formas de tratar a los sistemas más portables:

- ❖ Independencia del dominio, pudiendo trabajar en varios dominios o poder adaptarse a ellos.
- ❖ Independencia del Idioma, con sistemas multilingües que suministren la información en varios idiomas, y a su vez recibiendo las respuestas o entradas en distintos idiomas.
- ❖ Independencia de la Tecnología, con sistemas capaces de adaptarse a los diferentes dispositivos, arquitecturas o estructuras hardware.

Este último modelo de portabilidad puede entenderse desde el punto de vista del rápido desarrollo que sufre la tecnología en la época actual. Debido a que la potencia de los ordenadores continuará aumentando con costes cada vez más bajos de sus componentes, los sistemas portables tecnológicamente independientes deben estar desarrollados para poder adaptarse a ellos, como por ejemplo creando arquitecturas estándar. Una de estas arquitecturas es la propuesta por el MMI Working Group del W3C [\[2.2-J\]](#) cuyo objetivo principal es proporcionar un marco que asegure la interoperabilidad entre los componentes de diferentes fabricantes o de diferentes tecnologías.

## 2.4 Estándares

Los lenguajes de programación utilizados en este proyecto se ven sometidos a una serie de estándares recogidos por W3C en especificaciones útiles para desarrolladores. Dado que el estándar XHTML+Voice ha sido descrito ampliamente en el Capítulo 1, a continuación vamos a presentar el resto de los estándares haciendo destacar los más



importantes que han hecho posible la realización de este proyecto. Estos estándares han sido obtenidos de las páginas oficiales del W3C.

### 2.4.1 VoiceXML

El origen de VoiceXML se remonta a 1995, periodo en que los investigadores de la compañía AT&T trabajaban en una base para el desarrollo de diálogos y reconocimiento de voz. La empresa trabajó en este proyecto, denominado *Phone Markup Language* (PML), haciendo partícipes a otras empresas. Actualmente, junto a las empresas Lucent y Motorola, sigue trabajando en PML como lenguaje propio de diseño.

En 1998 el W3C organizó una conferencia sobre navegadores de voz. Para entonces AT&T y Lucent tenían ya varias versiones del original PML, mientras que Motorola investigaba en el desarrollo de VoxML, e IBM hacía lo propio con SpeechML. Muchos otros asistentes a la conferencia también estaban desarrollando lenguajes similares para estructuras de diálogo, como por ejemplo HP y TalkML.

El Foro VoiceXML fue formado por las compañías AT&T, IBM, Lucent y Motorola para aunar esfuerzos en la construcción de un lenguaje de diseño oral estándar que facilitase la labor de los desarrolladores a la hora de crear aplicaciones con comportamientos conversacionales. El Foro eligió XML como “placa de Petri” para este lenguaje debido a la firme inclinación de las compañías por unirse a la tendencia tecnológica contemporánea.

En el año 2000 el Foro VoiceXML lanza al público VoiceXML 1.0, el primer lenguaje de programación multimodal. Poco después, VoiceXML 1.0 se presentó a la W3C como base para la creación de un nuevo estándar internacional. La evolución a la versión VoiceXML 2.0 es el resultado del trabajo basado en las aportaciones de las empresas pertenecientes al W3C, otros grupos de trabajo del W3C, y del público.

VoiceXML está diseñado para crear diálogos de audio que incluyen voz sintetizada, audio digitalizado, reconocimiento del habla y *Dual Tone Multiple Frequency* (DTMF), registro de entrada oral, telefonía, e iniciativa mixta. Su principal objetivo es

llevar las ventajas de las aplicaciones desarrolladas para ser manejadas con la voz al entorno Web y sus contenidos interactivos.

En la Figura 11 podemos observar la representación gráfica de la arquitectura de VoiceXML. Un Servidor de Documentos (un servidor Web por ejemplo), procesa las solicitudes de una aplicación cliente (el Intérprete VoiceXML) a través de la Plataforma de Implementación. El Servidor de Documentos VoiceXML produce una serie de respuestas, procesadas posteriormente por el Intérprete. El Contexto VoiceXML puede supervisar las entradas de los usuarios en paralelo con el Intérprete. Por ejemplo, un Intérprete siempre puede escuchar una frase que lleve al usuario a un nivel personal de ayuda, y además puede escuchar frases que modifiquen las preferencias del usuario como el volumen o las características de la voz de la interfaz oral.

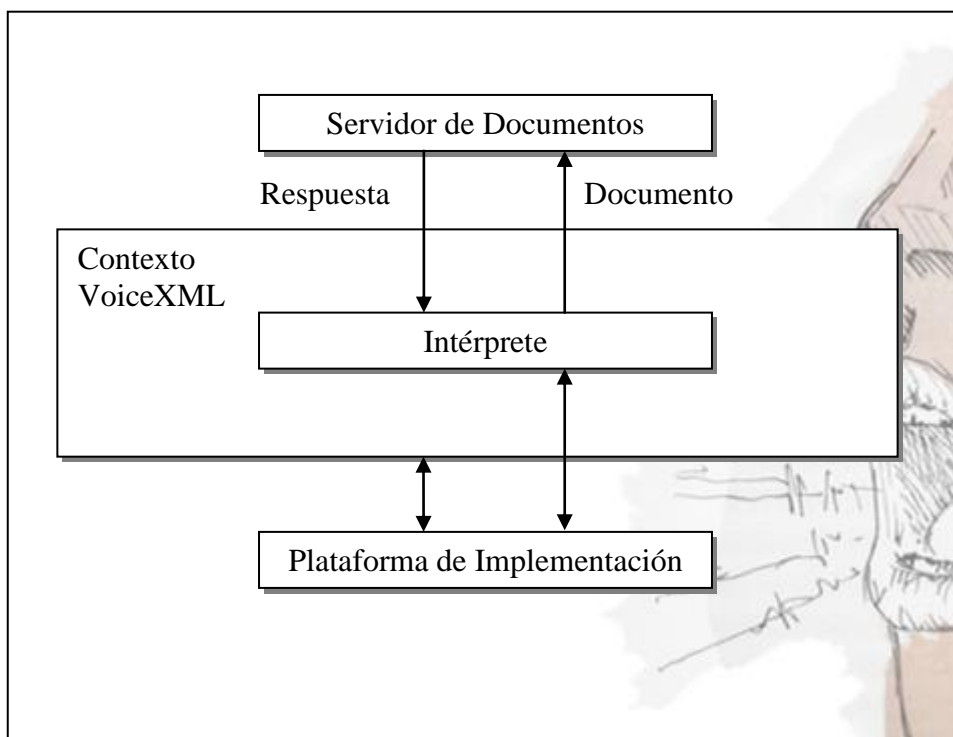


Figura 11. Arquitectura del modelo VoiceXML

**[2.4.1-A]**

La Plataforma de Implementación es controlada por el Contexto VoiceXML y por el Intérprete. Por ejemplo en una solicitud de respuesta de voz interactiva, el Contexto VoiceXML puede ser responsable de detectar una llamada entrante, la adquisición del documento inicial de VoiceXML, y contestar la llamada, mientras que el Intérprete lleva

a cabo el diálogo después de recibir la respuesta. La Plataforma de Implementación genera eventos en respuesta a las acciones del usuario (orales o en forma de acciones como desconectar) y los eventos del sistema (contador de limitadores temporales). Algunos de estos eventos son transmitidos por el Intérprete a sí mismo, tal como se especifica en el documento de VoiceXML, mientras que otros son transmitidos por el Contexto VoiceXML.

El objetivo principal de VoiceXML es llevar todo el poder del desarrollo Web y sus contenidos a los SDO, y favorecer a los desarrolladores la creación de aplicaciones complejas con solicitudes de programación de bajo nivel y amplio manejo de recursos. Este lenguaje permite la integración de servicios de voz con servicios de datos utilizando el familiar paradigma del cliente-servidor. Un servicio de voz es visto como una secuencia de cuadros de diálogo de interacción entre un usuario y una plataforma de aplicación. Los diálogos son proporcionados por Servidores de Documentos, que pueden ser externos a la Plataforma de Implementación. Los Servidores de Documentos deben mantener la lógica general del servicio, realizar operaciones de bases de datos y sistemas heredados, y además producir cuadros de diálogo. Un documento VoiceXML especifica cada interacción de diálogo que se llevó a cabo por el Intérprete. La entrada del usuario afecta a la interpretación del diálogo y se recoge en las solicitudes presentadas a un Servidor de Documentos. El Servidor de Documentos responde con otro documento VoiceXML para proseguir con la sesión del usuario que continúa con otros diálogos.

El lenguaje describe la interacción humano-máquina que ofrecen los sistemas de respuesta de voz, que incluye:

- ◆ La generación de síntesis de voz (text-to-speech).
- ◆ La reproducción de archivos de audio.
- ◆ Reconocimiento de la entrada oral.
- ◆ Registro de entrada oral.
- ◆ Control del flujo del diálogo.
- ◆ Funciones de telefonía como transferencia de llamadas y desconexión.

El lenguaje proporciona los medios necesarios para capturar la información de salida y/o entrada oral, para asignar los resultados de la entrada a la solicitud de variables definidas por el documento, y para la toma de decisiones que afectan a la interpretación de documentos escritos en el mismo idioma. Un documento puede estar vinculado a otros documentos a través del Identificador Uniforme de Recursos (URI).

VoiceXML es una aplicación XML, cuyos principios de diseño se basan en:

- 1) El lenguaje promueve la portabilidad de los servicios a través de la captación de recursos de la Plataforma de Interpretación.
- 2) El lenguaje permite una gran diversidad en cuanto a los formatos de archivos de audio, gramáticas con formato de expresión y regímenes de URI se refiere. Mientras que los diseñadores apoyan diversos formatos de gramáticas, se requiere un formato de gramática común. Esto es el formulario de XML del W3C denominado *Speech Recognition Grammar Specification (SRGS)*. De esta manera se facilita la interoperabilidad. Del mismo modo, mientras varios formatos de audio son válidos para reproducir y grabar, el número de formatos de audio está restringido.
- 3) El lenguaje soporta la facilidad de creación de tipos comunes de interacciones.
- 4) El lenguaje define una semántica bien formada que conserva la intención del diseñador en cuanto a la relación e interacciones con el usuario se refiere. Las heurísticas de los clientes no están obligadas a determinar los elementos de interpretación de documentos.
- 5) El lenguaje reconoce la interpretación semántica de gramáticas y pone esta información a disposición de la aplicación.
- 6) El lenguaje tiene un mecanismo de control de flujo.
- 7) EL lenguaje permite la separación de la lógica del comportamiento y de la interacción de servicios.
- 8) No está diseñado para el cálculo intensivo, las operaciones de base de datos o las operaciones de sistemas heredados. Estos se supone gestionarlo por los recursos externos al Intérprete de Documentos, por ejemplo, un Servidor de Documentos.
- 9) El servicio de lógica general, la administración del estado, la generación del diálogo, y la secuencia de diálogo se supone que residen fuera del intérprete de documentos.

- 10) El lenguaje proporciona vías de enlace a documentos usando URIs, además de enviar información al servidor de documentos usando URIs.
- 11) VoiceXML provee los medios necesarios para determinar con precisión los datos que se presentan al servidor, y el método http (GET o POST) para su uso en la presentación.
- 12) El lenguaje no requiere a los diseñadores del documento para asignar y liberar explícitamente los recursos del diálogo, ni tratar la concurrencia, asignación de recursos y procesos simultáneos de control que deben ser gestionados por la Plataforma de Implementación.

**Adquisición de documentos.** El Contexto del Intérprete VoiceXML espera para adquirir los documentos dirigidos al intérprete VoiceXML para actuar sobre estos. En algunos casos, la solicitud de documentos se genera por la interpretación de un documento de VoiceXML, mientras que las demás solicitudes son generadas por el contexto intérprete en respuesta a eventos fuera del ámbito del lenguaje, por ejemplo una llamada telefónica entrante.

**La salida de audio.** Una Plataforma de Implementación debe compatibilizar la salida de audio con los archivos de audio y texto a voz (*Text-to-Speech*, TTS). La plataforma debe ser capaz de generar una secuencia TTS y una salida de audio libremente. Si un recurso de salida de audio no está disponible, se lanza un evento de error. Los archivos de audio son llamados por un URI.

**La entrada de audio.** Una plataforma de implementación tiene la obligación de detectar e informar sobre la información oral y/o escrita de entrada de forma simultánea, y debe controlarlo mediante la duración del intervalo con un temporizador, cuya longitud se especifica mediante un documento VoiceXML. Si un recurso de entrada de audio no está disponible, un evento de error debe ser lanzado. Se debe informar sobre caracteres (por ejemplo, DTMF) introducidos por un usuario. Debe ser capaz de recibir datos de la gramática de reconocimiento oral de forma dinámica, y debe ser capaz de grabar audio recibido por parte del usuario.

La plataforma **de transferencia** debe ser capaz de apoyar la toma de una conexión de terceros a través de una red de comunicaciones, como el teléfono.

Un *documento de VoiceXML* (o un conjunto de documentos relacionados con una *aplicación*) forma una máquina oral de estados finitos. El usuario está siempre en un estado de conversación o *diálogo* a la vez. Cada diálogo determina el siguiente diálogo al que debe transitar. Las *Transiciones* se especifican mediante URI, que definen el próximo documento de diálogo a usar. Si un URI no se refiere a un documento, el documento actual se supone. Si no hace referencia a un cuadro de diálogo, el diálogo por primera vez en el documento se asume. La ejecución se termina cuando un cuadro de diálogo no especifica un sucesor, o si tiene un elemento que forma explícita las salidas de la conversación.

Hay dos tipos de constructores de diálogos: los llamados *<forms>* y los *<menus>*. Los *<forms>* definen una interacción que recoge los valores de un conjunto de variables de formulario de un documento. Cada campo puede especificar una gramática que define los valores permitidos para ese campo. Una misma gramática y valor de formulario puede ser utilizado para rellenar varios campos de un enunciado. Un *<menu>* se presenta al usuario con una variedad de opciones y, a continuación transiciones a otros diálogos en función de la elección escogida por el usuario.

Un *<subdialog>* es equivalente a una llamada a una función, ya que proporciona un mecanismo para hacer valer una nueva interacción, pudiendo volver a la forma original al concluir la interacción. Los valores de las variables, las gramáticas, e información de estado se guardan y están disponibles al regresar al documento que inicia la llamada. Los *<subdialog>* pueden utilizarse, por ejemplo, para crear una secuencia de confirmación que puede requerir una consulta de base de datos, para crear un conjunto de componentes que pueden ser compartidos entre los documentos en una sola aplicación, o para crear una biblioteca reutilizable de diálogos compartida entre varias aplicaciones.

Una *<session>* se inicia cuando el usuario comienza a interactuar con un Contexto del Intérprete VoiceXML, continúa cuando los documentos son cargados y procesados, y termina cuando lo solicite el usuario, el documento, o el Contexto del Intérprete.

Podemos definir una *aplicación* como un conjunto de documentos cuyo *documento de solicitud* comparte la misma *raíz*. Cada vez que el usuario interactúa con un documento en una aplicación, el documento raíz de la aplicación también está cargado. El documento raíz de la aplicación permanece cargado mientras el usuario se encuentra en una transición entre otros documentos de la misma solicitud, y cuando se carga, las transiciones de usuario se realizan a un documento que no está en la aplicación. Mientras se carga la aplicación, las variables del documento raíz están disponibles para los otros documentos, así como las variables de aplicación, y sus gramáticas permanecen activas durante la duración de la aplicación, sin perjudicar a la activación de una serie de reglas gramaticales.

Cada diálogo tiene una o más estructuras de discurso y/o *gramáticas* DTMF asociados a ella. Como ya hemos comentado con anterioridad, nos podemos encontrar con aplicaciones basadas en diálogos dirigidos, o en iniciativa mixta. En las aplicaciones de *diálogos dirigidos*, el campo asociado a cada una de las gramáticas se activa sólo cuando el usuario está interactuando en ese diálogo concreto. En las aplicaciones de *iniciativa mixta*, donde el usuario y la máquina se alternan en las decisiones de qué hacer a continuación, algunos de los cuadros de diálogo hacen *activa* su gramática (es decir, la aplicación escucha) incluso cuando el usuario está en otro diálogo en el mismo documento, o en otro documento cargado en la misma aplicación. En esta situación, si el usuario dice algo que concuerde con las gramáticas de diálogo activa de otro diálogo, las transiciones ejecutan ese otro diálogo con la expresión dicha por el usuario, tratándolo de esta manera como si se dijera en ese mismo diálogo. La iniciativa mixta suma flexibilidad y naturalidad a las aplicaciones de voz.

VoiceXML proporciona el mecanismo para completar un formulario por el usuario de forma “normal”, como si de una conversación natural se tratase. Además, VoiceXML define otro mecanismo que controla los eventos no cubiertos por el mecanismo del formulario.

Los eventos son lanzados por la Plataforma de Implementación bajo una variedad de circunstancias, como por ejemplo cuando el usuario no responde, no responde de manera inteligible, las realiza una petición de ayuda, etc. El Intérprete también lanza eventos en caso de descubrir un error semántico en un documento VoiceXML. Los eventos son capturados por sus elementos principales o su abreviatura sintáctica. Cada elemento en el que un evento puede actuar especifica los elementos de captura. Por otra parte, la captura de elementos también se realiza de forma heredada. De esta manera, los eventos comunes de manejo de conductas se pueden especificar en cualquier nivel, y se aplican a todos los niveles inferiores.

Por otra parte, un *enlace* apoya la iniciativa mixta. En él se especifica una gramática que se activa cuando el usuario está en el ámbito de la interacción. Si la entrada del usuario coincide con enlace de la gramática, transfiere el control al destino vinculado al URI. Un enlace puede utilizarse para lanzar un evento o ir a un destino URI.

## 2.4.2 XHTML

Con la introducción de la familia de módulos y tipos de documentos de XHTML, el W3C ha favorecido el avance en el desarrollo de la comunidad de Internet, creando un conjunto estándar, de formato correcto y bien formado en el mundo de XML.



En XHTML 1.0, este movimiento fue progresivamente aceptado en cuanto a la migración de contenidos de lenguajes como HTML 4 (y anteriores) se refiere. Con la llegada de los módulos XHTML definidos en XHTML Modularization, el W3C ha eliminado el soporte para los elementos y atributos rechazados por la familia XHTML. Estos elementos y atributos estaban en gran medida orientados a la funcionalidad de presentación de contenidos, manejado más eficazmente a través de las hojas de estilo o especificaciones de comportamiento predeterminado por el cliente.

En el futuro, la familia de tipos de documentos XHTML está basada en esta colección mucho más funcional y más estructural. Más específicamente, el W3C HTML Working Group ha definido un tipo de documento inicial basado exclusivamente en sus módulos. Este tipo de documento está diseñado para ser portable y compatible con una amplia colección de herramientas de los clientes, y aplicable a la mayoría de los contenidos de Internet. Los desarrolladores que basan su contenido en la funcionalidad expresada en esta especificación pueden estar seguros de que se trabajará de manera constantemente para mantener esa portabilidad en toda la familia XHTML.

Podemos definir XHTML como la reformulación de HTML 4 junto a una aplicación de XML. El código XHTML 1.0 especifica tres tipos de documentos XML que corresponden a las tres HTML 4 DTD: *Strict*, *Transitional* y *Frameset*.

XHTML Modularization es una descomposición de XHTML 1.0, y por referencia de HTML 4, en una colección de módulos resumen que proporcionan los tipos específicos de funcionalidad. Estos módulos resumen son implementados utilizando el esquema XML y su definición de tipos de documentos (DTD).

Estos módulos se pueden combinar entre sí y con otros módulos para crear subconjunto XHTML y documentar los tipos de extensión calificados como miembros de la familia XHTML DTD.

XHTML Modularization se refiere a la tarea de especificar los conjuntos bien definidos de los elementos XHTML que se pueden combinar y ampliar por los autores de documentos, dando otras especificaciones XML, y por la aplicación y los diseñadores para que sea económicamente viable para los desarrolladores de contenidos entregar el producto en un amplio abanico de plataformas.

Durante los últimos dos años, muchos mercados especializados han elegido HTML como el lenguaje de contenido para sus proyectos, convirtiéndose en el lenguaje favorito de las plataformas de Internet. Actualmente los desarrollos en HTML giran en torno a los dispositivos móviles (ordenadores portátiles, teléfonos móviles, etc.), dispositivos de televisión (televisores digitales, navegadores basados en Web-TV, etc.), y los PC como terminales fijos. Cada uno de estos dispositivos tiene diferentes requisitos y limitaciones.

XHTML Modularization proporciona un medio para que los diseñadores puedan especificar qué elementos son soportados por un dispositivo concreto y qué componentes se pueden utilizar para ese dispositivo usando bloques de construcción y métodos estándar. Estos módulos sirven como "puntos de adhesión" para la comunidad Web. De esta forma, actualmente el desarrollador puede dirigirse a la base que soporta una colección de módulos, en lugar de preocuparse por la instalación que soporta esta combinación de elementos XHTML o aquella.

El uso de estándares es crítico para modular XHTML y tener éxito a gran escala. No es económicamente viable para los desarrolladores adaptar el contenido a todas y cada una de las combinaciones de elementos XHTML. Al especificar un estándar los procesos software se adaptan perfectamente integrándose directamente en los dispositivos, o es el propio dispositivo el que puede cargar automáticamente el software requerido para procesar un módulo.

La Modularización también permite XHTML aumentar las capacidades de presentación usando los elementos compatibles de XML, sin romper el estándar XHTML. Este desarrollo proporciona estabilidad, usabilidad, una estructura de patrones de fácil implementación y el marco para los desarrolladores y editores de contenidos para gestionar el rápido ritmo del cambio tecnológico en la Web.

Un documento programado en XHTML se puede definir como un conjunto de módulos abstractos. Un módulo define un tipo abstracto de datos que es semánticamente diferente de todos los demás. Este tipo de módulos se pueden combinar en los diferentes tipos de documentos sin llegar a profundizar en los esquemas subyacentes que definen los módulos.

Una implementación del módulo consiste en un conjunto de tipos de elementos, un conjunto de declaraciones de listas de atributos, y un conjunto de modelos de declaraciones de contenido, donde cualquiera de estos tres grupos puede estar vacío. Una lista de declaración de atributos en un módulo puede modificar un tipo de elemento fuera de los tipos definidos en el módulo, y un modelo de declaración de contenidos puede modificar un tipo de elemento fuera del tipo de elemento de conjunto del módulo.

XML DTD es un mecanismo de aplicación. Un DTD XML es un medio de describir la estructura de una clase de documentos XML, conocidos comúnmente como una definición de tipo de documento XML. Otro mecanismo es la aplicación de esquemas XML refiriéndonos a las distintas implementaciones de XHTML Modularization como "esquema".

Un tipo de documento híbrido es un documento compuesto por una colección de tipos de documentos XML o módulos. El objetivo principal del marco de modularización es permitir a un diseñador combinar los elementos de múltiples módulos abstractos en un tipo de documento híbrido, desarrollarlos y validarlos.

Uno de los más valiosos beneficios de SGML XML a través de XML es que reduce la barrera de entrada para la normalización de los conjuntos de elementos que permitan a las comunidades de Internet intercambiar datos en un formato interoperable. Sin embargo, la naturaleza relativamente estática de HTML como lenguaje Web ha significado que cualquiera de estas comunidades haya declarado ya pocas esperanzas de que sus tipos de documento XML sean capaces de ser adoptados en el marco de estándares Web. El marco de modularización permite la incorporación dinámica de estos tipos de documentos diversos dentro de la familia XHTML, reduciendo aún más las

barreras a la incorporación de estos vocabularios específicos de dominio en los documentos XHTML.

Una familia de documentos XHTML es una instancia de un tipo de documentos concretos como se especifica en el documento del diseñador, ya sea en el prólogo del documento o través de algún otro mecanismo (por ejemplo, el atributo SchemaLocation). Validar el documento es el proceso de comprobar si el documento cumple con las normas en el esquema asociado.

Un documento puede estar formado por fragmentos de documentos múltiples. Validar sólo fragmentos de un documento, donde cada fragmento es de un tipo de documento diferente, está más allá del ámbito de aplicación de este marco, ya que requeriría una tecnología que aún no está definida.

Sin embargo, el marco de modularización permite múltiples esquemas que deben integrarse y formar un nuevo tipo de documento (por ejemplo, SVG integrado en XHTML). El nuevo esquema se puede utilizar para la validación en XML 1.0.

Las versiones anteriores de HTML tratan de definir las partes del modelo que los usuarios están obligados a utilizar para formatear un documento. Con la llegada de HTML 4, el W3C inicia el proceso de divorcio de las estructuras. XHTML 1.0 mantiene esta separación, y este documento continúa moviendo a los documentos HTML y sus descendientes por este camino. En consecuencia, este documento no hace ninguna restricción en el modelo de formato asociado a la presentación de los documentos marcados con la familia de documentos XHTML.

### 2.4.3 XML Events

Un *evento* en XML es la representación de algún acontecimiento asincrónico (como un clic del ratón sobre la presentación del elemento, un error aritmético en el valor de un atributo del elemento, o cualquier otra de las muchas posibilidades) que se asocia con un elemento (dirigido a él) en un documento XML.

En el modelo de eventos DOM, cuando ocurre un evento, se envía la información haciéndola pasar por el árbol del documento en una fase llamada de captura, que recoge la información del origen para del evento (llamado su objetivo), donde a continuación, puede repercutir nuevamente en el árbol en la fase denominada de propagación. En general, un evento puede ser respondido por cualquier elemento de la ruta de acceso (un observador) en cada fase al provocar una acción, y/o impidiendo el evento, y/o por la cancelación de la acción predeterminada para el evento. El siguiente diagrama, mostrado en la Figura 12 ilustra la explicación.

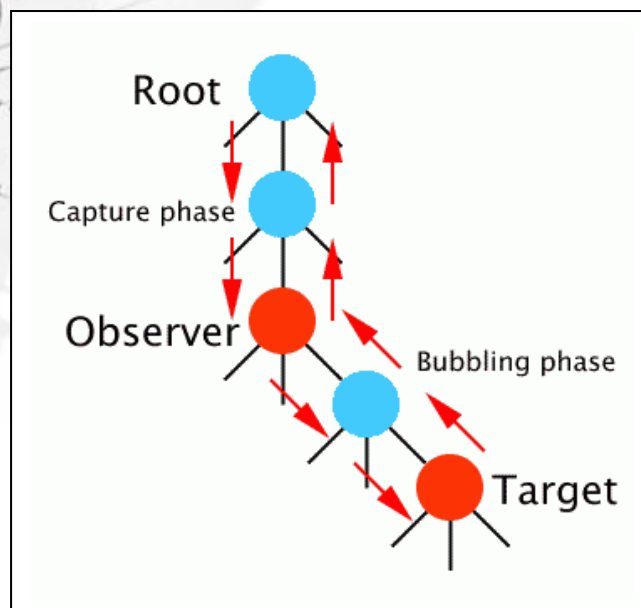


Figura 12. Diagrama dl flujo de eventos en DOM2

[\[2.4.3-A\]](#)

El flujo de eventos en DOM2 se produce entonces de la siguiente manera: un evento dirigido a un elemento (indicada como "objetivo") desciende por el árbol desde la raíz hasta la meta en la fase llamada "captura". Si el tipo de evento lo permite, el evento, viaja de nuevo por la misma ruta en el árbol en una fase llamada "burbujeante". Cualquier nodo en la ruta, incluyendo el nodo raíz y el objetivo, puede ser un "observador": es decir, un controlador unido al objetivo se activa cuando el evento pasa a través de cada fase. Un controlador sólo puede escuchar una fase. Para escuchar otra fase tiene que adjuntar dos controladores.

Una *acción* es una manera de responder a un evento. Un *controlador* es un pliego de condiciones de dicha acción, por ejemplo, mediante secuencias de comandos o algún otro método. Un *oyente* es una unión de ese controlador destinado a unos elementos concretos en el documento.

El HTML se une a un conjunto de eventos mediante la codificación de el nombre del evento en un nombre de atributo, de modo que el valor del atributo es la acción para ese evento en ese elemento. Este método tiene dos inconvenientes fundamentales: en primer lugar la dificultad a la hora de expresar esos eventos en un idioma, de modo que para agregar un nuevo evento, el desarrollador tiene que hacer una traducción, y en segundo lugar le obliga a mezclar el contenido del documento con las especificaciones de las secuencias de comandos y control de eventos, en lugar de permitirle separarlos. *Scalable Vector Graphics* (SVG) utiliza un método similar.

El proceso de definición de una nueva versión de HTML identificó la necesidad de un método de eventos y una extensa especificación. Los requisitos de diseño fueron los siguientes:

- Sintácticamente, exponer el modelo de eventos DOM de un documento XML.
- Proporcionar nuevos tipos de eventos sin necesidad de modificar el DOM o DTD.
- Permitir la integración con otros lenguajes XML.

El DOM especifica un modelo de eventos que ofrece las siguientes características:

- Un sistema de eventos genérico.
- Medios para registrar detectores de eventos y controladores.
- Medios para el enrutamiento de eventos a través de una estructura en árbol.
- El acceso a la información de contexto para cada evento.
- Una definición del flujo de eventos, como se muestra en la Figura 12.

El elemento *oyente* y sus atributos son el método de consolidación de los eventos del DOM nivel 2, encerrando diversos aspectos de la interfaz de eventos, y proporcionando etiquetas a nivel de especificación de las acciones que deben tomarse durante las distintas fases de la propagación de eventos.

Eventos XML no es un tipo de documento independiente. Su objetivo es integrar estos eventos a los lenguajes, tales como XHTML.

#### 2.4.4 DOM

El *Document Object Model* (DOM) es una interfaz de programación de aplicaciones (*Application Programming Interface*, API) para HTML válido y documentos XML bien formados. Define la estructura lógica de los documentos y la forma de acceso y manipulación. En la especificación del DOM, el término "documento" se utiliza en sentido amplio - cada vez más, XML se está utilizando como una forma de representar muchos tipos diferentes de información que pueden ser almacenados en sistemas diversos, y tradicionalmente sería visto como datos en lugar de como documentos. Sin

embargo, XML presenta estos datos como documentos, y el DOM puede ser usado para manejar estos datos.

Con el modelo de objetos de documento, los programadores pueden crear documentos, navegar por su estructura, y añadir, modificar o eliminar elementos y contenido. Cualquier cosa que se encuentra en un documento HTML o XML es accesible y manipulable utilizando el *Document Object Model*, con algunas excepciones. En particular, el DOM interfaces para el interior y exterior de los subconjuntos de XML todavía no se han especificado.

Como una especificación del W3C, un objetivo importante para el DOM es proporcionar una interfaz de programación estándar que puede utilizarse en una amplia variedad de entornos y aplicaciones. Se ha diseñado para ser usado con cualquier lenguaje de programación.

El DOM es un lenguaje de programación de la API para documentos. Estos documentos tienen una estructura lógica que es muy parecido a un árbol, para ser más precisos, que es como un "bosque" ya que puede contener más de un árbol. Cada documento contiene un cero nodos o nodos DOCTYPE, un nodo de elemento raíz (sirve como la raíz del árbol de elementos para el documento), y/o comentarios o más instrucciones de procesamiento. Sin embargo, el DOM no especifica que los documentos deben ser implementados como un árbol, ni tampoco especifica cómo deben ponerse en práctica las relaciones entre los objetos. Una propiedad importante de los modelos de estructura del DOM es el *isomorfismo estructural*: si dos implementaciones DOM se utilizan para crear una representación del mismo documento, se creará el mismo modelo de estructura, de conformidad con el Conjunto de Información XML.

El nombre de "*Document Object Model*" fue elegido porque es un "modelo de objetos" en el sentido tradicional, objeto de diseño orientado a los documentos que se modelan usando objetos. El modelo comprende no sólo la estructura de un documento, sino también el comportamiento de un documento y los objetos que lo componen. Como un modelo de objetos, el DOM identifica:



- Las interfaces y objetos utilizados para representar y manipular un documento.
- La semántica de estas interfaces y objetos, incluyendo los comportamientos y atributos.
- Las relaciones y colaboraciones entre estas interfaces y objetos.

La estructura de los documentos *Standard Generalized Markup Language* (SGML) tradicionalmente ha estado representada por un modelo de datos abstracto, no por un modelo de objetos. En un modelo de datos abstracto el modelo se centra en los datos. En la programación orientada a objetos, los propios datos son encapsulados en objetos que ocultan los datos, que los protege de manipulaciones externas. Las funciones asociadas con estos objetos determinan cómo los objetos pueden ser manipulados, y forman parte del modelo de objetos.

A continuación podemos ver de manera más precisa el DOM distinguiéndolo de otros sistemas que pueden parecersele.

- El DOM no es una especificación binaria. DOM programa escritos en el mismo lenguaje de forma que serán compatibles en código fuente a través de plataformas, pero el DOM no define ninguna forma de interoperabilidad binaria.
- El DOM no es una forma de representación de objetos XML o HTML. En lugar de especificar cómo los objetos pueden ser representados en XML, el DOM especifica cómo los documentos XML y HTML se representan como objetos, de modo que puedan ser utilizados en programas orientados a objetos.
- El DOM no es un conjunto de estructuras de datos, sino que es un modelo de objetos que especifica interfaces. Aunque puedan existir diagramas que muestren relaciones padre/hijo, éstas son relaciones lógicas definidas por las interfaces de programación, no representaciones de ninguna estructura interna de datos en particular.
- El DOM no define qué tipo de información es relevante o cómo está estructurada. Para XML, esto se especifica por el W3C *XML Information Set*. El DOM es simplemente una API que contiene un conjunto de información.
- El *Document Object Model*, a pesar de su nombre, no es un competidor para el *Component Object Model* (COM) siendo este, al igual que *Common Object Request Broker Architecture* (CORBA), una forma independiente del lenguaje

para especificar las interfaces y objetos. El DOM es un conjunto de interfaces y objetos diseñados para la gestión de documentos HTML y XML. No sólo se puede implementar usando sistemas independientes en lenguajes como COM o CORBA, sino que también se puede implementar usando lenguajes específicos como Java o ECMAScript.

El DOM se originó como una especificación para permitir las secuencias de comandos JavaScript y programas en Java permitiendo la portabilidad entre los navegadores Web. El "HTML Dinámico" fue el antecesor inmediato del DOM, pensado en origen en términos de los navegadores. Sin embargo, cuando el DOM *Working Group* se formó en el W3C, fue también proporcionado para los proveedores de otros ámbitos, incluyendo editores HTML o XML y servidores de documentos. Varios de estos proveedores habían trabajado con SGML antes de que XML se desarrollase, y como resultado, el DOM tenido una gran influenciada del SGML Groves y del estándar HyTime. Algunos de estos proveedores también habían desarrollado sus propios DOMs a fin de proporcionar un API para SGML/XML o editores de servidores de documentos, influyendo de igual manera en el DOM.

En las principales interfaces DOM, no hay objetos que representen entidades. Las referencias numéricas de caracteres y las referencias a las entidades predefinidas en HTML y XML, se sustituyen por el carácter individual que compone la entidad de reemplazo. Dado que las referencias de caracteres numéricos y las entidades predefinidas no están reconocidas como tales en las secciones CDATA o en el SCRIPT, y los elementos de estilo en el HTML, no se sustituyen por el carácter único al que parecen referirse. La representación de las entidades generales, tanto internas como externas, se definen dentro de la interfaz extendida del DOM Nivel 1 (XML).

El DOM especifica interfaces que pueden utilizarse para gestionar documentos XML o HTML. Es importante darse cuenta de que estas interfaces son una abstracción (como las "clases de base abstracta" en C++), un medio de especificar una forma de acceder y manipular la representación interna de un documento. Estas interfaces no implican una implementación concreta en particular. Cada aplicación DOM es libre de mantener los documentos en cualquier representación, siempre y cuando las interfaces se muestren en esta especificación como compatibles.

El DOM Nivel 2 no especifica los mecanismos de multithreading o multihilo, es decir, no especifica los mecanismos para la ejecución eficiente de múltiples hilos de ejecución.

El *Document Object Model* Nivel 2 (DOM 2) se ha diseñado con dos objetivos principales. El primer objetivo es el diseño de un sistema de eventos genéricos que permita el registro de controladores de eventos, describa el flujo del evento a través de una estructura de árbol, y proporcione información contextual básica para cada evento.

El segundo objetivo del modelo de eventos es proporcionar un subconjunto común de los sistemas actuales utilizados en caso de DOM Nivel 0. Con ello se pretende fomentar la interoperabilidad de las secuencias de comandos existentes y el contenido. No se esperaba que esta meta se lograra con plena compatibilidad. Sin embargo, la especificación de los intentos de lograr esto cuando sea posible.

### 2.4.5 Gramáticas SRGF y JSGF

Los sistemas de reconocimiento oral están equipados con dispositivos que albergan la capacidad de escuchar el discurso de los usuarios y determinar lo que se dice. La tecnología actual todavía no permite el reconocimiento de voz *sin restricciones*: la capacidad de escuchar cualquier palabra en cualquier contexto y transcribirla con exactitud. Para obtener una precisión razonable en reconocimiento y tiempo de respuesta, los reconocedores actuales limitan lo que escuchan mediante el uso de *gramáticas*.

La sintaxis del formato de la gramática se presenta en dos formas: *Forma Normal de Backus Aumentada* (ABNF) y un formulario XML. La especificación asegura que las dos representaciones son semánticamente cartografiables para permitir transformaciones automáticas entre las dos formas.

- Aumentando la sintaxis de BNF (ABNF): se trata de un texto plano (no XML) que es similar al tradicional BNF, de uso común en el ámbito del reconocimiento oral, incluyendo el *JSpeech Grammar Format* (JSGF). BNF aumentada no debe confundirse con la BNF extendido que se utiliza en la DTD para XML y SGML.
- XML: Esta sintaxis utiliza elementos XML para representar la gramática y adapta los diseños de la gramática PipeBeach y TalkML (variantes de XML) y a la gramática JSGF.

Tanto el Formulario ABNF como el XML *Form* tienen el poder expresivo de un Contexto Libre de Gramática (*Context-Free Grammar*, CFG). Un procesador de gramáticas que no admite gramáticas recursivas tiene el poder expresivo de una máquina de estados finitos (*Finite State Machine*, FSM) o de un lenguaje de expresiones regulares. Esta forma de expresión es suficiente para la gran mayoría de las solicitudes de reconocimiento oral.

Los reconocedores de voz son un tipo importante de procesador gramatical. Otro tipo de procesador es el detector *Dual-Tone Multi-Frequency* (DTMF). Por otro lado, podemos definir al llamado *agente de usuario* como un procesador de gramáticas que acepta datos del usuario y los ajusta a una gramática para producir un resultado que concuerde con la entrada detectada. El tipo de entrada aceptado por un agente de usuario está determinada por el *mode* o los modos de gramáticas que puede procesar, ya que los reconocedores orales y los DTMF utilizan cada uno sus modos de gramática.

Un *reconocedor de voz* es un agente de usuario con las siguientes entradas y salidas:

- Entrada: Una o múltiples gramáticas que informan al reconocedor de las palabras y los patrones de palabras que debe escuchar.
- Entrada: Un flujo de audio que puede contener un discurso que coincida con la gramática/s.
- Salida: Las descripciones de los *resultados* que indican los detalles sobre el discurso detectado por el reconocedor de voz.
- Salida: Errores y rendimiento que puede ser proporcionado por el medio de recepción: por ejemplo, para un navegador oral que incorpora un procesador de gramática.

El principal objetivo de una gramática usada por un reconocedor de voz es permitir que una aplicación oral le indique al reconocedor lo que debe escuchar, y esto puede ser:

- Las palabras que se pueden decir.
- Patrones se pueden producir con estas palabras.
- El idioma.

Los reconocedores de voz también pueden incluir modelos de lenguaje estocásticos. Estas especificaciones definen la manera de crear un reconocedor de voz para detectar la entrada oral, pero una cosa es definir las palabras y otra los patrones de las palabras, aunque sean complementarias. Algunos reconocedores permiten las referencias cruzadas entre las gramáticas de los dos formatos.

La especificación de las gramáticas no se ocupa de una serie de puntos que afectan al reconocimiento del habla:

- Algunos reconocedores de voz soportan la capacidad de adaptarse dinámicamente a la voz de un hablante y, a menudo poseen la capacidad de almacenar datos de adaptación para que esa voz pueda usarse en el futuro. Los datos de los altavoces también pueden incluir las listas de palabras más usadas por el usuario. El formato de la gramática no aborda de forma explícita esta capacidad.
- El formato de las gramáticas no incorpora funciones para la implantación de características tales como reconocedores de los tiempos de espera, los umbrales de los reconocedores, los tamaños de búsqueda o el mejor número de resultados.
- El formato de la gramática no se ocupa de la carga de léxicos o de la pronunciación de palabras. El navegador oral del W3C *Working Group* está estudiando la elaboración de un formato estándar de léxico.
- La tecnología de procesamiento de voz exige la identificación de idiomas, la verificación del altavoz (también conocido como la impresión de voz), el reconocimiento del altavoz (también conocido como la identificación del hablante), entre muchas otras capacidades. Aunque estas tecnologías pueden estar asociadas con un reconocedor de voz, están fuera del alcance de las gramáticas.

Debería ser posible convertir automáticamente una gramática de un formulario ABNF a una gramática de un formulario XML (o al revés), de modo que el rendimiento semántico de las gramáticas sean idénticos. La equivalencia de los resultados semántica implica que:

1. Ambas gramáticas acepten o rechacen el mismo idioma como entrada.
2. Ambas gramáticas analicen cualquier cadena de entrada de forma idéntica

Sin embargo, hay límites inherentes a esta conversión automática:

- El formato de un espacio en blanco no se puede preservar, pero si el significado.
- Algunas construcciones XML no tienen equivalente en ABNF: esquemas XML, DTD, algunos caracteres y declaraciones de entidades y referencias, instrucciones de procesamiento, etc.
- El orden de los comentarios con respecto a la construcción de las gramáticas pueden ser modificados.

Un reconocedor de voz es capaz de hacer coincidir la entrada de audio con una gramática para producir una transcripción del texto en bruto (también conocida como *texto literal*). Un reconocedor puede ser capaz de hacerlo, pero no está obligado a ello, debido a que un tratamiento posterior del texto prima para producir una *interpretación semántica* de la entrada.

Por ejemplo, el enunciado "*Quiero reservar un vuelo de Praga a París*" podría dar lugar a una estructura determinada de datos XML. Para realizar ese paso de enunciado a estructura se requiere interpretación adicional de las instrucciones de procesamiento semántico que pueden estar contenidos en una gramática que define la entrada oral o en un documento asociado.

Actualmente, el navegador por voz del W3C *Working Group* está integrando el desarrollo de la *Semantic Interpretation for Speech Recognition* (SEM). Esta especificación define un lenguaje que puede ser incrustado en las etiquetas dentro de las gramáticas SRGS para llevar a cabo el proceso de interpretación. El procesamiento semántico se define con respecto a la estructura del analizador lógico para el procesamiento de la gramática.

La salida del procesador con respecto a la interpretación semántica puede ser representada utilizando la *Natural Language Semantics Markup Language* (NLSML). Esta representación XML de entrada puede ser usada para transmitir los resultados.

La interpretación semántica llevada a cabo en el proceso de reconocimiento de voz se caracteriza normalmente por:

- La interpretación no se resuelve debido al lenguaje puesto que puede abarcar más de una expresión. Por ejemplo: si el enunciado "*Quiero reservar un vuelo de Praga a París*" fuera seguido por "*Quiero seguir de allí a Londres*" la referencia al "*allí*" no podría ser resuelta o interpretada por "*París*". Ello requiere un análisis que abarca más de un enunciado y se expresa fuera del ámbito de la aplicación del reconocedor de voz, pero de alcance para un gestor de diálogo (por ejemplo, una aplicación VoiceXML).

- Una gramática normalmente se limita a un dominio restringido de entradas (por ejemplo, recoger los datos de reserva de vuelo). Dentro de este dominio semántico la interpretación es una tarea alcanzable mientras que la interpretación semántica de una lengua entera es una tarea extraordinariamente compleja.
- Dado que cada idioma tiene sus propias estructuras lingüísticas, el proceso de conversión de un texto sin formato a un resultado semántico es necesariamente específico de cada lenguaje.

Las gramáticas se ven sometidas a este tipo de restricciones. En cambio, una aplicación VoiceXML recibe un resultado de voz con la interpretación semántica del usuario para llevar a cabo un diálogo. La aplicación también puede llevar a cabo análisis semánticos más profundos.

La *gramática JSpeech Grammar Format (JSGF)* define una plataforma independiente del proveedor en la manera de describir un tipo de gramática: la *gramática de reglas* (también conocido como *símbolos de control*) y la *gramática regular*. Utiliza una representación textual que es legible y editable por los desarrolladores y equipos, y puede ser incluido en el código fuente.

Una gramática de reglas especifica los tipos de *enunciados* que un usuario podría decir (una expresión oral igual a una frase escrita). Por ejemplo, una gramática con una ventana de control simple puede escuchar "abrir un archivo", "cierre la ventana", y comandos similares.

Lo que el usuario puede decir depende del contexto: ¿está realmente el usuario controlando una aplicación de correo electrónico, de lectura de un número de tarjeta de crédito, o de selección de una fuente de texto? Las aplicaciones conocen el contexto, por lo que las aplicaciones son responsables de proporcionar un reconocedor de voz con gramáticas adecuadas.



## 2.5 Ejemplos de SDOs

En las siguientes líneas presentamos algunas de las aplicaciones cuya interfaz está diseñada para interactuar con el usuario a modo de Sistema de Diálogo Oral [\[2.5-A\]](#):

**Ariadne [\[2.5-B\]](#):** Se trata de una plataforma cuyo uso principal es la creación de prototipos de SDOs de una manera rápida. Gracias a los motores MS SAPI y a los sistemas que transforman el texto a diálogo, el usuario puede construir un SDO entero. Esta plataforma ha sido diseñada de manera que, gracias a la ayuda de la codificación en unicódigo UTF8, permite el uso de otros idiomas, incluyendo chino y japonés.

**August [\[2.5-C\]](#):** Se trata de un sistema de diálogo multimodal de origen sueco, que usa un agente animado con el aspecto de un autor anterior al siglo XIX llamado August Strindberg. Basado en los componentes de la tecnología oral desarrollada por el *Center for Speech Technology*, una escuela de basada en estudios de la computación y la comunicación. La animación llamada August se comunica con el usuario a través de diálogos sintéticos, expresiones faciales y movimientos de cabeza, acordes con los movimientos de cabeza de los usuarios captados a través de una cámara. La información adicional se muestra a través de nubes de pensamiento que manan de la cabeza de August. El sistema ha sido desarrollado por investigadores del CTT de Estocolmo.

**Conquest [\[2.5-D\]](#):** Sistema de diálogo que muestra los horarios de las conferencias desarrolladas durante el congreso Interspeech-2006 y IJCAI-2007. El sistema ha sido desarrollado por investigadores pertenecientes a la Carnegie Mellon University, en los Estados Unidos.

**DINEX [\[2.5-E\]](#):** Se trata de un sistema conversacional que proporciona al usuario información sobre localización de restaurantes en el área de Boston y cómo encontrarlos.

**ELVIS (Email Voice Interactive Systems) [2.5-F]:** Este sistema oral de carácter experimental permite al usuario el acceso a su correo electrónico con un agente conversacional llamado “Elvis”. Desarrollado por investigadores de AT&T.

**How May I Help You? [2.5-G]:** En esta aplicación podemos ver un sistema *call-routing* desarrollado por el servicio de atención al cliente de AT&T. El sistema recoge la información del usuario transmitida a través de un diálogo y predice las necesidades del usuario, reconduciendo la conversación en el caso de que necesite información adicional.

**Intelligence Procedure Assistance (IPA) [2.5-H]:** Es un SDO multimodal que proporciona la ayuda necesaria a los astronautas situados en la Estación Espacial Internacional durante la ejecución de tareas y listados de comprobaciones de procesamientos. Los diseñadores de este sistema pertenecen a RIALIST Group, RIACS/NASA y Ames Research Center.

**Jupiter [2.5-I]:** Este Sistema de Diálogo Oral está diseñado para proveer al usuario de la información meteorológica actualizada día a día a través del teléfono. Ha sido desarrollado por el Instituto Tecnológico de Massachusetts (MIT), en Estados Unidos.

**LARRI (Language-based Agent for Retrieval of Repair Information) [2.5-J]:** Sistema encargado de la recuperación de información relacionada con las reparaciones. Es un SDO multimodal utilizado por el servicio de mantenimiento de los aviones F18 para la reparación y diagnóstico de los dispositivos. Los diseñadores pertenecen a la Universidad Carnegie Mellon, a la escuela de Ciencias de la Computación.

**Mercury [2.5-K]:** Es una interfaz oral que proporciona información sobre vuelos (precios, horarios, trayectos) a través del teléfono. Mostramos un ejemplo de conversación entre un usuario y el sistema Mercury:

- **Mercury:** *Welcome to Mercury, the MIT air travel planning system. I can help plan your air travel itinerary. Please tell me your user name. If you do not have a user name, please say new user.*
- **USER:** NEW USER

- **Mercury:** *Welcome. You have successfully logged onto Mercury as a new user. How may I help you?*
- **USER:** I WOULD LIKE TO FLY FROM BOSTON TO LONDON
- **Mercury:** *Flights from Boston to London. What date will you be travelling?*
- **USER:** NEXT FRIDAY IN THE EVENING
- **Mercury:** *Okay. Flights from Boston to London departing between 5:00 pm and midnight on June 25. Can you provide a departure or arrival time? The earliest nonstop flight from Boston to London departing between 5:00 pm and midnight on June 25 leaves at 5:30 p.m. and arrives at 4:55 a.m. on British Airways. You arrive one day later.*
- **USER:** IS THERE A UNITED FLIGHT
- **Mercury:** *Okay. United flights from Boston to London departing between 5:00 pm and mid night on June 25. One moment please... I have a nonstop united flight leaving at 6:40 p.m. and arriving at 6:25 a.m.?*
- **USER:** BOOK THAT FLIGHT
- **Mercury:** *United flight 998 has been added to your itinerary. You depart Boston at 6:40 pm and arrive in London Heathrow at 6:25 am. You arrive one day later. Where would you like to go next?*
- **USER:** I WOULD LIKE TO RETURN TO J F K THE FOLLOWING TUESDAY
- **Mercury:** *Okay. United flights from London Heathrow to New York's Kennedy on June 29. One moment please... I have a flight departing at 1:55 p.m., a flight departing at 9:00 a.m., and a flight departing at 6:00 p.m. Please select one.*
- **USER:** BOOK THE EARLIEST FLIGHT
- **Mercury:** *United flight 957 has been added to your itinerary. You depart London Heathrow at 9:00 am and arrive in New York's Kennedy at 11:50 am. Where would you like to go next?*
- **USER:** I WOULD LIKE TO RETURN TO BOSTON THAT SAME EVENING
- **Mercury:** *Okay. United flights from New York's Kennedy to Boston departing between 5:00 p.m. and midnight on June 29. One moment*

*please... I have a flight departing at 8:55 p.m., and a flight departing at 7:00 p.m. Please select one.*

- **USER:** I WOULD LIKE THE SEVEN O'CLOCK FLIGHT
- **Mercury:** *United Airlines flight 7282 leaves New York's Kennedy at 7:00 pm and arrives in Boston at 8:31 pm. The total flight time is 1 hours and 31 minutes. The aircraft is a Jetstream 41 Turbo. Shall I book t his flight for you?*
- **USER:** YES PLEASE
- **Mercury:** *United flight 7282 has been added to your itinerary. You depart New York's Kennedy at 7:0 0 pm and arrive in Boston at 8:31 pm. Shall I price your itinerary?*
- **USER:** YES
- **Mercury:** *The price for your itinerary is \$2065,40. Is there anything else I can do for you?*
- **USER:** NO THANK YOU GOOD BYE
- **Mercury:** *Thanks for using mercury. Have a nice day!*





# Capítulo 3

## *Entorno X+V: Voice Applications*

### 3.1 Introducción

El sistema desarrollado para el Proyecto Final de Carrera con la tecnología XHTML+Voice para el manejo de un entorno Web de manera multimodal ha sido denominado *Voice Applications* (VA). A través del acceso a la página principal del entorno nos encontramos un portal amigable que alberga cinco aplicaciones preparadas para ser utilizadas tanto de forma tradicional con ratón y teclado, como mediante la voz, alternando e incluso acoplando a la vez los modos visual y acústico.

En esta primera página de presentación, el sistema tiene predeterminado un mensaje de bienvenida, a través de la interfaz oral, que permite su interrupción si se desea acceder directamente a cualquiera de las aplicaciones, haciendo gala la iniciativa mixta anteriormente mencionada. La interacción oral con la página, en este caso, se realiza mediante dos tipos diferentes de interacciones:

- Haciendo que el puntero del ratón se sitúe encima o pase por encima de cualquiera de los textos descriptivos que acompañan a las imágenes de las aplicaciones, activando la interfaz oral y haciendo que el navegador lea literalmente esos textos al usuario.
- A través de la navegabilidad insertada en la estructura de la página, que permite, en cualquier momento, hablarle al entorno y decirle el nombre de cualquiera de las aplicaciones que el usuario desee visitar, haciendo que el sistema cargue dicha aplicación.

En la interfaz visual de esta primera página se presenta una imagen a modo de logotipo corporativo en la cabecera de la página, tras el cual se observa un menú con las cinco aplicaciones, el cuerpo de la página con la información, y un pie de página con acceso a las páginas de información legal, información de contacto, y políticas de privacidad. Esta estructura se ha mantenido en todas y cada una de las páginas del entorno VA.

El contenido del cuerpo de esta página principal (a la cual el usuario puede acceder desde cualquier diálogo del entorno pronunciando la palabra “Home”) consta de una sencilla presentación de cada una de las aplicaciones. Estas aplicaciones se muestran con su imagen representativa a la izquierda, y alineado a la derecha un texto que describe brevemente el contenido de la aplicación. Ambos elementos, texto e imagen, contienen el enlace correspondiente a la aplicación. El enlace se presenta tanto en su interfaz visual como en su interfaz oral.

Como se ha descrito anteriormente, en lo que respecta a la interfaz oral, el texto descriptivo tiene insertado el código el evento de XML *“onMouseOver”*. Este evento llama al formulario de voz correspondiente, que a su vez activa la voz del navegador para narrar el texto situado bajo el puntero. De esta forma el entorno permite la lectura de los textos por los que el usuario pasa el ratón. Esta sencilla presentación nos muestra el proceso a través del cual se activa el motor oral:

- 1) Elemento visual diseñado en XHTML, en este caso el elemento es el texto descriptivo de una aplicación.
- 2) Evento de XML a modo de enlace (evento *“onMouseOver”*)
- 3) Formulario de VoiceXML, mediante el texto descriptivo de esa aplicación entre etiquetas de narración de contenidos *“<vxml:block>”*.

En la Figura 13 podemos observar la página principal del entorno VA y su estructura, diseñada en la interfaz visual:

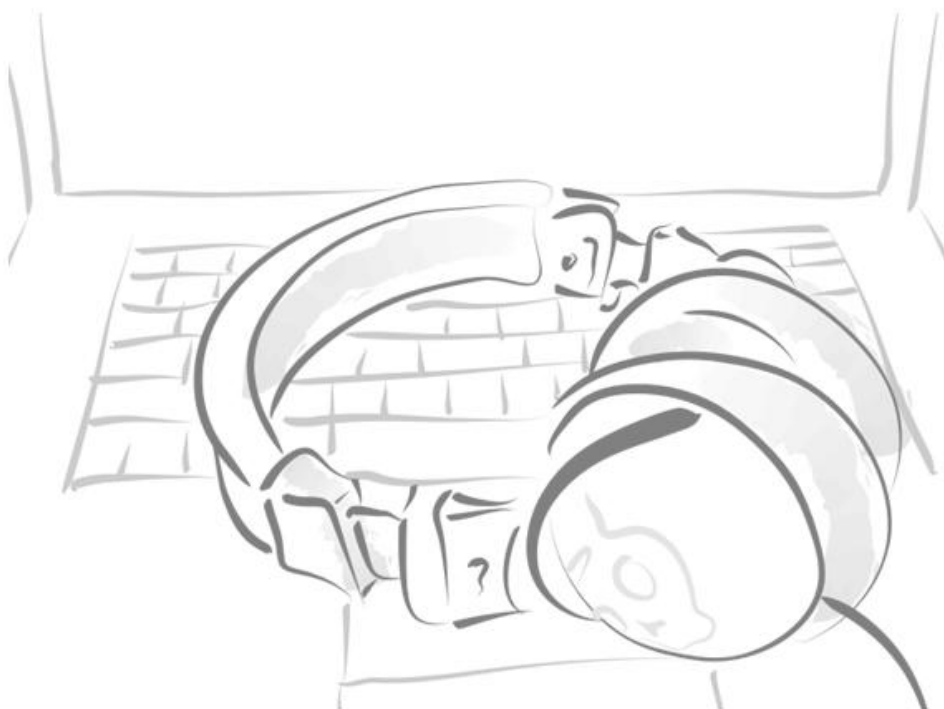






Figura 13. Presentación de Voice Applications

El entorno tiene como pie de página los vínculos enlazados a las Webs que contienen toda la información relacionada con la privacidad y legalidad. Se puede acceder a las páginas de *Legal Information*, *Contact Us*, y *Privacy Policy* a través de cualquiera de las páginas del entorno VA, como si de una aplicación más se tratase, favoreciendo así la navegabilidad completa de todo el entorno. Las tres páginas tienen el mismo diseño y arquitectura, diferenciándose únicamente por el contenido.

La página *Legal Information* tiene registrada en el cuerpo de su estructura toda la información legal presente en el entorno web. Esta información trata la legalidad vigente sobre la propiedad intelectual de los contenidos presentados en el entorno (la propiedad de las páginas y del código utilizado en estas, el permiso a cualquier página para vincular a este entorno excepto en forma de frame sin el permiso de los propietarios de VA, la no responsabilidad de los links a otros entornos que pudiese haber en VA, etc). Todo este contenido es leído al usuario a través de la interfaz oral de la página, dando la posibilidad de interrumpir este diálogo en cualquier momento, y de navegar por el entorno.

*Privacy Policy* es la página en cuyo cuerpo se redactan los puntos esenciales de privacidad de contenidos y datos de usuarios que acceden al entorno (VA no recopila información de usuarios, las configuraciones de seguridad protegen del mal uso de los contenidos, los usuarios son libres de navegar por el entorno sin ningún tipo de registro, etc.). Esta información se transmite al usuario a través de la interfaz visual (texto escrito), y también a través de la interfaz oral, siendo leído al usuario cuando accede a esta página mediante el evento de XML “*onLoad*” que activa el diálogo.

En la página *Contact Us* el usuario puede encontrar toda la información relacionada con el diseñador del entorno (nombre del diseñador, título universitario, correo electrónico, y última fecha de revisión del entorno). Al igual que ocurre con el resto de páginas pertenecientes al pie de página, toda esta información es narrada al usuario a través de la interfaz oral.

## 3.2 Voice Dictionary

La primera aplicación del entorno a la que podemos acceder ha sido denominada *Voice Dictionary* (VD). A través de esta aplicación entramos en un entorno sencillo mediante el cual podemos dialogar con la aplicación y buscar definiciones o biografías obteniendo el mismo resultado que produce Wikipedia, con la particularidad de que en VD el contenido resultante de la búsqueda es narrado íntegramente al usuario. Además, esta búsqueda de contenidos se puede realizar tanto de manera tradicional (ratón y teclado) como mediante el uso de la voz. De esta forma, proporcionamos a un entorno robusto y de gran utilidad como es Wikipedia, la multimodalidad necesaria para eliminar la barrera arquitectónica de Internet más importante. En la Figura 14 se presenta la aplicación VD en su forma visual:

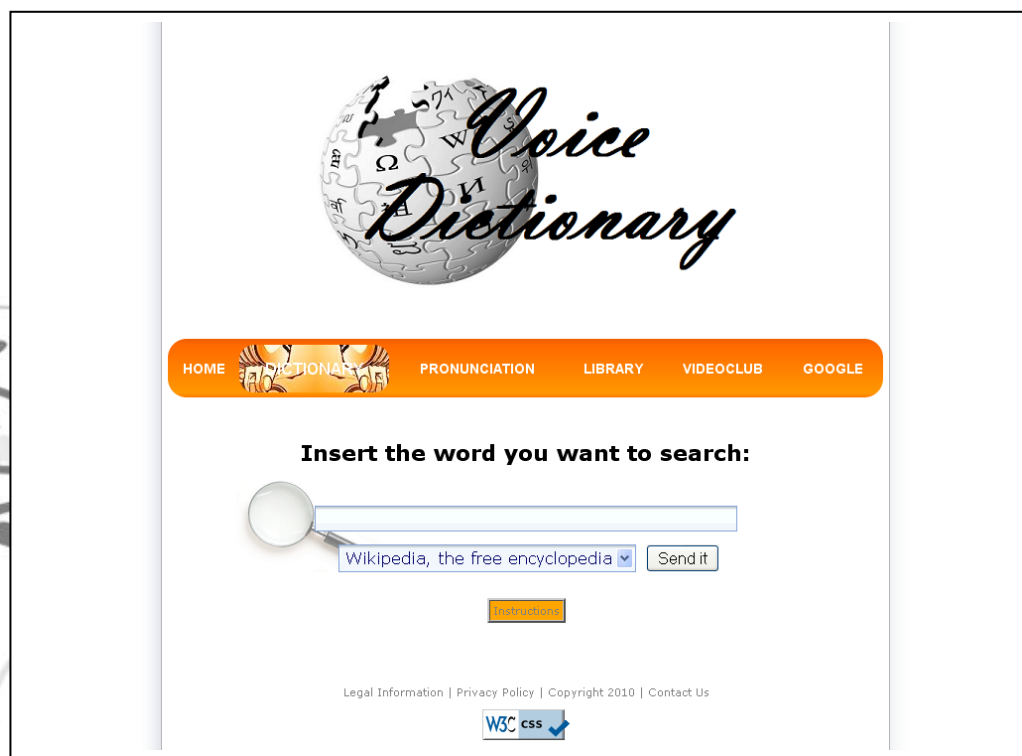
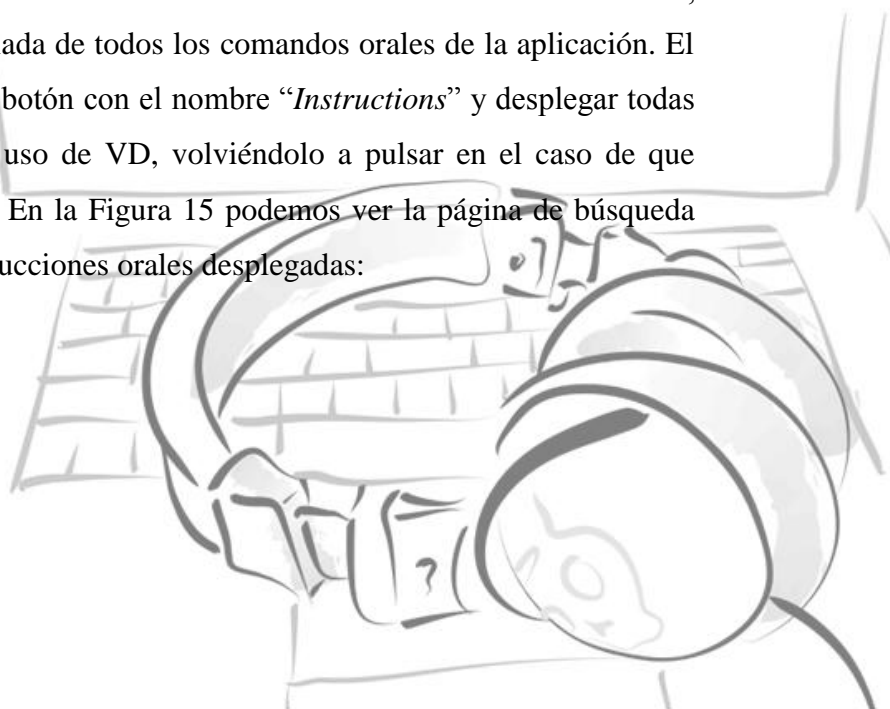


Figura 14. Presentación de aplicación *Voice Dictionary*

Como podemos observar en esta imagen, al igual que en la presentación del portal VA en su Figura 13, la interacción entre el entorno y el usuario, tanto en su interfaz visual como en su interfaz oral, se produce en inglés. Todos los diálogos son en este idioma, de manera que el usuario debe responder siempre en inglés. Esto favorece la internacionalidad del portal VA, permitiendo el acceso a un mayor número de usuario, incluso a aquellos con un nivel bajo de inglés, pues, como veremos más adelante, se presenta una aplicación en forma de juego que ayuda a mejorar la pronunciación y el vocabulario en este idioma.

### 3.2.1 Funcionamiento

La aplicación muestra en su inicio un mensaje de bienvenida en su interfaz oral, explicando brevemente el funcionamiento y los comandos básicos mediante la voz al usuario. Esta breve introducción informa al usuario del comando “*INSTRUCTIONS*”, cuyo uso es la descripción detallada de todos los comandos orales de la aplicación. El usuario también puede pulsar el botón con el nombre “*Instructions*” y desplegar todas las instrucciones orales para el uso de VD, volviéndolo a pulsar en el caso de que desee cerrar estas instrucciones. En la Figura 15 podemos ver la página de búsqueda de la aplicación VD con las instrucciones orales desplegadas:



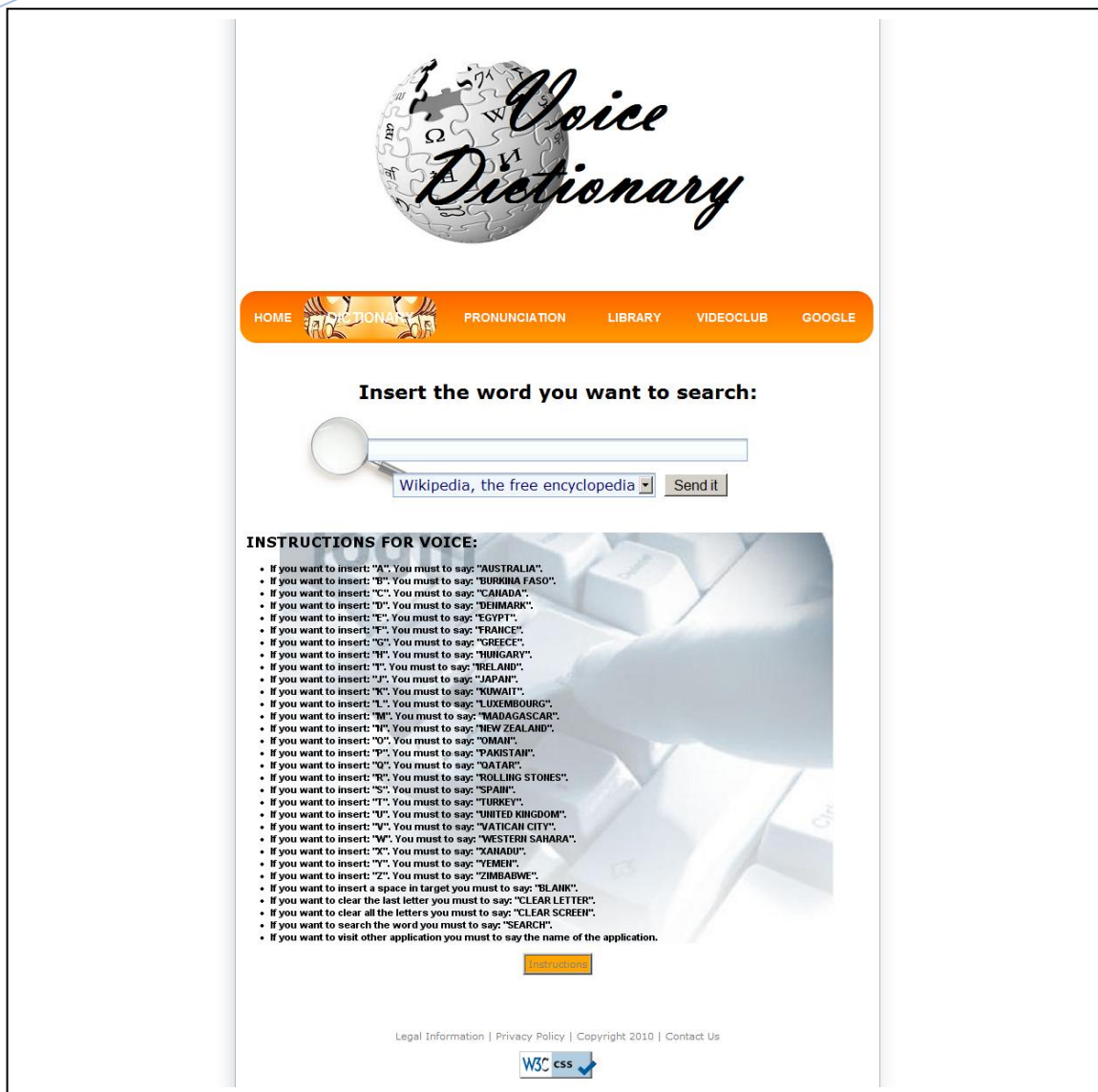
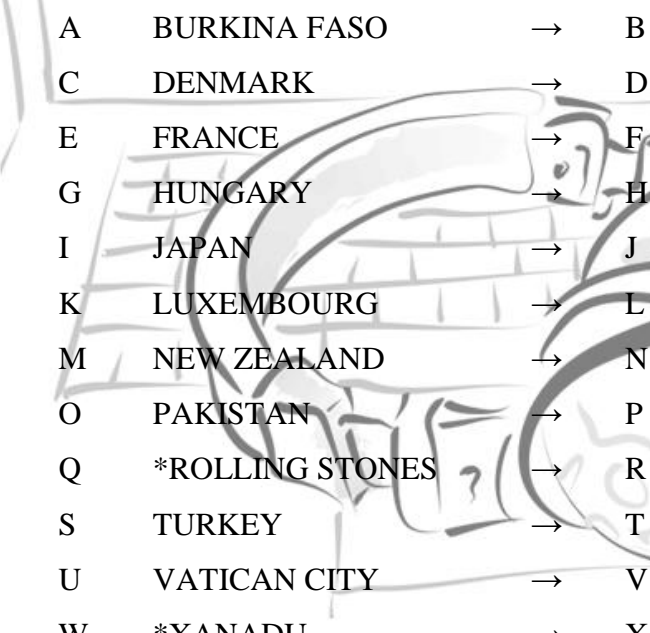


Figura 15. Aplicación VD con las instrucciones desplegadas

Como hemos visto en la descripción de los diferentes estándares de gramáticas en el capítulo anterior, el uso de interfaces orales en entornos Web implica el uso de gramáticas. Estas gramáticas ayudan al diseñador a establecer los límites de recepción de información del motor oral de la aplicación. Este listado de palabras especificado en la gramática limita en exceso la tarea de un buscador, debido principalmente a que no se puede restringir o compilar un listado de palabras de una gramática a todas las posibles combinaciones de palabras de una lengua.

En el caso de la aplicación VD, las gramáticas impedían la libertad de búsqueda de los usuarios, pues limitaban el número de palabras, obligando al desarrollador a solucionarlo mediante el establecimiento de parámetros que delimitaban la búsqueda a un género o contexto específico. Sin embargo, para poder cubrir el mayor rango de posibilidades de búsqueda se estableció en la aplicación un sistema que favorecía esta total y absoluta libertad de búsqueda sin excluir el uso de las gramáticas: deletrear el contenido de la búsqueda, haciendo que el listado de palabras sea el conjunto de letras del alfabeto. A priori, puede parecer un método de interacción con el entorno menos natural, de diálogo más forzado. Sin embargo, a cambio podemos realizar cualquier tipo de búsqueda sin restricciones de contenidos.

A lo hora de desarrollar el sistema que aceptase las letras en lugar de las palabras nos vimos en la dificultad de hacer entender a la interfaz oral que lo que está diciendo el usuario es una “M” y no una “N”. La similitud a la hora de pronunciar las letras en inglés dificultaba la tarea del reconocedor oral, complicando la interacción con el usuario, dado que el motor oral necesita palabras largas para asociarlas con los elementos de la gramática. Como solución, introdujimos un listado de países asociados a cada una de las letras, de manera que cuando el usuario diga “SPAIN”, la aplicación se encargue de pintar en el *input* del formulario de búsqueda sólo la letra “S”. A continuación podemos ver el listado completo de palabras asignadas a cada letra:



AUSTRALIA	→	A	BURKINA FASO	→	B
CANADA	→	C	DENMARK	→	D
EGYPT	→	E	FRANCE	→	F
GREECE	→	G	HUNGARY	→	H
IRELAND	→	I	JAPAN	→	J
KUWAIT	→	K	LUXEMBOURG	→	L
MADAGASCAR	→	M	NEW ZEALAND	→	N
OMAN	→	O	PAKISTAN	→	P
QATAR	→	Q	*ROLLING STONES	→	R
SPAIN	→	S	TURKEY	→	T
UNITED KINGDOM	→	U	VATICAN CITY	→	V
WESTERN SAHARA	→	W	*XANADU	→	X
YEMEN	→	Y	ZIMBABWE	→	Z

(\*) Algunos de los comandos usados no son países por la inexistencia de un país con esa letra, o por la dificultad de pronunciación de los países existentes con esa letra. Veremos el proceso de concatenación de las letras en el apartado de Arquitectura de esta misma aplicación.

Para introducir un espacio en blanco, en caso de que el usuario desee insertar varias palabras, existe en el entorno el comando “*BLANK*”, cuya función es insertar en el campo del formulario correspondiente este espacio. La combinación de comandos alberga además la posibilidad de que el usuario hierre a la hora de escribir una letra. Para este supuesto existe el comando “*CLEAR LETTER*”, encargado de borrar la última letra insertada. Si el usuario desea borrar además todas las letras debe decir el comando “*CLEAR SCREEN*”, borrando todo el contenido del campo del formulario. Una vez que el usuario ha finalizado el deletreo del contenido puede o bien usar el comando “*SEARCH*” para buscar ese contenido en la plataforma, o bien puede (en este instante o cualquier instante durante el proceso de deletreo) decir el nombre de cualquiera de las aplicaciones para visitarlas.

En la aplicación VD, al igual que en toda la plataforma VA, se pueden intercalar la forma de interactuar con la página, pudiéndose utilizar los comandos orales para introducir una palabra y a la vez insertar algunas letras mediante las pulsaciones de teclado. De esta forma, favorecemos la total intercomunicación entre las interfaces orales y visuales.

Una vez que el contenido de la búsqueda se muestra por pantalla y la interfaz oral comienza a narrarlo, el usuario puede perfectamente visitar cualquier otra aplicación interrumpiendo la narración, o acceder a cualquiera de los enlaces destacados en el texto. Esta función se aleja del deletreo inicial favoreciendo la natural interacción oral entre el usuario y la plataforma VA. El proceso almacena cada uno de los vínculos encontrados en el contenido resultante de la búsqueda en el listado de la gramática asociada, favoreciendo al usuario el acceso a cualquiera de los vínculos creando una red de hipertexto en el entorno VA. De esta forma, podemos navegar por la plataforma Wikipedia utilizando únicamente la voz, y sin salir del entorno del proyecto.

### 3.2.2 Diseño

El sistema está desarrollado no sólo para ser lo más eficiente posible sino también para mostrar al usuario una estructura elegante y acorde a la función que cumple. Se ha diseñado una imagen principal para esta aplicación con las letras propias del entorno y el logotipo de la plataforma Wikipedia como fondo. De esta forma conseguimos enfocarlo como se pretende: una aplicación nueva pero a la vez basada en otra ya existente a la que le damos una nueva funcionalidad.

Toda la página está centrada para poder ofrecer la misma definición y calidad independientemente del navegador que se utilice. Si bien es cierto que la funcionalidad oral sólo se consigue con el navegador Opera, debido a que todo el entorno está preparado para ser manejado desde el punto de vista tradicional (ratón y teclado) las aplicaciones pueden usarse con cualquier navegador. Esta estructura centrada utiliza un “contenedor” limitado por unas barras sombreadas. Este efecto se consigue mediante una degradación de una serie de píxeles ordenado en una línea de negro a blanco en dos extremos separados. La imagen que produce a simple vista pasa desapercibida, en cambio aplicando una repetición de la imagen en toda la página se consiguen dos líneas verticales paralelas sombreadas, produciendo un efecto de contenedor resaltado sobre un fondo blanco.

Se ha diseñado un menú principal común a todo el entorno de forma que combinase a la vez la simplicidad de programación y la dinamicidad estética de un entorno serio y profesional. Este menú se ha conseguido diseñando una tabla sin bordes donde cada celda es un elemento del menú. El fondo de la celda es el fondo de un menú en tonos naranjas y formas resaltadas para darle un aspecto más creativo. A la vez, se han buscado dos imágenes para indicar al usuario en qué posición del menú se encuentra (mediante una imagen de tonos naranjas y blancos para el fondo de la celda correspondiente a la aplicación que se está visitando actualmente), y otra imagen más oscura, con tonos naranjas y negros, para resaltar las celdas cuando el usuario pasa el ratón por encima, dando el aspecto deseado de botones. De esta manera tan



simple se consigue un menú a la altura de las circunstancias sin entrar en programaciones más complicadas.

El buscador incrustado en la aplicación fue desarrollado de manera que las instrucciones de la búsqueda apareciesen en la propia barra de búsqueda, en un color claro, en lugar del título tradicional que aparece en derredor. Esta información desaparecía de la barra cuando el usuario ubicaba el foco en ese elemento (haciendo click), y el color de la fuente del texto cambiaba de gris a negro, para que el contenido de la búsqueda quedase resaltado respecto al texto predefinido. Así conseguíamos un efecto atractivo economizando espacio entorno al *input*. En cambio, este nuevo desarrollo tuvo que eliminarse de proyecto por dificultar las funciones orales relacionadas con el elemento visual. Este detalle quedará mejor explicado en el apartado de Pruebas y Limitación de Errores.

Una vez encontrado el contenido de la búsqueda del usuario, se ha “limpiado” ese contenido desde el punto de vista visual y se ha mostrado al usuario de manera que encaje perfectamente con el entorno que visita. El texto mostrado se encuentra encuadrado en el contenedor principal, con un tipo de letra y estilo que no haga destacar más que los enlaces que el usuario puede visitar de manera multimodal. Los aspectos de esta “limpieza” de código se muestran más desarrollados en el apartado de Arquitectura de esta misma aplicación.

### 3.2.3 Arquitectura

La aplicación VD tiene como acceso inicial el archivo “*sdictionary.xhtml*”. Este archivo, programado en texto plano, contiene las líneas de código relacionadas con la interfaz visual y oral del buscador VD. Este archivo envía la información de búsqueda a un archivo de proceso programado en PHP, denominado “*dictionary\_process.php*”. El proceso recoge la información, la envía al entorno Wikipedia, y recoge la

información de resultado. Posteriormente procesa esa información y crea un archivo de resultado denominado “*rdictionary2.xhtml*” a partir de una plantilla creada anteriormente (“*rdictionary.xhtml*”), el cual contiene las interfaces orales y visuales necesarias para poder interactuar con el usuario y ofrecerle la posibilidad de realizar búsquedas en hipertexto. La Figura 16 representa el flujo de información que realiza la aplicación VD, desde que el usuario transmite el contenido de la búsqueda hasta que recibe el resultado.

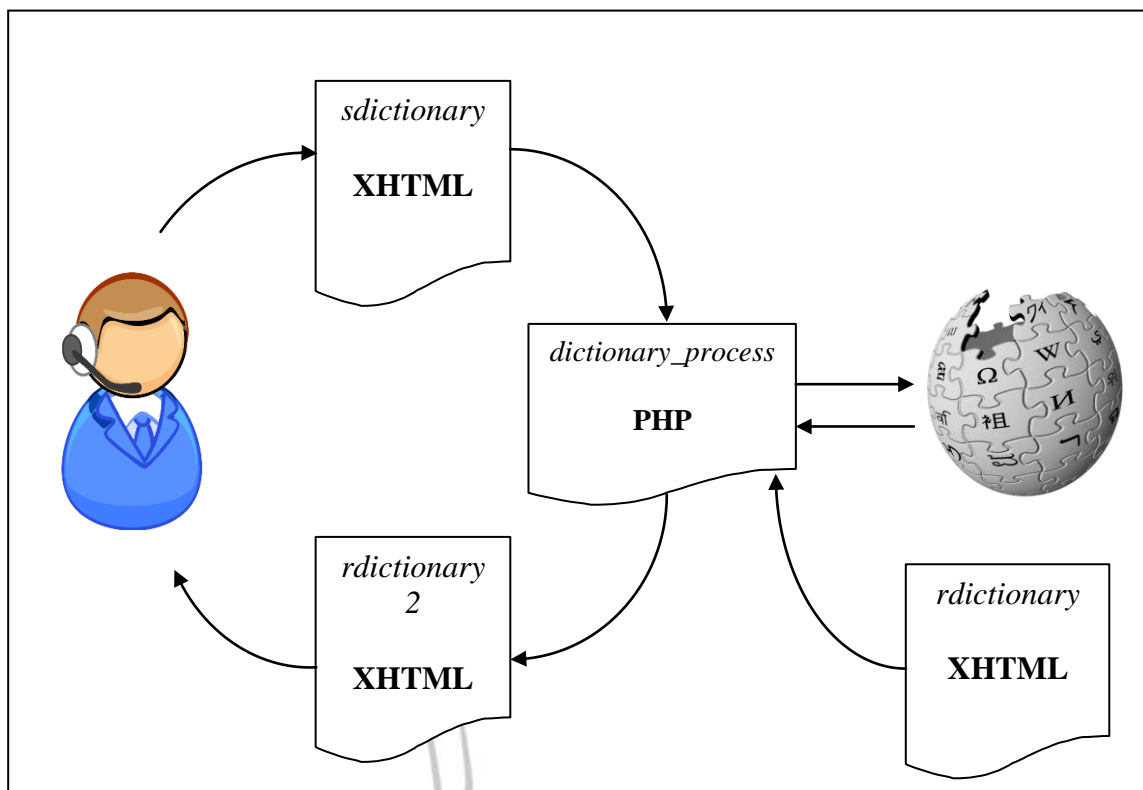


Figura 16. Flujo de información de VD

El código contenido en el primero de los archivos que componen la aplicación VD está estructurado en tres partes:

- 1) Cabecera con toda la información relacionada con la versión, etiqueta <html>, y DOCTYPE.
- 2) Etiqueta <head> con toda la información de referencia a la plantilla de estilos CSS, y JavaScript, además de todos los diálogos que componen la interfaz oral.

- 3) Etiqueta <body> guardando toda la información relacionada con los elementos visuales y de formularios con los que el usuario puede interactuar de manera tradicional, además de todas los eventos XML que activan los correspondientes diálogos de la interfaz oral.

Veamos más detenidamente cada una de las partes de la estructura de este archivo. La primera parte contiene la siguiente información y cabeceras, avisando al compilador de su condición no sólo de código HTML, sino de contenidos XML (XHTML), y contenidos vxml, relacionando a cada uno con su estándar del W3C, mostrado en la Figura 17:

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml"  
    xmlns:vxml="http://www.w3.org/2001/vxml"  
    xmlns:ev="http://www.w3.org/2001/xml-events"  
    xmlns:xv="http://www.voicexml.org/2002/xhtml+voice">
```

Figura 17. Código de información de cabecera.

Esta información se mantiene constante en cada uno de los archivos codificados en XHTML.

En la segunda parte de la estructura tenemos, como hemos visto antes, las referencias a las plantillas de estilos CSS y al código JavaScript. Además tenemos programada en esta parte toda la estructura VoiceXML. Esta estructura se compone de un formulario llamado "nav", cuyo contenido introduce al usuario mediante un diálogo, y se mantiene escuchando a la espera de recibir la información del usuario. Si no recibe una respuesta correcta pide al usuario que repita la información, y vuelve a esperar. Esta información es comparada con la gramática contenida en el archivo

“abd.jsgf”. La gramática tiene asignada una lista de palabras fuera de las cuales toda información es rechazada. Podemos ver estas palabras en la Figura 18:

*search | clear screen | clear letter | blank | instructions | Australia | Burkina Faso | Canada | Denmark | Egypt | France | Greece | Hungary | Ireland | Japan | Kuwait | Luxembourg | Madagascar | New Zealand | Oman | Pakistan | Qatar | Rolling Stones | Spain | Turkey | United Kingdom | Vatican City | Western Sahara | Xanadu | Yemen | Zimbabwe | home | dictionary | pronunciation | library | videoclub | google | legal information | privacy | contact us*

Figura 18. Código de información de cabecera.

En esta lista podemos encontrar todos los comandos a través de los cuales el usuario puede interactuar. Una vez que el formulario “nav” captura la información correcta correspondida con uno de los comandos, procede a ejecutar el proceso correspondiente a cada comando. Si se trata del nombre de una aplicación modifica el valor de *window.location* haciéndolo coincidir con la URL de la aplicación correspondiente. En caso de que se trate de un comando de la lista de países, la aplicación se encarga de concatenar el contenido del *input* con la primera letra del país referenciado por el usuario, recogiendo el valor actual del elemento, añadiéndole la nueva letra, y sustituyendo el valor actual por el nuevo valor.

Al finalizar el diálogo, una variable se encarga de almacenar el contenido para poder volver a un estado anterior en el caso de que el usuario decida borrar una letra. Si el usuario desea borrar todo el contenido del elemento de búsqueda, la aplicación manda el valor de vacío al elemento. En caso de que el usuario hubiese terminado de introducir todas las letras del contenido de su búsqueda y diga el comando “SEARCH”, se ejecuta el siguiente archivo del proceso contenido en la aplicación VD.

Una vez ha concluido el desarrollo del código VoiceXML del formulario “nav”, se ejecuta la sincronización de la variable que almacena el comando ejecutado del usuario con el elemento visual del formulario contenido en la siguiente parte de la estructura.

La tercera y última parte de este archivo incluye los eventos incrustados en sus etiquetas correspondientes. La etiqueta <body> contiene el evento que lanza el formulario de voz “nav” al cargar esta etiqueta, y la etiqueta que corresponde al contenedor lanza el diálogo cuando se hace click sobre este. El resto de contenidos de esta parte hacen referencia a todos los elementos visuales que podemos ver cuando cargamos la página, incluyendo la función que oculta o muestra las instrucciones en función de su estado anterior.

El archivo correspondiente al proceso de extracción y limpieza del contenido de la búsqueda en Wikipedia es el ya mencionado “*dictionary\_process.php*”. El código del archivo realiza las siguientes tareas:

1. Recoge el valor de búsqueda introducido por el usuario de forma manual u oral en el *input* correspondiente del formulario.
2. A través de funciones predefinidas, cambiamos los espacios entre las palabras por guiones bajos, y transformamos cada carácter del contenido de la búsqueda a minúsculas, dejando las letras iniciales como mayúsculas. Toda esta información transformada la guardamos en una variable, dejando el contenido como valor válido para el entorno de Wikipedia.
3. Concatenamos la cadena de caracteres “*http://en.wikipedia.org/wiki/*” con la variable anterior y lo dejamos almacenado como una dirección URL.
4. Lanzamos la función “*file\_get\_contents()*” sobre la URL almacenada en la variable anterior para capturar el contenido de esa dirección y almacenarlo como una cadena de texto en una nueva variable.
5. Si la variable donde almacenamos la página de Wikipedia da error, capturamos el contenido de la plantilla almacenada en el archivo “*rdictionary.xhtml*” y mostramos un error al usuario para esa búsqueda, además de narrárselo. En caso de que no se produzca ningún error, procedemos a depurar esa variable para seleccionar el contenido válido.

6. Esta limpieza o depuración de la variable se realiza eliminando la información anterior al contenido de la búsqueda, y la información posterior. A continuación, sobre el contenido útil, eliminamos todas las etiquetas y caracteres conflictivos, excepto las etiquetas de vínculos que favorecen el hipertexto.
7. Una vez limpiado el código, seleccionamos todos los enlaces del contenido y los guardamos en una lista.
8. Después, incrustamos el contenido útil en el contenedor de la plantilla de resultado, y uno a uno vamos seleccionando todos y cada uno de los enlaces de la lista previamente almacenada y los incrustamos en la gramática y diálogo de la plantilla de resultados para poder dar la funcionalidad de hipertexto al usuario.
9. Con toda esta información de la plantilla de resultado en una variable, procedemos a extraerla en forma de archivo ("*rdictionary2.xhtml*") a través de la función predefinida "*file\_put\_contents()*".
10. Una vez creado el archivo final de resultado con el contenido de la búsqueda incrustado tanto en su interfaz oral como en su interfaz visual, procedemos a cambiar la cabecera del navegador para redirigir al usuario a esta página de resultado.

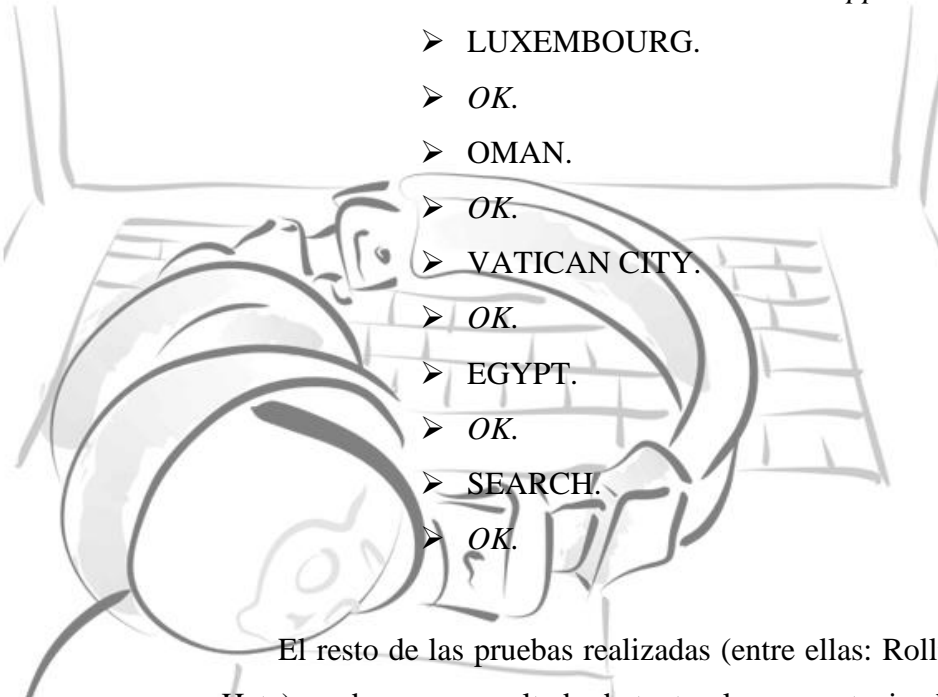
La página de resultado presenta un estilo idéntico a la página de búsqueda, a excepción de la nueva funcionalidad oral. En la zona de codificación del código correspondiente a la interfaz oral, insertamos la gramática en lugar de ubicar una referencia a un archivo externo. Esto permite, desde el archivo de proceso, crear una gramática diferente en función de la búsqueda realizada por el usuario, permitiendo además la visita a los enlaces generados por el contenido resultante mediante la voz, llamando de nuevo al archivo de proceso y volviendo a generar un resultado con el nuevo contenido. Esta funcionalidad permite una aplicación en constante cambio, aparentemente viva.

### 3.2.4 Pruebas y limitación de errores

Numerosas han sido las pruebas realizadas a esta aplicación hasta lograr un punto de funcionalidad que explota las capacidades de XHTML+Voice. Los errores encontrados se confunden con los errores de pronunciación de la lengua inglesa del usuario escogido, favoreciendo la mejor elección en la creación de la lista de países en aquellos más complicados. Si bien es verdad que existen ocasiones en las que el elevado número de palabras de una gramática y la libertad dada a la aplicación para crearlas, genera incongruencias haciendo coincidir palabras de pronunciación similar a las palabras reservadas por el navegador, o por la propia aplicación. Se han limitado estas incongruencias al máximo de forma que las posibles concordancias entre las palabras seleccionadas se ven prácticamente eliminadas.

A continuación podemos observar el diálogo obtenido de una de las pruebas, cuyo resultado final se omite debido a la longitud del texto obtenido.

#### PRUEBA DE DIÁLOGO ORAL:

- 
- *Welcome to Voice Dictionary. What do you want to search? If you do not know how to use this application, say: "INSTRUCTIONS".*
  - LUXEMBOURG.
  - OK.
  - OMAN.
  - OK.
  - VATICAN CITY.
  - OK.
  - EGYPT.
  - OK.
  - SEARCH.
  - OK.

El resto de las pruebas realizadas (entre ellas: Rolling Stones, The Beatles, Love, Hate) producen un resultado de texto plano, manteniendo los vínculos y algunas de las imágenes, similar. Con los resultados en pantalla de alguno de los ejemplos (Rolling

Stones, The Beatles) se ha comprobado la conectividad de los vínculos y su resultado en este entorno, llegando a una profundidad de diez hipervínculos.

La mayor dificultad encontrada a la hora de desarrollar VD ha sido poder extraer y limpiar el contenido de la búsqueda correctamente. Acceder a un entorno ajeno a la aplicación, y eliminar toda la información innecesaria para quedarnos con el contenido buscado no ha sido tarea fácil, pero se logró programando el código correcto. Por otro lado, la sombra de un servidor frágil favorece la aparición de errores como los encontrados en el área de redireccionamiento de URLs. La función predefinida de PHP “header()”, se ha visto más de una vez truncada por el envío de más información a la cabecera de la que pueda soportar. La configuración y código complementario sobre el envío de información a las cabeceras ha solucionado estos inconvenientes.

### 3.3 Voice Pronunciation

A medio camino entre una aplicación y un divertido juego, presentamos *Voice Pronunciation* (VP). La idea original de esta aplicación surgió de la necesidad de un mayor conocimiento de la pronunciación de la lengua inglesa, y de un vocabulario más extenso. Si bien es cierto que la composición de todo el proyecto en esta lengua favorece la internacionalidad de la aplicación y su uso en otros países, además de un desarrollo realizado con el mayor cuidado para facilitar al máximo la tarea del usuario, a veces resulta complicado pronunciar correctamente en esta lengua, y se incrementan las posibilidades de error en la interpretación de los comandos por parte de la interfaz oral, al igual que la comunicación entre el usuario y la máquina. Con el uso de esta aplicación se entrenará y mejorará notablemente la pronunciación y la adquisición de nuevo vocabulario de una manera fácil, y lo más importante: divertida.

Una vez accedemos a la aplicación desde cualquier lugar del entorno VA mediante el comando “*PRONUNCIATION*”, el usuario se encuentra con una bienvenida a través de discurso oral, y la posibilidad de elegir entre las diferentes



opciones: *Words*, *Pictures* e *Instructions*. Cada una de estas opciones representa una sección independiente de esta aplicación. Al utilizar el comando “*INSTRUCTIONS*” accedemos a la sección dónde se describen todas las instrucciones para utilizar esta aplicación. Veremos esta sección más detalladamente en el apartado de Funcionamiento.

Las secciones de la aplicación VP denominadas *Words* y *Pictures* representan los juegos de pronunciación, uno desarrollado para un uso más dirigido a la interfaz oral y el otro más encaminado a la interfaz visual, aunque ambos juegos realizan su interacción con el usuario en ambas interfaces. En el primero se muestra una de las más de cien mil palabras almacenadas en la base de datos y a continuación la definición de la palabra, además de una puntuación máxima conseguida por un jugador anterior junto a la puntuación del usuario. El usuario deberá pronunciar correctamente la palabra mostrada. La interfaz oral sólo narra la descripción de la palabra. Mientras que en la sección *Pictures*, en lugar de mostrar una palabra, se muestra una imagen elegida de entre las diversas dificultades. El usuario deberá averiguar el nombre exacto de la imagen mostrada y pronunciarlo correctamente para continuar y aumentar la puntuación.

### 3.3.1 Funcionamiento

El usuario puede interactuar con esta aplicación de manera muy sencilla, sin apenas instrucciones de uso. Al acceder al inicio de la aplicación se encuentra con la posibilidad de elegir entre los dos juegos y las instrucciones. Puede, al igual que en el resto del entorno, decir el comando de cualquiera de las aplicaciones, en cualquier momento para viajar a esa aplicación. Para acceder a los juegos sólo tiene que decir los comandos “*WORDS*” o “*PICTURES*”. Si desea acceder a las instrucciones deberá decir el comando “*INSTRUCTIONS*”. Cuando el usuario accede a las instrucciones se muestra y narra el siguiente contenido:

- *Welcome to the Instructions of Voice Pronunciation.*
- *If you want to play the game of Words, you must to say: "WORDS".*
- *Then, I tell you the description of the word you must to say.*

- *If you say the word with a perfect pronunciation, I give you ten points.*
- *If you don't know how to pronounce this word, say: "SOLUTION".*
- *If you want to play the next word, you must to pronounce the word correctly or you can say: "NEXT".*
- *If you say next, I don't give you any point.*
- *I show you your punctuation and the maximum punctuation all the time.*
- *If you want to play the game of Pictures, you must to say: "PICTURES".*
- *Then, I tell you the description of the picture you must to watch in the screen.*
- *If you guess the name of the picture with a perfect pronunciation, I give you ten points.*
- *If you don't know how to pronounce the name of the picture, say: "SOLUTION".*
- *If you want to play the next picture, you must to pronounce the name of the picture correctly or you can say: "NEXT".*
- *If you say next, I don't give you any point.*
- *I show you your punctuation and the maximum punctuation all the time.*
- *If you want to go out the games, you must to say: "EXIT".*

Con estas sencillas instrucciones el usuario puede jugar en esta aplicación y aumentar su vocabulario en lengua inglesa, además de mejorar considerablemente su pronunciación.

### 3.3.2 Diseño

La presentación de la aplicación se muestra inicialmente como un menú de elección donde podemos escoger, de manera oral o manual, cualquiera de las secciones apuntadas por el menú. Desde esta pantalla inicial accedemos a las

instrucciones, al juego de palabras o al juego de imágenes. En la Figura 19 podemos ver esta pantalla inicial de la aplicación VP. El diseño de la interfaz oral de esta página se corresponde con un mensaje de bienvenida y una breve descripción de los comandos a usar en esta pantalla.

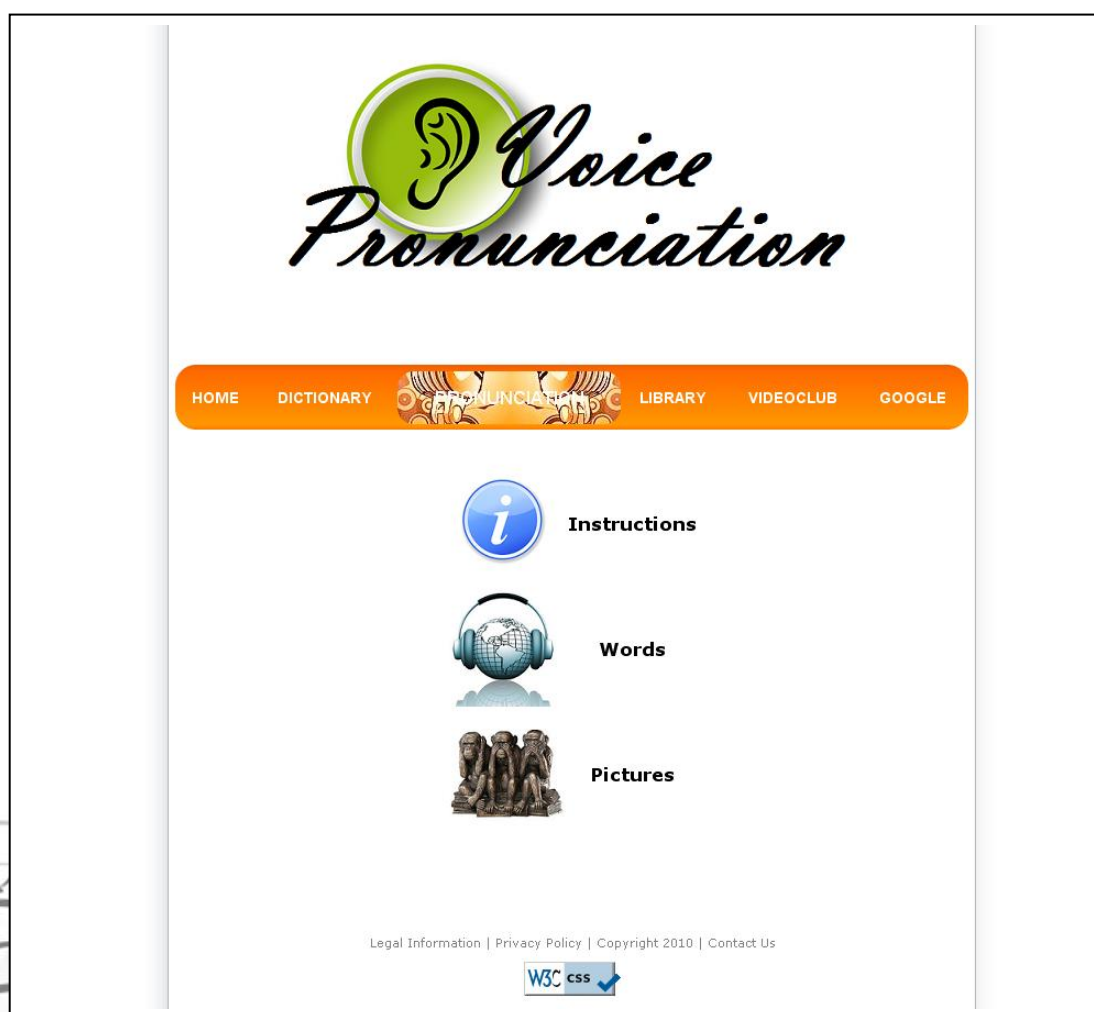


Figura 19. Diseño de pantalla inicial de VP.

Manteniendo la temática de colores y diseños de todo el entorno, podemos ver cómo en la imagen corporativa de esta aplicación se ha seleccionado un símbolo auditivo representando tanto la multimodalidad como la pronunciación en un tono verde tras las letras características del entorno, encajando a la perfección con el resto

de la página. Para el menú de elección de secciones se ha elegido una imagen representativa para cada sección, favoreciendo la representación visual de los juegos. En la Figura 20 observamos cómo se ha elegido un color diferente para cada frase de las instrucciones, dándoles un aspecto divertido afín a los juegos que describe. En relación con el diseño de la interfaz oral se narra completamente todas y cada una de las instrucciones para equilibrar aspectos visuales y orales.



**Voice Pronunciation**

HOME DICTIONARY PRONUNCIACIÓN LIBRARY VIDEOCLUB GOOGLE

**i** Instructions

**Welcome to the Instructions of Voice Pronunciation.**

**If you want to play the game of Words, you must to say: "WORDS".**  
Then, I tell you the description of the word you must to say.  
**If you say the word with a perfect pronunciation, I give you ten points.**  
**If you don't know how to pronounce this word, say: "SOLUTION".**  
**If you want to play the next word, you must to pronounce the word correctly or you can say: "NEXT".**  
**If you say next, I don't give you any point.**  
**I show you your puntuation and the maximum puntuation all the time.**

**If you want to play the game of Pictures, you must to say: "PICTURES".**  
Then, I tell you the description of the picture you must to wath in the screen.  
**If you guess the name of the picture with a perfect pronunciation, I give you ten points.**  
If you don't know how to pronounce the name of the picture, say: "SOLUTION".  
**If you want to play the next picture, you must to pronounce the name of the picture correctly or you can say: "NEXT".**  
**If you say next, I don't give you any point.**  
**I show you your puntuation and the maximum puntuation all the time.**  
**If you want to go out the games, you must to say: "EXIT".**

Go Back!

Legal Information | Privacy Policy | Copyright 2010 | Contact Us

W3C CSS

Figura 20. Diseño de sección Instructions de VP.

Avanzamos y vemos en la Figura 21 el diseño empleado para desarrollar el juego *Words* de la aplicación VP. Junto al título del juego se presenta la imagen que lo representa, y bajo este la palabra a pronunciar y la definición de esta. En el aspecto oral, se narra la definición dando al usuario las pistas suficientes para averiguar la palabra y pronunciarla correctamente. En la esquina superior derecha podemos ver como el marcador del usuario va aumentando su puntuación a medida que pronuncia las palabras correctamente, además de la puntuación máxima conseguida en este juego con anterioridad.

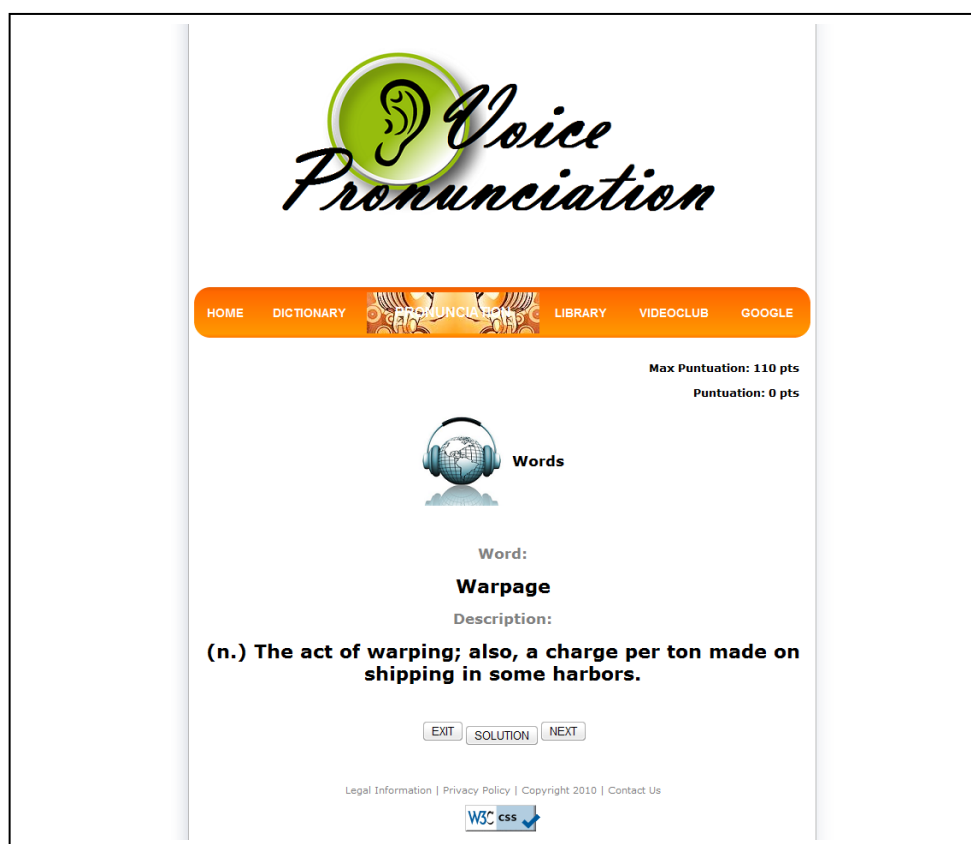


Figura 21. Diseño de juego *Words* de VP.

A continuación, en la Figura 22 se presenta el diseño del juego denominado como *Pictures*. Al igual que en el juego anterior, se presenta el título junto a la imagen corporativa que lo representa. Tras el título se muestra la imagen y su descripción, siendo esta descripción narrada por la interfaz oral. Una vez que el usuario acierta el nombre correcto de la imagen y lo pronuncia adecuadamente, se añaden diez puntos visualizados en la esquina superior izquierda del cuerpo de la página, tras los cuales se pueden ver los puntos totales conseguidos con anterioridad.



Figura 22. Diseño de juego Pictures de VP.

### 3.3.3 Arquitectura

Accedemos a la aplicación a través del archivo “*spronunciation.xhtml*”, cuya construcción es similar al archivo principal de la aplicación anterior. Dividido en tres secciones (Información, cabecera con código oral, y cuerpo con código visual), alberga un submenú compuesto por enlaces e imágenes representativas. Desde aquí accedemos a las instrucciones, al juego de las palabras, y al juego de las imágenes. Podemos ver el flujo de información en la Figura 23.

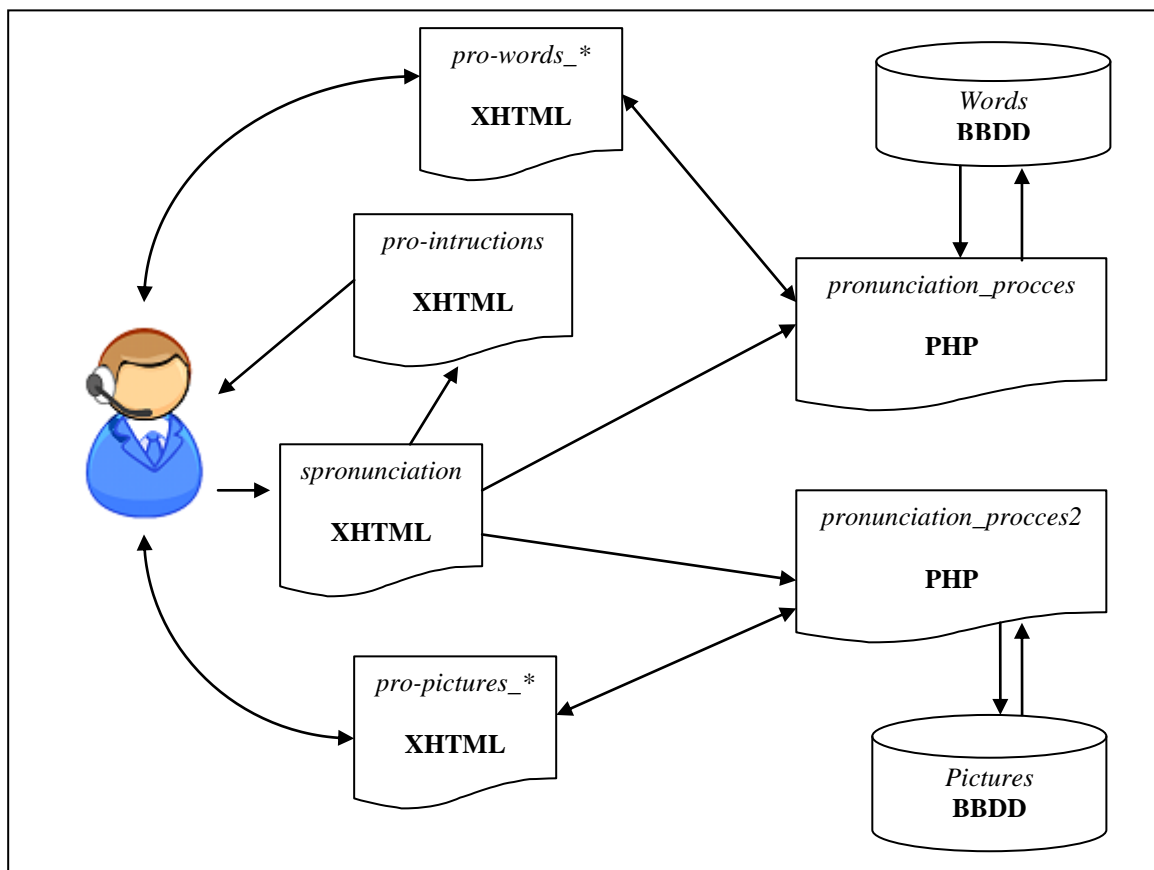


Figura 23. Flujo de información de la aplicación VP.



Aunque a priori la figura anterior parezca un poco confusa o enrevesada, el flujo de información recorre un camino muy sencillo. Inicialmente el usuario accede al archivo anteriormente mencionado, “*spronunciation.xhtml*”. Desde este archivo el usuario puede abrir las instrucciones de la aplicación o los juegos de palabras o imágenes. Al acceder a las instrucciones la información pasa al archivo “*pro-intructions.xhtml*”, retornando posteriormente al usuario de forma tanto visual como oral. La estructura de los archivos en formato XHTML, entre los que se incluyen los codificados en XHTML+Voice, es prácticamente similar en todo el entorno, modificando únicamente el contenido del cuerpo de la página, o información concreta de la aplicación, y los diálogos de las interfaces orales, narrando discursos acordes a las presentaciones.

Al iniciar el usuario el juego de las palabras, la información se transmite al archivo denominado como “*pronunciation\_procces.php*”. Éste accede a la base de datos de palabras a través de un proceso aleatorio, y selecciona una de las más de cien mil palabras almacenadas junto con sus definiciones correspondientes. Una vez seleccionada la palabra, genera un archivo incrustando la palabra en el nombre del archivo, de tal manera que queda como: “*pro-word\_[PALABRA].xhtml*”. Este archivo es el que se muestra al usuario con toda la información. En caso de que el usuario acceda a la siguiente palabra (bien sea porque acierta la palabra y su pronunciación, porque selecciona ir a la siguiente, o porque selecciona salir del juego), este archivo se elimina, favoreciendo la liberación del espacio en disco. Si el usuario continúa con el proceso del juego se accede nuevamente al archivo “*pronunciation\_process.php*” y se pone en movimiento de nuevo todo el proceso de creación. Si el usuario inicia el juego de las imágenes, el proceso es el mismo, salvo que el archivo que realiza las instrucciones de acceso a la base de datos se denomina “*pronunciation\_process2.php*”, y la base de datos trata con más de seiscientos imágenes almacenadas previamente junto con sus descripciones.

Además de los archivos mencionados, se utiliza un archivo denominado “*random.php*”, encargado del algoritmo de acceso aleatorio a las bases de datos, y archivos externos para almacenar las puntuaciones. La puntuación máxima se almacena en un archivo común a todos los usuarios que deseen acceder a la aplicación VP. En cambio, para evitar errores de acceso a los archivos en caso de un elevado número de usuarios activos en la aplicación, la puntuación de cada usuario se almacena en un archivo independiente creado a partir de la IP del usuario. Este archivo es eliminado de la aplicación cuando el usuario la abandona.

Tanto el archivo “*pronunciation\_process.php*” como el archivo denominado como “*pronunciation\_process2.php*”, realizan el mismo proceso, salvo por la base de datos a la que acceden y la maquetación del archivo que generan y muestran al usuario. Abren el proceso aleatorio del archivo externo, acceden a la base de datos, extraen la información, procesan el archivo de resultado, generan la puntuación en archivos independientes, y muestran el resultado. En caso de finalizar la aplicación se borran los archivos de resultados actuales y se eliminan los archivos de puntuaciones individuales de usuarios.

Como vimos al inicio de este documento en las Figuras 7 y 8, las bases de datos de palabras y libros contienen una estructura de tabla similar. En la tabla denominada como “*Words*” podemos ver un número asignado unívocamente a cada palabra, el nombre de la palabra, y su definición. En la tabla denominada “*Pictures*” nos encontramos con el número de identificación única, el nombre, la descripción, y el tipo, diferenciando entre los siguientes tipos: *Animals, Cartoon Characters, Colours, Fruits, Office Items, Places and Buildings, Professions, Sports, Tools, y Vegetables*.

Las bases de datos tienen todos los privilegios asignados de forma que permitan el acceso, no a los usuarios que ejecuten la aplicación, sino a los archivos codificados en PHP que extraen la información para incrustarla en los archivos que genera como resultado. La base de datos propia de esta aplicación se denomina “*Pronunciation*”, y contiene las tablas del juego de las palabras y del juego de las imágenes.

### 3.3.4 Pruebas y limitación de errores

Las pruebas realizadas en esta aplicación llegan a los diez aciertos encadenados en el juego de las imágenes, y a once en el juego de las palabras. Se ha tenido en cuenta la introducción de palabras erróneas, intentando provocar la ruptura de la aplicación sin éxito. Incluso se ha probado a acceder a otras aplicaciones desde cualquier punto de los juegos, consiguiéndolo tanto a nivel visual como a nivel oral. Las gramáticas, y bases de datos limitan la aplicación de tal manera que la posibilidad de error por parte de la aplicación se reduce a cero.

## 3.4 Voice Library & Voice Videoclub

Describimos las aplicaciones *Voice Library* (VL) y *Voice Videoclub* (VV) juntas debido a su gran similitud en muchos de los aspectos que destacamos en este documento. La página principal de ambas páginas se asemeja a la misma página en la aplicación VD. Esta primera página se compone esencialmente de su imagen corporativa correspondiente junto a las letras propias del entorno VA. Bajo el menú nos encontramos con la barra de buscador en la que podemos introducir el libro en el caso de VL o la película en el caso de VV que se desee buscar. Estas búsquedas se realizan en las bases de datos creadas con más de trescientos libros y entorno a novecientas películas. En la Figura 24 podemos ver la página principal de VL, y en la Figura 25 podemos ver la página principal de VV.

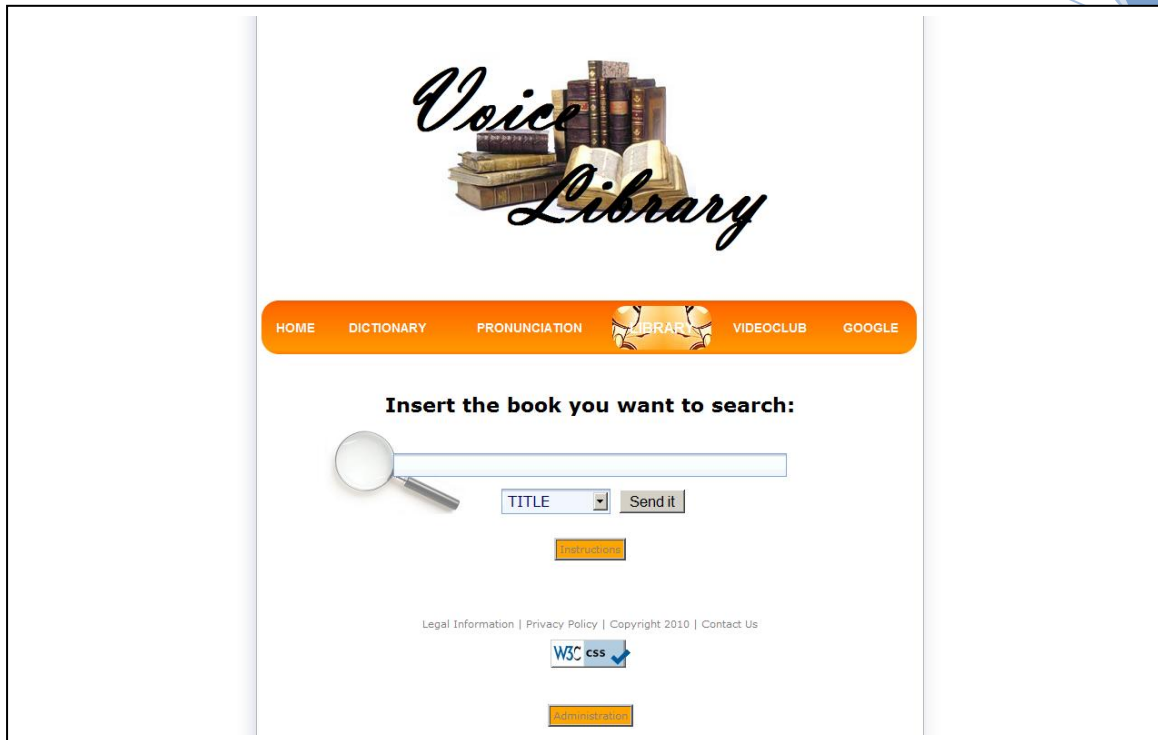


Figura 24. Página principal de VL.

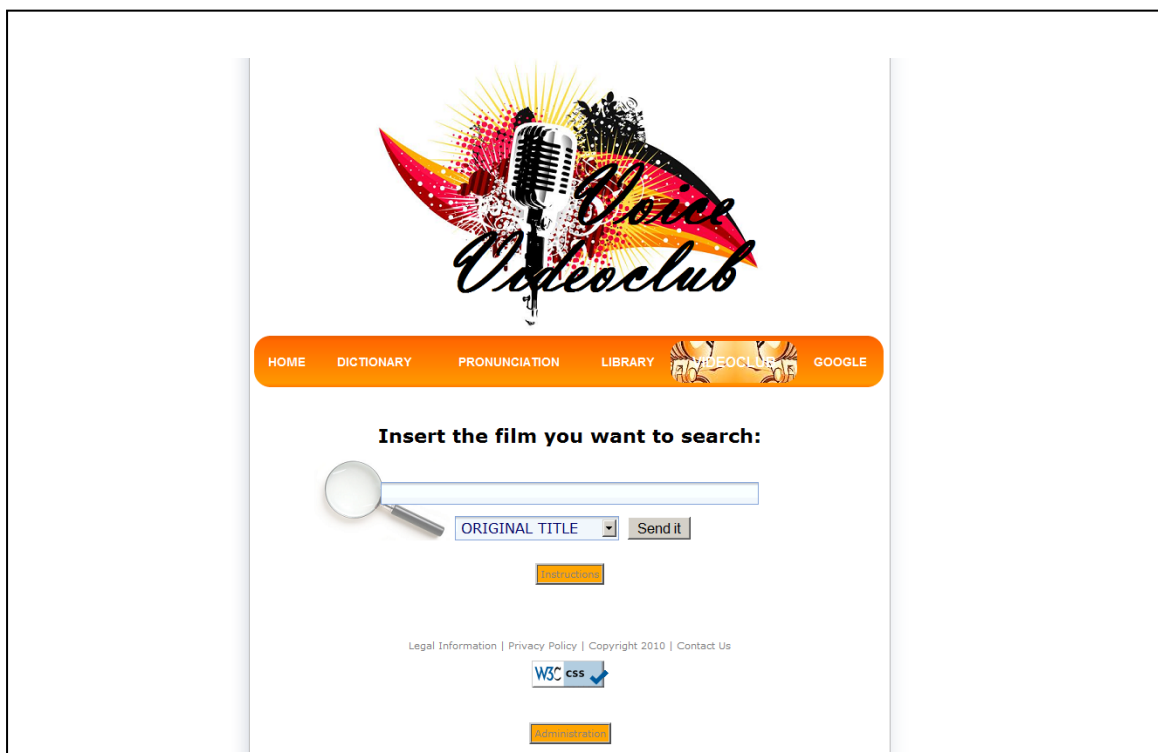


Figura 25. Página principal de VV.

La gran diferencia con aplicaciones anteriores es la incorporación de un objeto bajo la barra de búsqueda que facilita la tarea de búsqueda diciéndole a la aplicación que tipo de búsqueda desea realizar. En el caso de VL, como vemos en la imagen, la búsqueda se realiza por el título del libro, y en el caso de VV se realiza por el título original de la película. Sin embargo, como veremos más adelante, podemos elegir el tipo de búsqueda que queremos realizar. Además, se ha incorporado una nueva funcionalidad, a la que le dedicaremos un apartado exclusivo al final del desarrollo de estas aplicaciones denominado Gestión privada de bases de datos.

### 3.4.1 Funcionamiento

Al tratarse de aplicaciones de búsqueda, emplean el mismo método de funcionamiento que la aplicación VD. Para introducir el contenido de la búsqueda de manera oral el usuario debe deletrearlo a expresando el país correspondiente a la letra a insertar. Bajo la barra de búsqueda se pueden desplegar las instrucciones, además de ser narradas gracias al comando “*INSTRUCTIONS*”.

El usuario puede, además, decir en cualquier momento del deletreo del contenido de la búsqueda el tipo de contenido que está introduciendo. Para ello, puede seleccionarlo de entre las opciones desplegadas bajo la barra de búsqueda, o decirlo a la interfaz oral de manera natural, sin necesidad de deletreo. Los tipos de búsquedas para ambas aplicaciones son:

*Voice Library:*

- *Title.*
- *Author.*
- *ISBN*
- *Editorial.*

*Voice Videoclub:*

- *Original Title.*
- *Translated Title.*
- *Director.*
- *Category.*

El tipo puede ser mencionado por el usuario en cualquier momento y la aplicación mostrará esa elección en la barra desplegable. Los comandos para narrar estos tipos son los propios tipos, favoreciendo la interacción natural entre el usuario y la máquina.

Una vez mostrado el contenido de la búsqueda, el usuario puede volver a realizar una nueva búsqueda a través del comando “*SEARCH AGAIN*” en la interfaz oral, o haciendo click en este comando mostrado al final de los contenidos resultantes.

### 3.4.2 Diseño

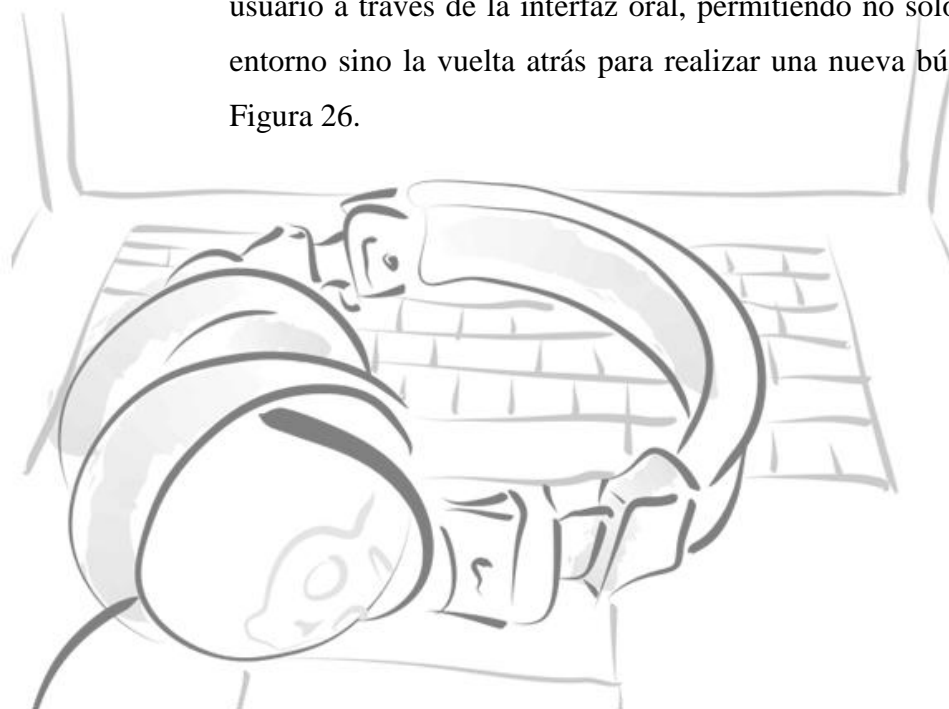
Como se ha podido ver en la Figuras 24 y 25, los diseños de las páginas principales de búsqueda de las aplicaciones VL y VV son similares al ya mostrado en VD, con la única salvedad de la barra desplegable de los tipos, y el botón al final de la página donde pone “*Administration*”.

Una vez que el resultado del contenido de la búsqueda insertado por el usuario es mostrado, la aplicación VL nos muestra el número total de libros encontrados, y una tabla con todos los libros, donde cada columna representa una información o tipo (*ISBN, Title, Author, Editorial, Description, Edition, Link*), y cada fila representa un libro. La tabla se muestra con los colores representativos del entorno, y una imagen como fondo de tabla, dando el aspecto visual de una librería, o estantería repleta de libros.

La aplicación VV tiene organizada la información resultante de la búsqueda de una manera totalmente diferente, siendo cada película la que se va mostrando en una lista, con una imagen junto al número de la película. La lista muestra a la izquierda los tipos de contenidos que se muestran a la derecha. En el apartado correspondiente a las Pruebas realizadas sobre ambas aplicaciones se mostrarán figuras que ilustren los diseños de contenidos resultantes para las aplicaciones VL y VV.

### 3.4.3 Arquitectura

La construcción de las aplicaciones VL y VV se caracteriza por una interfaz inicial de búsqueda de contenidos que envía la información del usuario a la base de datos correspondiente a través de los algoritmos encapsulados en los archivos de proceso, y muestra al final una interfaz de resultado que narra los contenidos al usuario a través de la interfaz oral, permitiendo no sólo la navegabilidad a través del entorno sino la vuelta atrás para realizar una nueva búsqueda, tal y como muestra la Figura 26.



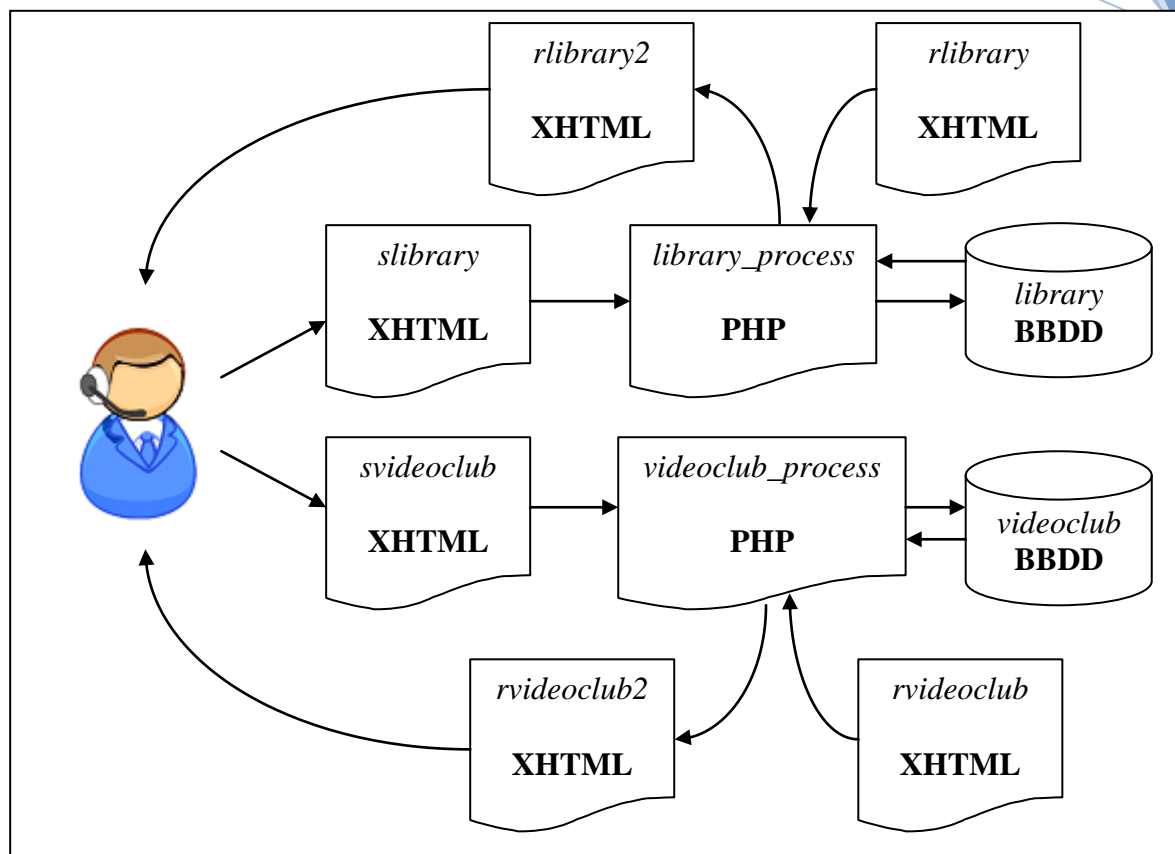


Figura 26. Flujo de información de las aplicaciones VL y VV.

El usuario accede a la aplicación VL abriendo la página correspondiente al archivo “*slibrary.xhtml*”. El código representado en este archivo hace referencia a una gramática en su interfaz oral que favorece la interacción en tres aspectos:

- i. El diálogo correspondiente a la navegabilidad del entorno desde esta aplicación.
- ii. Los comandos de deletreo para insertar el contenido de la búsqueda en el elemento visual de búsqueda.
- iii. Los comandos correspondientes a los tipos de búsquedas visualizados en la barra desplegable bajo el elemento visual de búsqueda.

Esta combinación de la gramática junto a los formularios de diálogo representados en el código, invita al usuario a interactuar con más libertad.



Una vez que el usuario inserta la información que desea buscar, este contenido pasa al archivo de proceso denominado “*library\_process.php*”. Este archivo abre la base de datos “*Library*”, accede a la tabla “*Books*”, y recupera la información a través de una consulta de MySQL. Una vez que tiene esa información, genera una tabla con tantas filas como filas haya extraído de la base de datos, y la incrusta en la plantilla “*rlibrary.xhtml*” creando el archivo de resultado “*rlibrary2.xhtml*”. Por último, redirecciona a este archivo de resultados y lo muestra al usuario.

Para la aplicación VV, la secuencia de archivos y procesos es básicamente la misma. Se inicia en “*svideoclub.xhtml*”, que envía la información a “*videoclub\_process.php*”, el cual accede a la base de datos “*Videoclub*” en su tabla “*films*”. Extra la información y la incrusta en la plantilla “*rvideoclub.xhtml*” generando el archivo de resultado “*rvideoclub2.xhtml*”. En esta aplicación, los diálogos de los archivos de inicio de la búsqueda responden también a los tres aspectos de interacción mencionados anteriormente para la aplicación VL.

### 3.4.4 Pruebas y limitación de errores

Las pruebas realizadas se corresponden a búsquedas de información cambiándolas con los diferentes tipos mostrados en la barra desplegable. Las aplicaciones han mostrado un mínimo margen de error a la hora de limitarlos, mostrándose únicamente débiles cuando se realizan repetidas consultas a la misma base de datos. Este error se atribuye a las interacciones entre Opera, el servidor de documentos Apache, y la consola de bases de datos MySQL. Cuando esto ocurre, la aplicación no accede a la base de datos y genera siempre el mismo resultado para cualquier búsqueda. Para solucionarlo se debe reiniciar Opera, o realizar una consulta desde otra aplicación a otra base de datos para formatear la conexión. Aunque la aparición de este error sea prácticamente nula, se ha investigado el origen de dicha problemática sin hallar mayor

respuesta que la de un flujo excesivo de información entre los operadores antes mencionados.

Un error en el disco duro dónde se almacenaba toda la información del presente proyecto propició la pérdida de todos los datos. Este error inhabilitó el disco duro para su uso y hubo que comenzar a restablecer todos los contenidos desde una copia de seguridad de una versión anterior. Este inconveniente no supuso mayores problemas, pues las bases de datos tenían sus correspondientes copias de seguridad, sin embargo a la hora de restituir esas copias de seguridad se encontró que los archivos estaban corruptos. Fue necesario diseñar un algoritmo en Pascal que accediese a las tablas de los archivos .MYD y fuese extrayendo todos y cada uno de los campos restringidos por los correspondientes delimitadores de campo (caracteres ASCII ajenos al contenido que marcan el comienzo y fin de la información relevante para su optimización de espacio en disco). Al recupera todas las bases de datos, se optimizó la exportación de resultados generando copias de seguridad solventes. La aplicación VV, a raíz de este bache en el camino, se vio notablemente mejorada al ver su base de datos incrementada en más de trescientas películas nuevas, con nuevos campos de descripción e imagen donde se ofrece el cartel de la película buscada.

A continuación mostramos tres pruebas realizadas con los diferentes tipos de tanto de la aplicación VL como de la aplicación VV, describiendo el diálogo con la interfaz oral (adjudicando el formato cursivo a la narración por parte de la interfaz oral), y a continuación una imagen con el resultado obtenido (Figuras de la 27 a la 32, ambas incluidas):

#### PRUEBA #1:

- *Welcome to Voice Library. What is your favorite book? If you do not know how to use this application, say: "INSTRUCTIONS".*
- AUSTRALIA.
- *Ok.*
- NEW ZEALAND.
- *Ok.*

- IRELAND.
- *Ok.*
- LUXEMBOURG.
- *Ok.*
- LUXEMBOURG.
- *Ok.*
- OMAN.
- *Ok.*
- SPAIN.
- *Ok.*
- SEARCH.
- *Ok.*
- *Results for TITLE "ANILLOS" (#3):*
  - *BOOK #1:*
    - *ISBN: 281.*
    - *TITLE: El Señor de los Anillos I.*
    - *AUTHOR: Tolkien, John Ronald Reue.*
    - *EDITORIAL: Minotauro.*
    - *DESCRIPTION: no.*
    - *EDITION: 29.*
    - *LINK: no...*
  - *BOOK #2:*
    - *ISBN: 280.*
    - *TITLE: El Señor de los Anillos II.*
    - *AUTHOR: Tolkien, John Ronald Reue.*
    - *EDITORIAL: Minotauro.*
    - *DESCRIPTION: no.*
    - *EDITION: 25.*
    - *LINK: no...*
  - *BOOK #3:*
    - *ISBN: 279.*
    - *TITLE: El Señor de los Anillos III.*
    - *AUTHOR: Tolkien, John Ronald Reue.*
    - *EDITORIAL: Minotauro.*

- *DESCRIPTION: no.*
- *EDITION: 23.*
- *LINK: no...*

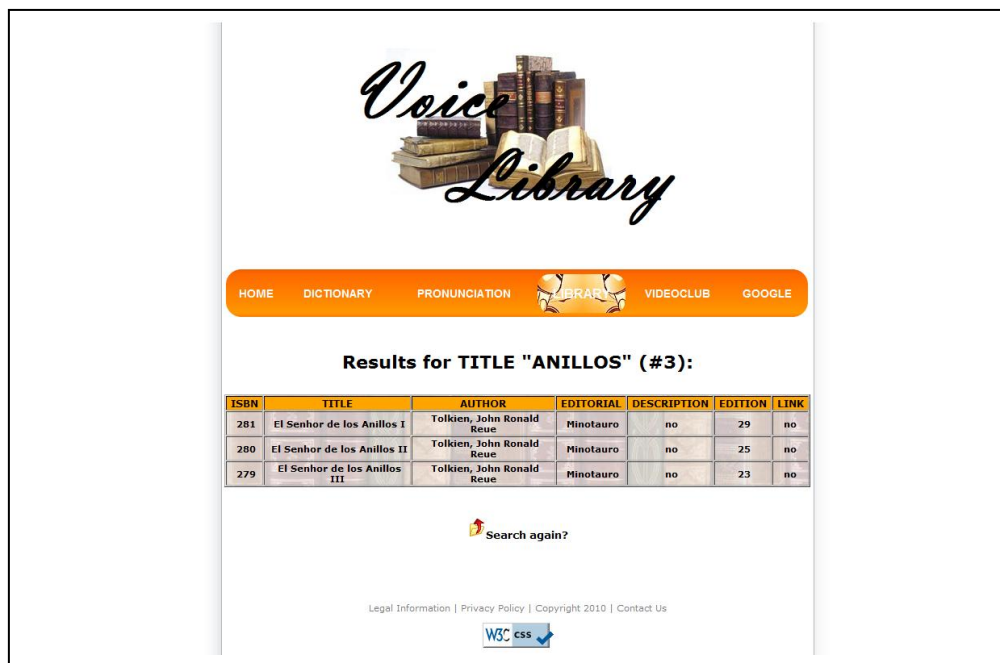


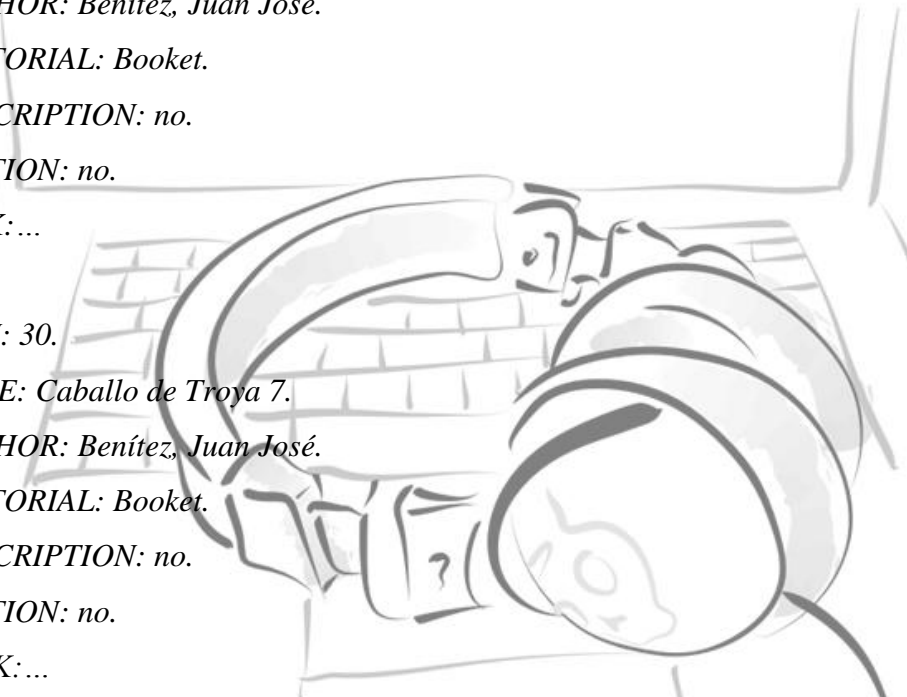
Figura 27. Resultados de PRUEBA #1 sobre VL.

## PRUEBA #2:

- *Welcome to Voice Library. What is your favorite book? If you do not know how to use this application, say: "INSTRUCTIONS".*
- BURKINA FASO.
- *Ok.*
- EGYPT.
- *Ok.*
- NEW ZEALAND.
- *Ok.*
- IRELAND.
- *Ok.*
- TURKEY.
- *Ok.*

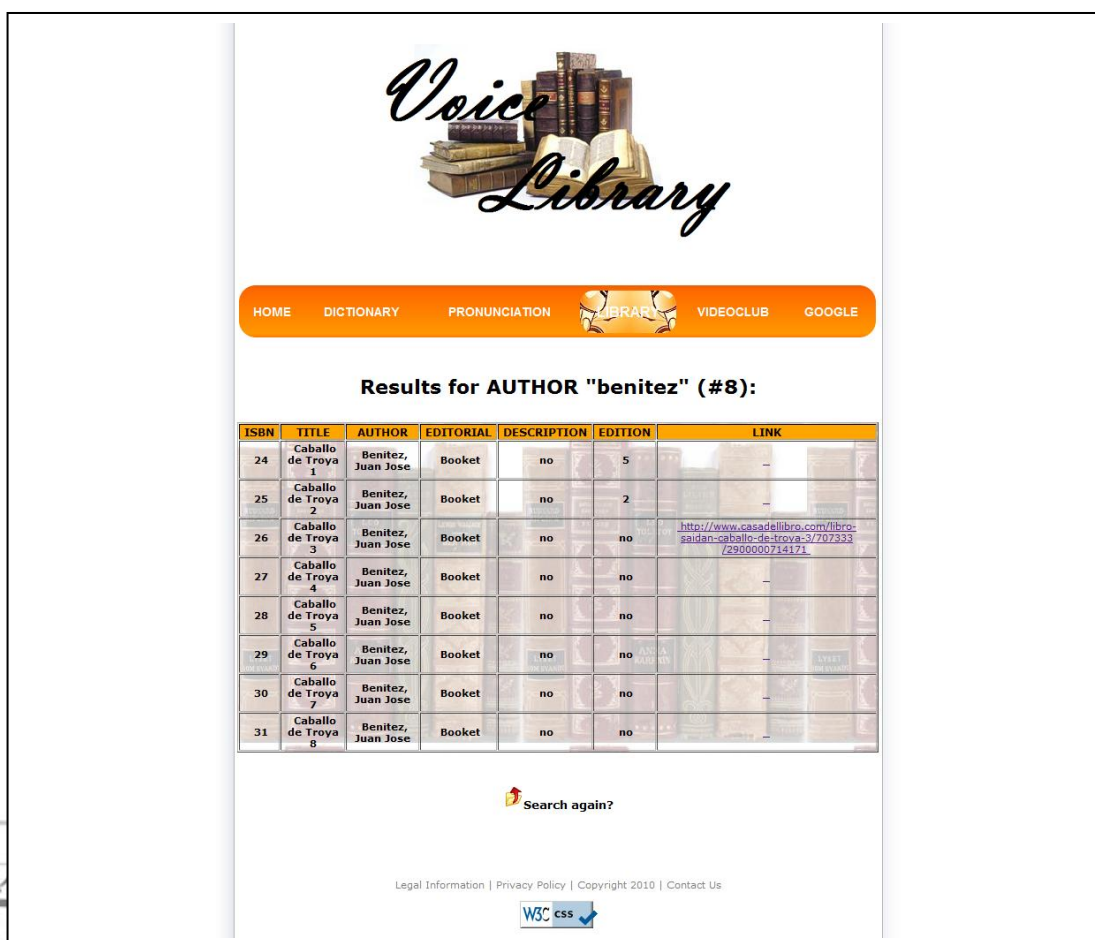
- EGYPT.
- *Ok.*
- ZIMBABWE.
- *Ok.*
- AUTHOR
- *Ok.*
- SEARCH.
- *Ok.*
- *Results for AUTHOR "benítez" (#8):*
  - *BOOK #1:*
  - *ISBN: 24.*
  - *TITLE: Caballo de Troya 1.*
  - *AUTHOR: Benítez, Juan José.*
  - *EDITORIAL: Booket.*
  - *DESCRIPTION: no.*
  - *EDITION: 5.*
  - *LINK:...*
  - *BOOK #2:*
    - *ISBN: 25.*
    - *TITLE: Caballo de Troya 2.*
    - *AUTHOR: Benítez, Juan José.*
    - *EDITORIAL: Booket.*
    - *DESCRIPTION: no.*
    - *EDITION: 2.*
    - *LINK:...*
    - *BOOK #3:*
      - *ISBN: 26.*
      - *TITLE: Caballo de Troya 3.*
      - *AUTHOR: Benítez, Juan José.*
      - *EDITORIAL: Booket.*
      - *DESCRIPTION: no.*
      - *EDITION: no.*
      - *LINK: <http://www.casadellibro.com/libro-saidan-caballo-de-troya-3/707333/2900000714171>.*

- **BOOK #4:**
  - *ISBN: 27.*
  - *TITLE: Caballo de Troya 4.*
  - *AUTHOR: Benítez, Juan José.*
  - *EDITORIAL: Booket.*
  - *DESCRIPTION: no.*
  - *EDITION: no.*
  - *LINK:...*
- **BOOK #5:**
  - *ISBN: 28.*
  - *TITLE: Caballo de Troya 5.*
  - *AUTHOR: Benítez, Juan José.*
  - *EDITORIAL: Booket.*
  - *DESCRIPTION: no.*
  - *EDITION: no.*
  - *LINK:...*
- **BOOK #6:**
  - *ISBN: 29.*
  - *TITLE: Caballo de Troya 6.*
  - *AUTHOR: Benítez, Juan José.*
  - *EDITORIAL: Booket.*
  - *DESCRIPTION: no.*
  - *EDITION: no.*
  - *LINK:...*
- **BOOK #7:**
  - *ISBN: 30.*
  - *TITLE: Caballo de Troya 7.*
  - *AUTHOR: Benítez, Juan José.*
  - *EDITORIAL: Booket.*
  - *DESCRIPTION: no.*
  - *EDITION: no.*
  - *LINK:...*
- **BOOK #8:**
  - *ISBN: 31.*



## CAPÍTULO 4: CONCLUSIONES

- *TITLE: Caballo de Troya 8.*
- *AUTHOR: Benitez, Juan Jose.*
- *EDITORIAL: Booket.*
- *DESCRIPTION: no.*
- *EDITION: no.*
- *LINK:...*



The screenshot shows the Voice Library website interface. At the top, there is a navigation bar with links for HOME, DICTIONARY, PRONUNCIATION, LIBRARY, VIDEOCLUB, and GOOGLE. Below the navigation bar, the search results for the author "benitez" are displayed, showing 8 results. The results are presented in a table with columns for ISBN, TITLE, AUTHOR, EDITORIAL, DESCRIPTION, EDITION, and LINK.

ISBN	TITLE	AUTHOR	EDITORIAL	DESCRIPTION	EDITION	LINK
24	Caballo de Troya 1	Benitez, Juan Jose	Booket	no	5	---
25	Caballo de Troya 2	Benitez, Juan Jose	Booket	no	2	---
26	Caballo de Troya 3	Benitez, Juan Jose	Booket	no	no	<a href="http://www.casadelibro.com/libro-saidan-caballo-de-troya-3/707333/?s=00074171">http://www.casadelibro.com/libro-saidan-caballo-de-troya-3/707333/?s=00074171</a>
27	Caballo de Troya 4	Benitez, Juan Jose	Booket	no	no	---
28	Caballo de Troya 5	Benitez, Juan Jose	Booket	no	no	---
29	Caballo de Troya 6	Benitez, Juan Jose	Booket	no	no	---
30	Caballo de Troya 7	Benitez, Juan Jose	Booket	no	no	---
31	Caballo de Troya 8	Benitez, Juan Jose	Booket	no	no	---

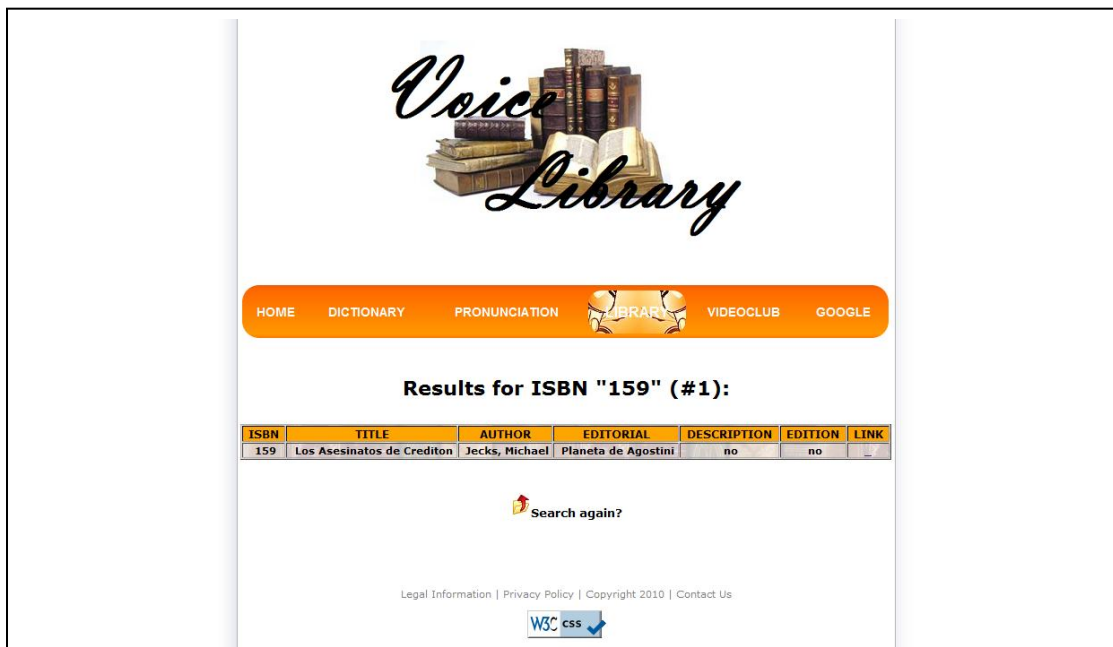
Below the table, there is a "Search again?" button and a footer with links for Legal Information, Privacy Policy, Copyright 2010, and Contact Us. A W3C CSS logo is also visible.

Figura 28. Resultados de PRUEBA #2 sobre VL.

### PRUEBA #3:

- *Wellcome to Voice Library. What is your favorite book? If you do not know how to use this application, say: "INSTRUCTIONS".*
- ONE.
- Ok.

- FIVE.
- *Ok.*
- NINE.
- *Ok.*
- BOOK NUMBER.
- *Ok.*
- SEARCH.
- *Ok.*
- *Results for ISBN "159" (#1):*
  - *BOOK #1:*
    - *ISBN: 159.*
    - *TITLE: Los Asesinatos de Crediton.*
    - *AUTHOR: Jecks, Michael.*
    - *EDITORIAL: Planeta de Agostini.*
    - *DESCRIPTION: no.*
    - *EDITION: no.*
    - *LINK:...*



The screenshot shows the 'Voice Library' website interface. At the top, there is a navigation bar with links for HOME, DICTIONARY, PRONUNCIATION, LIBRAR, VIDEOCLUB, and GOOGLE. Below the navigation bar, the search results are displayed for ISBN "159" (#1). The results are presented in a table with the following columns: ISBN, TITLE, AUTHOR, EDITORIAL, DESCRIPTION, EDITION, and LINK. The table contains one row of data. Below the table, there is a 'Search again?' button with a magnifying glass icon. At the bottom of the page, there is a footer with links for Legal Information, Privacy Policy, Copyright 2010, and Contact Us, along with a 'W3C CSS' logo.

ISBN	TITLE	AUTHOR	EDITORIAL	DESCRIPTION	EDITION	LINK
159	Los Asesinatos de Crediton	Jecks, Michael	Planeta de Agostini	no	no	-

Figura 29. Resultados de PRUEBA #3 sobre VL.

#### PRUEBA #4:



- *Welcome to Voice Videoclub. All you want to know about your favorite films. If you do not know how to use this application, say: "INSTRUCTIONS".*
- MADAGASCAR.
- *Ok.*
- AUSTRALIA.
- *Ok.*
- TURKEY.
- *Ok.*
- ROLLING STONES.
- *Ok.*
- IRELAND.
- *Ok.*
- XANADU.
- *Ok.*
- SEARCH.
- *Ok.*
- *3 Results for ORIGINAL TITLE "MATRIX":*
  - *FILM #1:*
    - *ORIGINAL TITLE: Matrix.*
    - *TRANSLATED TITLE:.*
    - *YEAR: No Da.*
    - *DIRECTOR: No Data.*
    - *LANGUAJES: No Data.*
    - *CATEGORY: No Data.*
    - *DESCRIPTION: No Data.*
    - *IMAGE: No Data.*
  - *FILM #2:*
    - *ORIGINAL TITLE: Matrix Reloaded.*
    - *TRANSLATED TITLE:.*
    - *YEAR: 2003.*
    - *DIRECTOR: Andy Wachowski y Larry Wachowski.*
    - *LANGUAJES: Inglés y Castellano.*

- **CATEGORY:** *Acción/Ciencia Ficción.*
- **DESCRIPTION:** *Anderson tomo una costosa decisión cuando decidió hacer la pregunta que Morfeo y Trinity habían formulado antes que él. Buscar y aceptar la verdad. Liberar su mente de Matrix. Ahora, Neo adquiere un mayor dominio de sus extraordinarios poderes mientras Sion cae sitiada por el Ejército de las Máquinas. Sólo una cuestión de horas separa al último enclave humano en la Tierra de 250.000 Centinelas programados para destruir a la humanidad. Pero los ciudadanos de Sion, animados por la convicción de Morfeo de que el Elegido hará realidad la Profecía del Oráculo y pondrá fin a la guerra con las Máquinas, ponen todas sus esperanzas y expectativas en Neo, que se encuentra bloqueado por visiones inquietantes mientras busca que medidas tomar. Fortalecidos por su amor mutuo y por su creencia en sí mismos, Neo y Trinity deciden volver a Matrix con Morfeo y desatar su arsenal de extraordinarias destrezas y armas contra las sistemáticas fuerzas de represión y explotación. Pero existen poderosas figuras dentro de Matrix que niegan el artificio de la decisión, eludiendo la responsabilidad que supone, al tiempo que se alimentan de las verdades emocionales de los demás. Entretanto, hay exiliados como el Agente Smith, cuya inexorable conexión con Neo le obliga a desobedecer al sistema que ha pedido su eliminación. Impulsado por la humanidad que en otro tiempo desprecio, Smith destruirá todo lo que se interponga en su camino en su búsqueda de venganza. En su peligroso viaje para comprender mejor al constructor de Matrix y su decisivo papel en el destino de la humanidad, Neo se enfrentará a una mayor resistencia, a una verdad aun mayor y a una decisión más imposible de lo jamás imagino. En la confluencia de amor y verdad, fe y conocimiento, intención y razón, Neo debe seguir el camino que ha elegido.*

## CAPÍTULO 4: CONCLUSIONES

- *IMAGE: Image number: 820\_119681312.*
- *FILM #3:*
  - *ORIGINAL TITLE: Matrix Revolutions.*
  - *TRANSLATED TITLE:.*
  - *YEAR: 2003.*
  - *DIRECTOR: Andy Wachowski y Larry Wachowski.*
  - *LANGUAJES: Inglés y Castellano.*
  - *CATEGORY: Acción/Ciencia Ficción.*
  - *DESCRIPTION: Al finalizar The Matrix Reloaded la escena es la siguiente: La flota de Zion destruida casi por completo (sólo quedan dos naves, la Hammer y la Logos), Neo (Reeves) en un estado de coma y el agente Smith adueñado del cuerpo de Bane. La tripulación de Nabucodonosor han de rescatar a Neo del letargo que se encuentra, similar a una conexión a Matrix, pero sin cables. Neo se encuentra realmente en poder del Merovingio. Gracias a la ayuda de Trinity y Morfeo, es rescatado, y tras descubrir que sus poderes como el Elegido no se limitan a Matrix, sino que puede destruir a las máquinas en el mundo real, se encamina a La Ciudad de las Máquinas para intentar solucionar dos acuciantes problemas: por una parte el Agente Smith (Weaving) ha saturado Matrix al reproducirse una y otra vez. Dentro de Matrix ya no existe nadie más que no sea una copia de Smith. Por otra parte, los centinelas se encuentran en ese preciso momento arrasando por completo Zion y pocas esperanzas quedan para la humanidad de sobrevivir. Morfeo y Niobe (Pinkett Smith) llegaron a Zion a tiempo de disparar un Pulso Electromagnético, pero esto sólo retrasará la derrota final por breve espacio de tiempo. Neo y Morfeo se despiden en Zion y más tarde él y Trinity parten en una nave hasta la ciudad de las máquinas. Bane (humano controlado por el agente Smith) se infiltra en la nave e intenta matarlos, pero fracasa, aunque en la pelea*

*Neo queda ciego. Trinity pilota la nave hasta la ciudad de las máquinas, pero debido a la resistencia de los centinelas tienen un aterrizaje brusco, tras el cual Trinity muere. Neo, herido, llega hasta el centro de la ciudad de las máquinas donde se encuentra con Deus ex machina, aparentemente la máquina que gobierna a todas y el Elegido hace un trato con él: la humanidad a cambio de la destrucción del agente Smith. Neo deberá dirimir de una vez por todas el conflicto con su alter ego -Smith, que equilibra la ecuación- en una épica batalla final entre humanos y máquinas. Al final Neo recuerda las palabras del Oráculo: Todo principio tiene un final, y recuerda también que el principio, la razón de existir de Smith, es balancear la ecuación (Neo lo sabe por el Oráculo y por el propio Smith: El objetivo nos une... Neo se da cuenta de que si él mismo deja de existir en Matrix, Smith también lo hará, por lo que se deja copiar por Smith dando por finalizado el objetivo de ambos y permitiendo a Deus ex machina reiniciar Matrix. Deus ex machina da por cumplido el trato. Se termina la guerra y las máquinas evacúan Zion. En lo sucesivo, las máquinas no impedirán que los humanos liberen a los que quieran salir de Matrix. Así se lo asegura el Arquitecto a la Oráculo. Luego se preguntaron si Neo muere lo cual es impreciso por las palabras de la Oráculo el cual dice que muy pronto verán a su salvador. En cuanto al final, en que se puede observar a Neo viendo en forma digital a las máquinas y todo su alrededor, esto es una clara similitud con las ideas que tenía.*

- *IMAGE: Image number: 821\_119681328.*



**PRUEBA #5:**

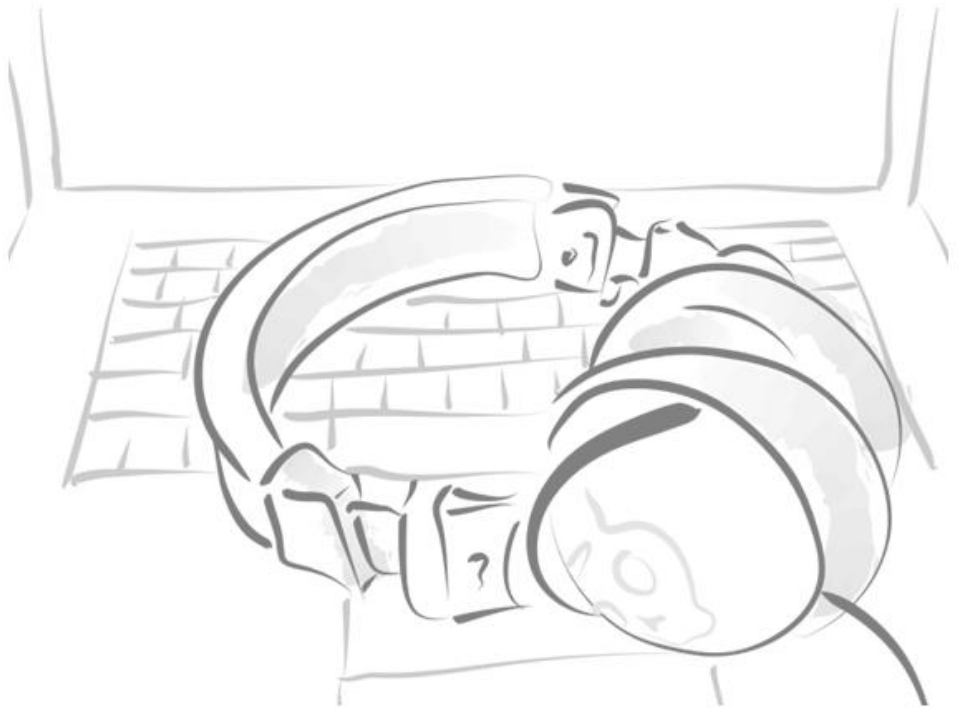
- *Welcome to Voice Videoclub. All you want to know about your favorite films. If you do not know how to use this application, say: "INSTRUCTIONS".*
- JAPAN.
- *Ok.*
- AUSTRALIA.
- *Ok.*
- CANADA.
- *Ok.*
- KUWAIT.
- *Ok.*
- SPAIN.
- *Ok.*
- OMAN.
- *Ok.*
- NEW ZEALAND.
- *Ok.*
- DIRECTOR.
- *Ok.*
- SEARCH.
- *Ok.*
- *1 Results for DIRECTOR "JACKSON":*
  - *FILM #1:*
    - *ORIGINAL TITLE: Blancanieves y los Siete Enanitos.*
    - *TRANSLATED TITLE: Snow White and the Seven Dwarfs.*
    - *YEAR: 1937.*
    - *DIRECTOR: William Cottrell, Wilfred Jackson, Larry Morey, Perce Pearce, Ben Sharpsteen.*
    - *LANGUAJES: Castellano.*
    - *CATEGORY: Animación Disney.*
    - *DESCRIPTION: Hace muchos años, en un país lejano, vivía una princesa llamada Blancanieves. Su madrastra,*

## CAPÍTULO 4: CONCLUSIONES

*una bellísima hechicera, consulto con su espejo mágico quien era la mujer mas hermosa del reino, y al contestarle que Blancanieves ordenó a uno de sus vasallos que llevara a la princesa al interior del bosque y le quitara la vida. En lugar de cumplir la cruel misión que le había sido encomendada, el verdugo dejo libre a Blancanieves, con la condición de que no volviera nunca más al castillo, de modo que la reina no supiera que continuaba viva. Guiada por un grupo de animalillos del bosque, la princesa llegó a una casita situada en lo más profundo de la floresta. En su interior todo era muy pequeño: muebles, enseres, como si estuviera habitada por niños. Todo estaba muy sucio, tanto que Blancanieves y los animales del bosque necesitaron el día entero para limpiarlo. Al fin, cuando la casa brillaba resplandeciente y el puchero hervía alegremente en el fuego, la princesa se acostó sobre tres camas y se quedó dormida. En el otro lugar del bosque, siete enanitos daban por terminada su jornada de trabajo en la mina de diamantes. Caminando, volvieron a su casa y se quedaron muy sorprendidos al ver una luz brillando en la ventana y humo saliendo por la chimenea. Al principio pensaron que la casa estaba invadida por los duendes y tenían mucho miedo a entrar, por lo que enviaron por delante al más pequeño de todos, mientras ellos le seguían, armados de garrotes, dispuestos a dar una paliza al intruso. Dormida sobre su cama encontraron a la princesa Blancanieves. Despertó, admirándose de encontrar a los siete enanitos, a los que contó la causa por la que se hallaba allí. A cambio de que le permitieran quedarse, Blancanieves se ofreció a limpiar la casa y cocinar para ellos, cosa que los enanitos admitieron encantados, y tras la abundante cena celebraron una fiesta. Mientras tanto, en su castillo, la reina consultaba a su espejo mágico sobre la identidad de la mujer más bella del reino, a lo que el espejo contestaba*

*que continuaba siéndolo Blancanieves y que vivía en el bosque en la casa de los enanitos. La madrastra tomó el aspecto de una horrible bruja y encanto una manzana con un hechizo que haría que quién la mordiese caería como muerto y sólo recobraría la vida si recibía un beso de amor, tras lo cual se dirigió al bosque dispuesta a acabar por sí misma con la vida de la princesa. Cuando los enanitos se fueron al trabajo, la bruja le dio a Blancanieves la manzana envenenada y cayó como muerta. Al momento se apercibió de que los amigos de la princesa regresaban, por lo que huyó hacía el monte, encontrando la muerte en su fuga. Los siete enanitos depositaron el cuerpo de su amiga en una urna de cristal. Lloraban amargamente la desaparición de Blancanieves, cuando apareció un príncipe llegado de un lejano país, atraído por la fama de la belleza de la princesa. Al verla tan bonita, el príncipe la besó, con lo que el hechizo quedó deshecho. El príncipe se llevó a Blanca.*

- **IMAGE:** Image number: 110\_119679390.





**Results for DIRECTOR "JACKSON":**

**Film #1:**

**ORIGINAL TITLE:** Blancanieves y los Siete Enanitos

**TRANSLATED TITLE:** Snow White and the Seven Dwarfs

**YEAR:** 1937

**DIRECTOR:** William Cottrell, Wilfred Jackson, Larry Morey, Perce Pearce, Ben Sharpsteen

**LANGUAGES:** Castellano

**CATEGORY:** Animacion Disney

**DESCRIPTION:**  
 Hace muchos años, en un país lejano, vivía una princesa llamada Blancanieves. Su madrastra, una bellísima hechicera, consultó con su espejo mágico quien era la mujer más hermosa del reino, y al contestarle que Blancanieves ordenó a uno de sus vasallos que llevara a la princesa al interior del bosque y le quitara la vida. En lugar de cumplir la cruel misión que le había sido encomendada, el verdugo dejó libre a Blancanieves, con la condición de que no volviera nunca más al castillo, de modo que la reina no supiera que continuaba viva. Guiada por un grupo de animalillos del bosque, la princesa llegó a una casita situada en lo más profundo de la floresta. En su interior todo era muy pequeño: muebles, enseres, como si estuviera habitada por niños. Todo estaba muy sucio, tanto que Blancanieves y los animales del bosque necesitaron el día entero para limpiarlo. Al fin, cuando la casa brillaba resplandeciente y el puchero hervía alegremente en el fuego, la princesa se acostó sobre tres camas y se quedó dormida. En el otro lugar del bosque, siete enanitos daban por terminada su jornada de trabajo en la mina de diamantes. Caminando, volvieron a su casa y se quedaron muy sorprendidos al ver una luz brillando en la ventana y humo saliendo por la chimenea. Al principio pensaron que la casa estaba invadida por los duendes y tenían mucho miedo a entrar, por lo que enviaron por delante al más pequeño de todos, mientras ellos le seguían, armados de garrotes, dispuestos a dar una paliza al intruso. Dormida sobre su cama encontraron a la princesa Blancanieves. Despierto, admirándose de encontrar a los siete enanitos, a los que contó la causa por la que se hallaba allí. A cambio de que le permitieran quedarse, Blancanieves se ofreció a limpiar la casa y cocinar para ellos, cosa que los enanitos admitieron encantados, y tras la abundante cena celebraron una fiesta. Mientras tanto, en su castillo, la reina consultaba a su espejo mágico sobre la identidad de la mujer más bella del reino, a lo que el espejo contestaba que continuaba siendo Blancanieves y que vivía en el bosque en la casa de los enanitos. La madrastra tomó el aspecto de una horrible bruja y encantó una manzana con un hechizo que haría que quien la mordiese cayera como muerto y solo recobraría la vida si recibía un beso de amor, tras lo cual se dirigió al bosque dispuesta a acabar por sí misma con la vida de la princesa. Cuando los enanitos se fueron al trabajo, la bruja le dio a Blancanieves la manzana envenenada y cayó como muerta. Al momento se acercó de que los amigos de la princesa regresaban, por lo que huyó hacia el monte, encontrando la muerte en su fuga. Los siete enanitos depositaron el cuerpo de su amiga en una urna de cristal. Lloraban amargamente la desaparición de Blancanieves, cuando apareció un príncipe llegado de un lejano país, atraído por la fama de la belleza de la princesa. Al verla tan bonita, el príncipe le besó, con lo que el hechizo quedó deshecho. El príncipe se llevó a Blanca

**IMAGE:**

[Search again?](#)

Legal Information | Privacy Policy | Copyright 2010 | Contact Us

W3C CSS

Figura 31. Resultados de PRUEBA #5 sobre VV.

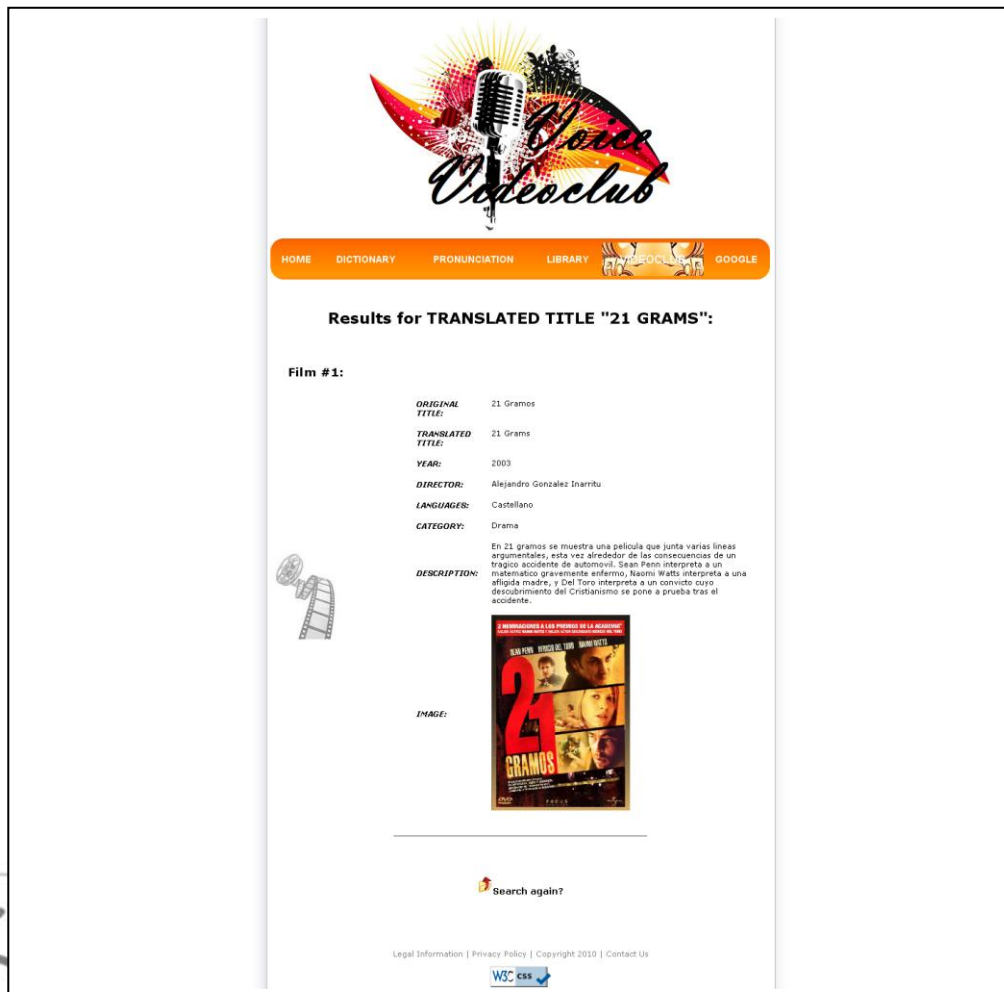
**PRUEBA #6:**

- *Welcome to Voice Videoclub. All you want to know about your favorite films. If you do not know how to use this application, say: "INSTRUCTIONS".*
- TWO.
- *Ok.*
- ONE.
- *Ok.*
- BLANK.
- *Ok.*
- GREECE.
- *Ok.*
- ROLLING STONES.
- *Ok.*
- AUSTRALIA.
- *Ok.*
- MADAGASCAR.
- *Ok.*
- SPAIN.
- *Ok.*
- TRANSLATED TITLE.
- *Ok.*
- SEARCH.
- *Ok.*
- *1 Results for TRANSLATED TITLE "21 GRAMS":*
  - *FILM #1:*
    - *ORIGINAL TITLE: 21 Gramos.*
    - *TRANSLATED TITLE: 21 Grams.*
    - *YEAR: 2003. DIRECTOR: Alejandro González Inarritu.*  
*LANGUAJES: Castellano.*
    - *CATEGORY: Drama.*
    - *DESCRIPTION: En 21 gramos se muestra una película que junta varias líneas argumentales, esta vez alrededor de las*


## CAPÍTULO 4: CONCLUSIONES

*consecuencias de un trágico accidente de automóvil. Sean Penn interpreta a un matemático gravemente enfermo, Naomi Watts interpreta a una afligida madre, y Del Toro interpreta a un convicto cuyo descubrimiento del Cristianismo se pone a prueba tras el accidente.*

- **IMAGE:** Image number: 11\_119678390.



**Voces Videoclub**


HOME   DICTIONARY   PRONUNCIATION   LIBRARY      GOOGLE


**Results for TRANSLATED TITLE "21 GRAMS":**

**Film #1:**

<b>ORIGINAL TITLE:</b>	21 Gramos
<b>TRANSLATED TITLE:</b>	21 Grams
<b>YEAR:</b>	2003
<b>DIRECTOR:</b>	Alejandro Gonzalez Inarritu
<b>LANGUAGES:</b>	Castellano
<b>CATEGORY:</b>	Drama

**DESCRIPTION:** En 21 gramos se muestra una película que junta varias líneas argumentales, esta vez alrededor de las consecuencias de un trágico accidente de automóvil. Sean Penn interpreta a un matemático gravemente enfermo, Naomi Watts interpreta a una afligida madre, y Del Toro interpreta a un convicto cuyo descubrimiento del Cristianismo se pone a prueba tras el accidente.

**IMAGE:** 

 Search again?

Legal Information | Privacy Policy | Copyright 2010 | Contact Us




Figura 32. Resultados de PRUEBA #6 sobre VV.

### 3.4.5 Gestión privada de bases de datos

La etiqueta situada en la parte inferior de las páginas principales de las aplicaciones VL y VV, despliega el menú de acceso al sistema de gestión de las bases de datos correspondientes a las aplicaciones. Para acceder al sistema de gestión es necesario introducir un nombre de usuario y contraseña de administrador. El control de seguridad está creado para evitar que cualquier usuario externo a la administración del entorno pueda acceder al sistema gestor de las bases de datos. En la Figura 33 podemos ver el menú de acceso en la aplicación VV.

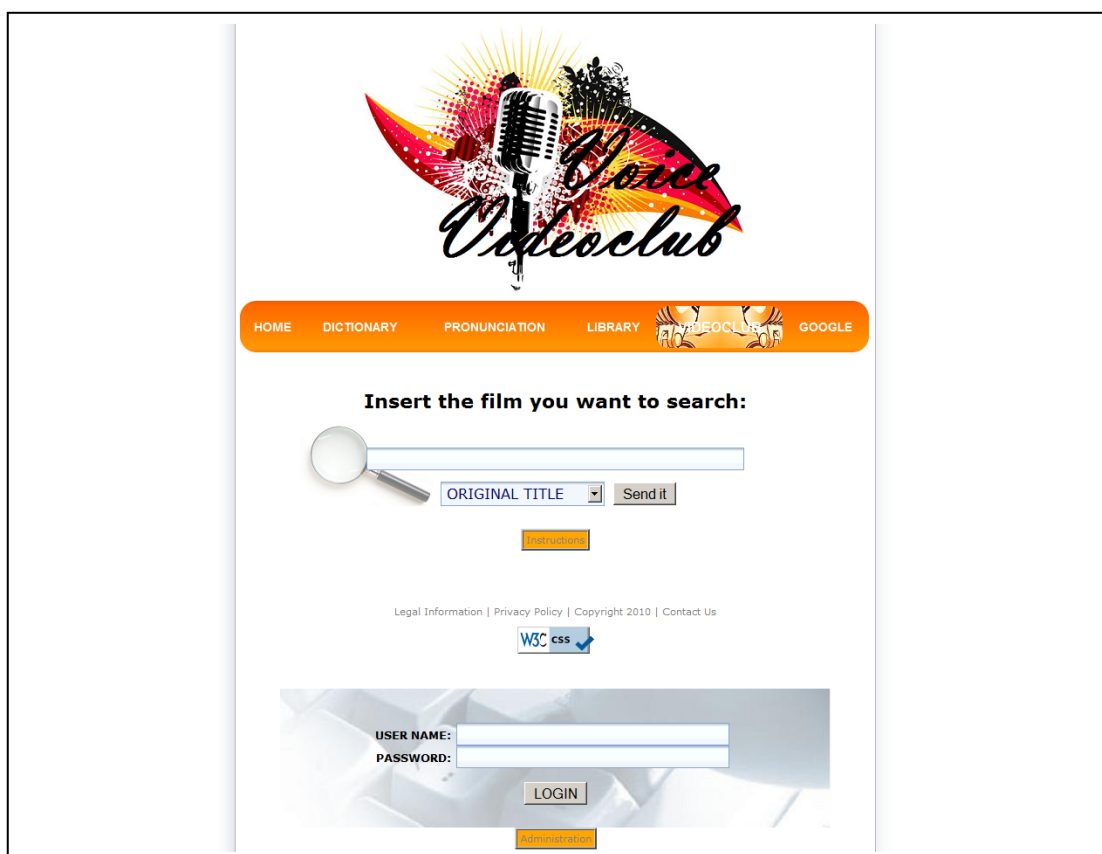


Figura 33. Menú de acceso al sistema gestor de BBDD.

Este menú está diseñado de manera independiente a la aplicación XHTML+Voice, excluyendo completamente la gestión a través de interfaz oral debido a la ausencia en este entorno de usuarios.

Al hacer click en LOGIN se ejecuta un proceso en PHP que comprueba que los datos de nombre de usuario y contraseña sean correctos, comparándolos con una base de datos. En caso de que no sean correctos, el sistema redirige a una página de error. En caso de que sean correctos, el sistema muestra la página de gestión de la base de datos correspondiente. La página contiene el menú de gestión mostrado en la Figura 34, en el que se puede elegir entre insertar, modificar, o borrar una película (en el caso de la aplicación VV; la misma gestión pero con libros se realiza en la aplicación VL) de la base de datos.

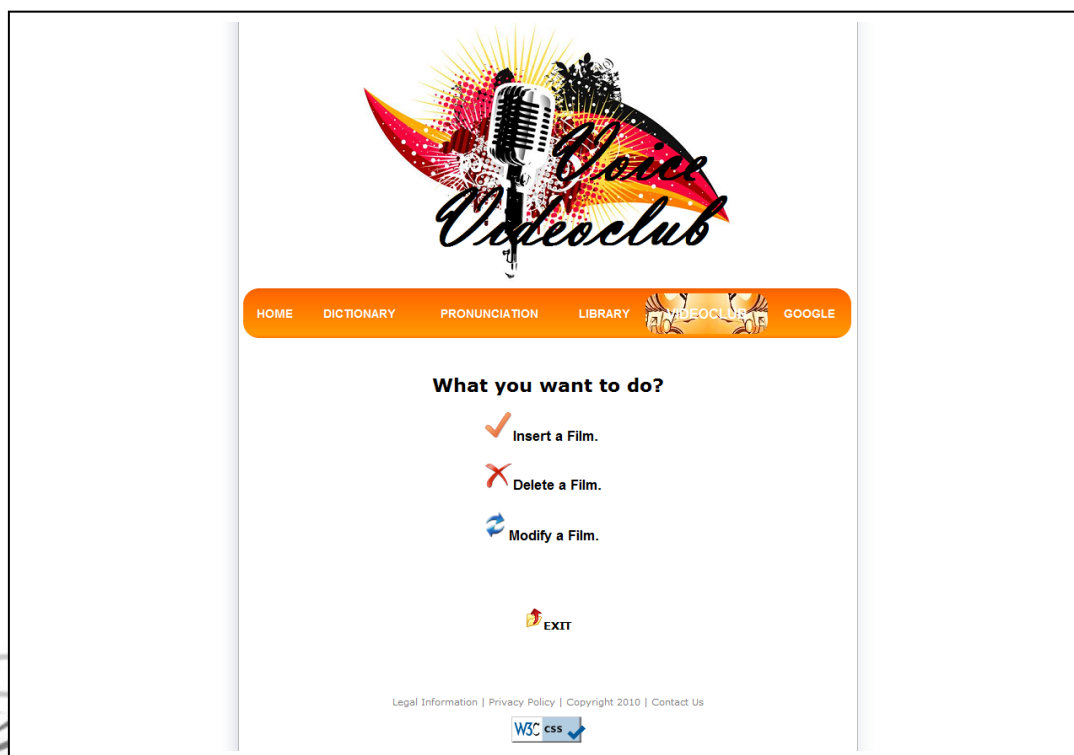


Figura 34. Menú de gestión de la BBDD Videoclub.

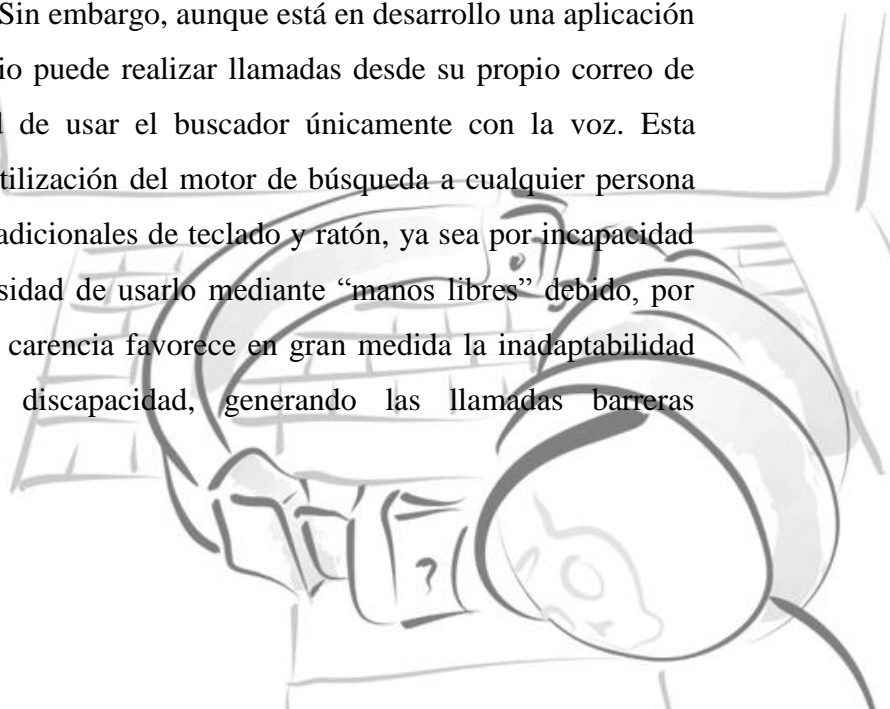
Una vez seleccionada la tarea de gestión se accede a una nueva página con los formularios a rellenar. En caso de querer insertar un elemento, se completan todos los campos correspondientes a las características de la película/libro. Para borrar un elemento de la base de datos sería necesario introducir el título original en el caso de las películas, y el ISBN en el caso de los libros. Si se pretende modificar los datos de un elemento almacenado previamente se realizan las dos operaciones anteriores, se

borra el elemento buscándolo por su identificador, y se completan los nuevos datos para poder insertarlo.

Cada una de las tareas de gestión nos lleva a una página de proceso codificada en PHP con permisos de acceso a las bases de datos de MySQL. La página de proceso envía la consulta, inserción o borrado del elemento directamente a la base de datos correspondiente, recibiendo una respuesta de OK/Error mostrada al administrador por pantalla en el entorno de VA.

## 3.5 Voice Google

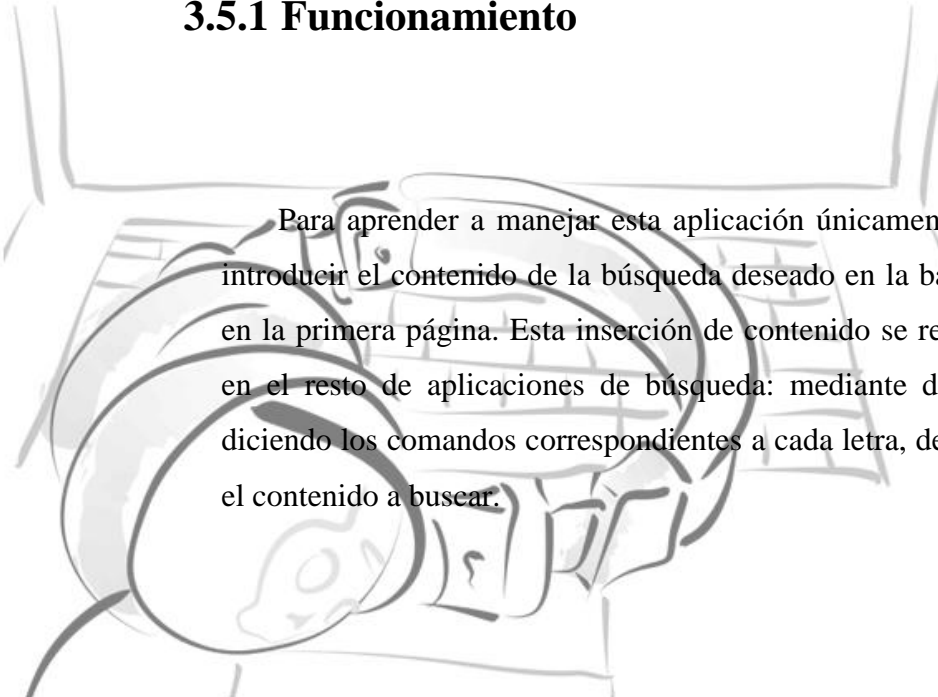
Una de las plataformas más importantes actualmente gracias a su motor de búsqueda de internet es Google. Sin embargo, aunque está en desarrollo una aplicación oral a través de la cual el usuario puede realizar llamadas desde su propio correo de Gmail, no existe la posibilidad de usar el buscador únicamente con la voz. Esta imposibilidad deja fuera de la utilización del motor de búsqueda a cualquier persona que no pueda usar los medios tradicionales de teclado y ratón, ya sea por incapacidad motora o por la imperiosa necesidad de usarlo mediante “manos libres” debido, por ejemplo, a ir conduciendo. Esta carencia favorece en gran medida la inadaptabilidad de internet a personas con discapacidad, generando las llamadas barreras arquitectónicas de internet.



Existe un uso parcial de los contenidos de internet en general gracias a ordenadores completamente adaptados a personas con la capacidad motora disminuida, o visibilidad parcial completamente disminuida. Incluso, navegadores como Opera, favorecen el uso de internet a este sector de la población bastante desatendido. Sin embargo el uso que puedan hacer de estas herramientas dista mucho del uso que le da el resto de la población.

Gracias a la aplicación *Voice Google* (VG) el usuario puede acceder al buscador e introducir cualquier contenido que desee en la plataforma mediante su voz. La interfaz de la aplicación recibirá el contenido y mostrará al usuario los resultados tanto a nivel visual como a nivel oral. Además, la aplicación permitirá seleccionar al usuario cualquiera de los enlaces mostrados en la página de resultados mediante su voz, y la plataforma le redirigirá esa página. De esta manera, se ampliará una funcionalidad esencial al mejor motor de búsqueda jamás creado, integrando a sectores de usuarios para los que antaño podría resultar impensable adaptar. Las interfaces visuales y orales están igualadas de tal manera que desde ambas modalidades se realiza el mismo manejo de la herramienta.

### 3.5.1 Funcionamiento



Para aprender a manejar esta aplicación únicamente con la voz, el usuario debe introducir el contenido de la búsqueda deseado en la barra de formulario que aparece en la primera página. Esta inserción de contenido se realiza de la misma manera que en el resto de aplicaciones de búsqueda: mediante deletreo. El deletreo se realiza diciendo los comandos correspondientes a cada letra, de forma que se va completando el contenido a buscar.

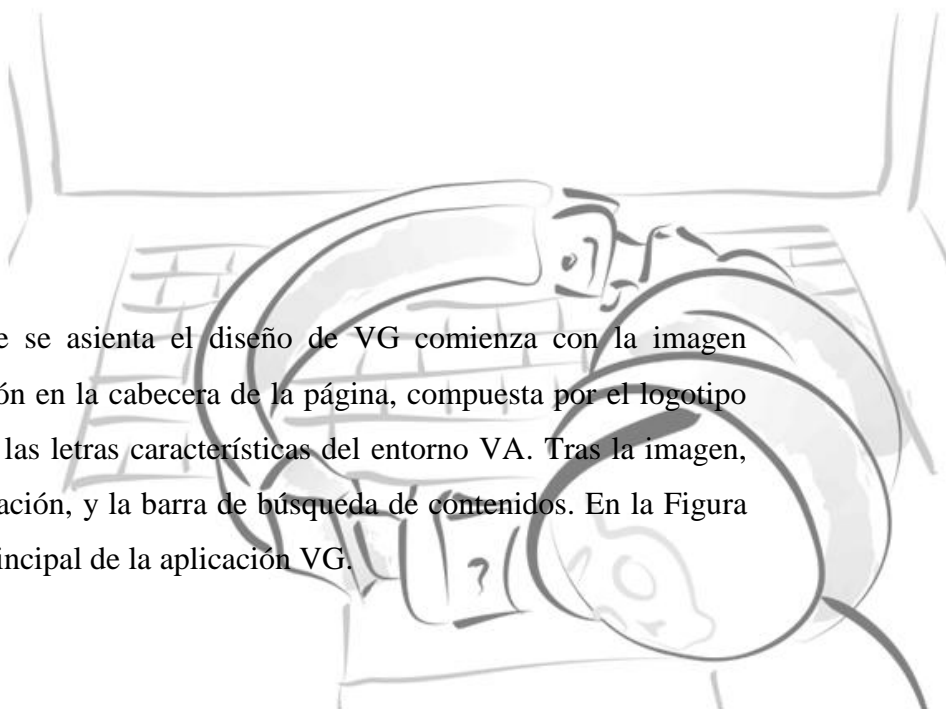
Una vez que el resultado se muestra por pantalla la interfaz oral lee los contenidos al usuario además de mostrarlos por pantalla. Se muestran los diez primeros resultados, encabezando cada vínculo encontrado con la palabra “LINK” y el número correspondiente al orden que ocupa en la narración. De esta forma, si el usuario desea visitar el enlace número cuatro, no debe aprenderse el nombre completo del enlace y decir a la aplicación “*www.enlace.com*”, sino que únicamente debe decir el comando “*LINK FOUR*”, y la aplicación redirigirá al usuario a el vínculo correspondiente.

En cualquier momento de la narración, al igual que ocurre en el resto de aplicaciones, el usuario puede interrumpirla, visitar otras aplicaciones, o directamente si ya ha decidido el enlace que desea visitar, decir el comando correspondiente a ese enlace. Además, el usuario puede decir el comando “*SEARCH AGAIN*” y volver a la página principal de la aplicación VG para realizar una nueva búsqueda.

Todas las instrucciones necesarias para manejar la aplicación únicamente con la voz, se especifican en una pestaña bajo la barra de búsqueda en la pantalla principal de la aplicación, en la que al hacer click, son desplegadas.

### 3.5.2 Diseño

La estructura en la que se asienta el diseño de VG comienza con la imagen representativa de la aplicación en la cabecera de la página, compuesta por el logotipo de Google, un micrófono, y las letras características del entorno VA. Tras la imagen, se muestra el menú de aplicación, y la barra de búsqueda de contenidos. En la Figura 35 podemos ver la página principal de la aplicación VG.





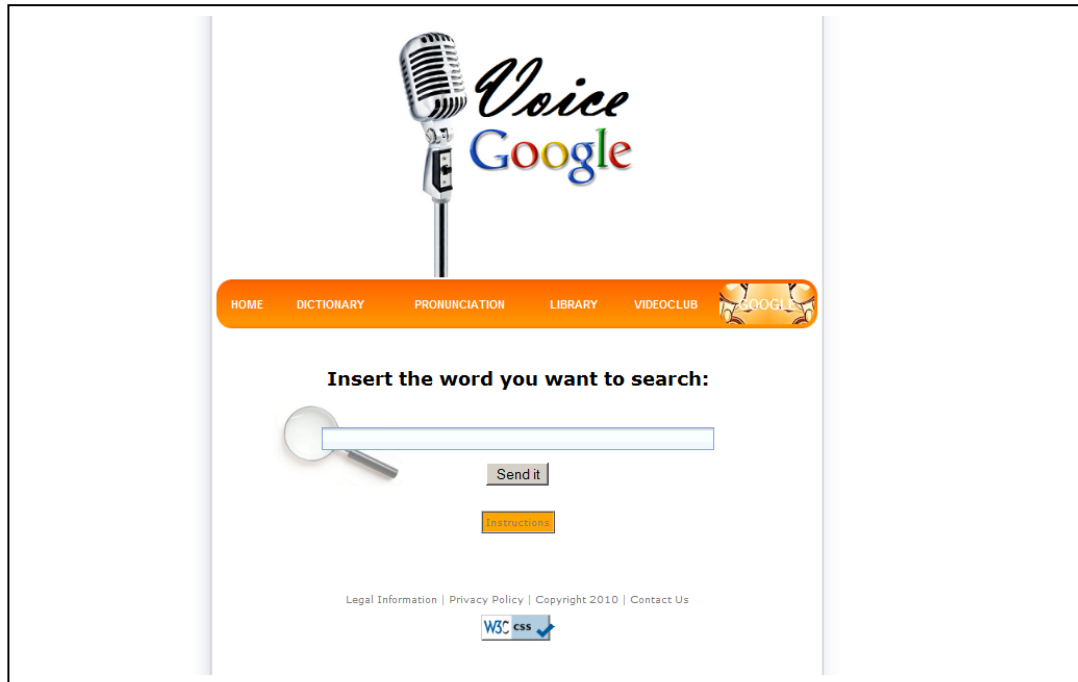


Figura 35. Página principal de la aplicación VG.

En la página de resultado, una vez el usuario ha obtenido el contenido buscado, se muestra el mismo aspecto y diseño, salvo en el cuerpo de la página, donde se visualizan los diez primeros enlaces de resultado, y una etiqueta para volver a buscar más contenidos. El aspecto oral, se caracteriza por seguir los patrones de aplicaciones anteriores, narrando un diálogo de presentación en la página principal, recibiendo los comandos del usuario, y narrando los contenidos de los resultados para que el usuario pueda seleccionar un vínculo a visitar, volver atrás, o navegar por el resto de la aplicación.

### 3.5.3 Arquitectura

El código de la página principal de la aplicación, está distribuido de la misma forma que en aplicaciones anteriores, dejando por un lado la información, por otro la interfaz de diálogo, y por último la interfaz oral. El archivo de acceso a la aplicación que contiene la estructura anterior se denomina “*sgoogle.xhtml*”. El usuario introduce la información que desea buscar, a través de cualquiera de las dos interfaces, y el archivo lanza el proceso codificado en el archivo “*google\_process.php*”. Este proceso recoge el contenido de la información, lo incrusta en la dirección del buscador, y lanza la función predefinida “*file\_get\_contents()*”.

En un intento por optimizar las búsquedas se lanzó la función predefinida sobre la página principal del buscador Google, sin embargo el resultado no fue el esperado. La información obtenida distaba mucho de los resultados de búsqueda. Esto nos llevó a lanzar la función sobre las páginas de resultados del propio Google, sin obtener resultados satisfactorios. Se probó a utilizar otro buscador, como es el proporcionado por El Mundo, Ariadna, con un resultado óptimo. Este buscador utiliza el motor de Google para sus propias búsquedas, favoreciendo unos resultados acordes a las expectativas de la aplicación VG. Sin embargo, este buscador actualizó su código, incluyendo más publicidad, y líneas de código variables, imposibles de predecir, lo que desestructuró los resultados de nuestra propia aplicación, y nos obligó a volver a encontrar otro buscador. Éste fue Lycos, cuyo código, similar al antiguo Ariadna, favoreció la tarea de obtención de resultados. El nombre de la aplicación y logotipo se han mantenido por la facilidad de asimilación de conceptos a la hora de hablar de Google y buscador web, asociándolos en la actualidad como verdaderos sinónimos.

Una vez que almacenamos la información obtenida con la función “*file\_get\_contents()*”, procedemos a filtrar los contenidos importantes. Eliminamos las etiquetas innecesarias, la publicidad y la información irrelevante, y mantenemos los vínculos resultantes con sus respectivas etiquetas. Una vez que el contenido está procesado y listo, cogemos la plantilla de resultados denominada como “*rgoogle.xhtml*” e incrustamos la información tanto a nivel oral como a nivel visual,

creando el archivo “*rgoogle2.xhtml*”. Este último archivo está ya preparado para ser mostrado al usuario y soportar las interacciones con este, por lo que se redirige desde el archivo de proceso al archivo de resultados. En la Figura 36 podemos ver el flujo de la información descrito en las líneas anteriores.

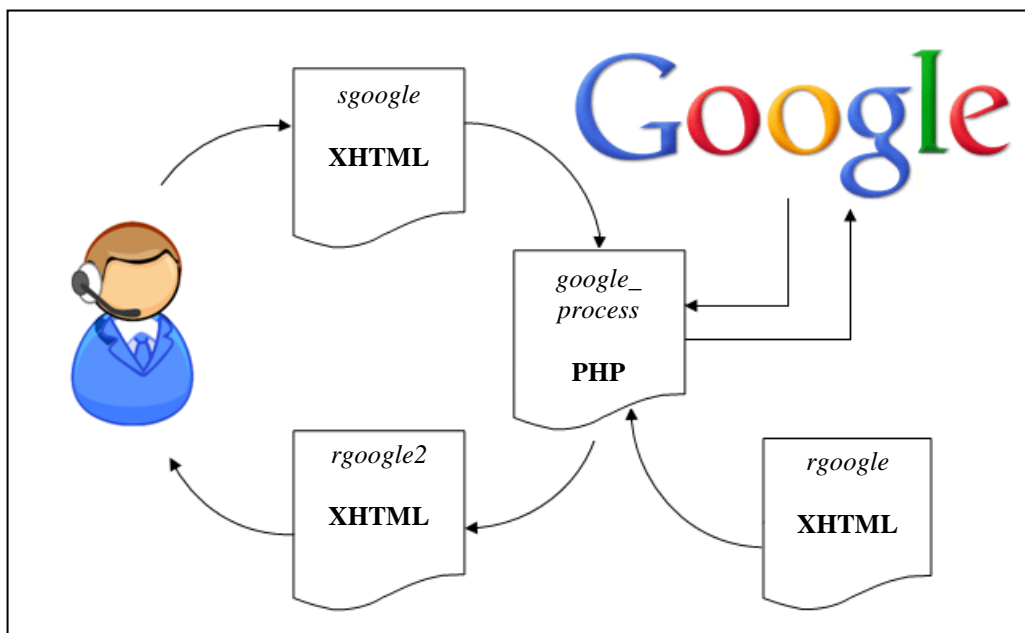


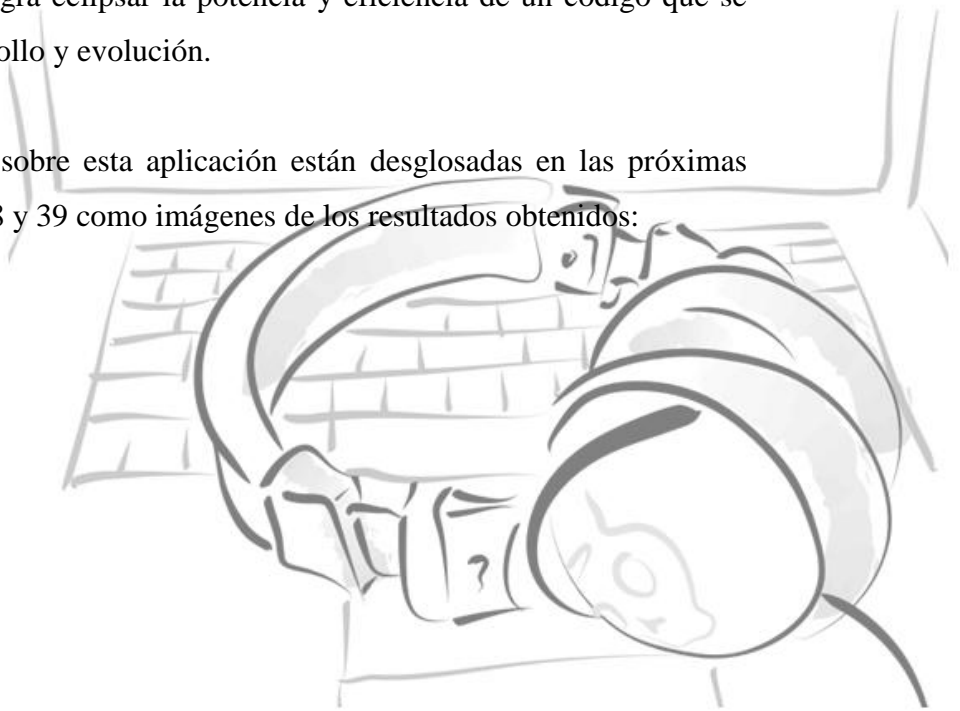
Figura 36. Flujo de información de la aplicación VG.

El archivo correspondiente a los resultados que se muestran al usuario, registra la información de tal manera que muestra al usuario los vínculos encontrados y sus descripciones junto a la palabra “LINK” y el número (del uno al diez) que le corresponde, según el orden que le sitúa en la lista. Estos enlaces son incrustados también a nivel oral, de manera que el usuario recibe la narración de la máquina ordenada de forma que dice “LINK”, vínculo, y descripción. Una vez que el usuario asocia los comandos numéricos orales con los vínculos puede decirle a la máquina el enlace que desea visitar, redireccionándole inmediatamente a ese enlace.

### 3.5.4 Pruebas y limitación de errores

A la hora de limitar los errores, se ha observado que en la construcción de la aplicación VG, como ocurre también con la aplicación VD, es muy difícil establecer un código que permanezca inalterable tanto en su funcionamiento como en su estructura. Esta dificultad viene determinada por la extracción del código de páginas, como Wikipedia y Google (Lycos, como vimos en el apartado anterior), que pueden actualizar sus páginas y modificar su código, alterando la estructura de los procesos. Los procesos de estas aplicaciones convierten las páginas en texto, y seleccionan los contenidos mediante búsquedas de etiquetas predefinidas que se entiende permanecerán constantes. Sin embargo, como comprobamos con el buscador Ariadna, los desarrolladores modifican sus entorno, cambiando etiquetas e identificadores muchas veces sin que estos cambios se aprecien a nivel visual. Estos cambios afectan en gran medida a los patrones de conducta de los procesos, por lo que las aplicaciones se ven sometidas a la voluntad de desarrolladores externos a la aplicación. Si bien, esta aplicación está preparada para estos cambios, además de ser una guía de futuros desarrollos en esta tecnología más que una aplicación comercial, esta inadaptabilidad y dependencia dinámica no logra eclipsar la potencia y eficiencia de un código que se pretende en constante desarrollo y evolución.

Las pruebas realizadas sobre esta aplicación están desglosadas en las próximas líneas, con las Figuras 37, 38 y 39 como imágenes de los resultados obtenidos:



**PRUEBA #1:**

- *Welcome to Voice Google. What do you want to search? If you don't know how to use this application, say: "INSTRUCTIONS".*
- AUSTRALIA.
- *Ok.*
- TURKEY.
- *Ok.*
- IRELAND.
- *Ok.*
- LUXEMBOURG.
- *Ok.*
- AUSTRALIA.
- *Ok.*
- SEARCH.
- *Ok.*
- *Results for "atila":*
  - *LINK 1: Attila - Wikipedia, the free encyclopedia... Atilla/Etele in Hungarian (all the three name variants are used in Hungary; Attila is the most popular variant), Etzel in the German Nibelungenlied, or Attila, Atila or ... en.wikipedia.org*
  - *LINK 2: Atila - WelcomeAtila toured with the legendary Buddy Greco and earned a new fan in the process. Buddy said, "I was knocked out by Atila. He has the voice and the class to show the ... www.atila.co.uk*
  - *LINK 3: Atila - Wikipedia, the free encyclopedia. Atila was a Spanish progressive rock group of the 1970s. The four-member band released two studio albums and a live album. Discography. 1975 - The Beginning Of The End – Live en.wikipedia.org*
  - *LINK 4: ATILA music, discography, MP3, videos and reviews ATILA is a Symphonic Prog / Progressive Rock artist from Spain. This page includes ATILA's : biography, official website, pictures, videos from YouTube, MP3 (free ... www.progarchives.com*

- *LINK 5: Attila (TV 2001) - IMDb Directed by Dick Lowry. With Gerard Butler, Powers Boothe, Simone Mackinnon, Reg Rogers. A romanced story of Attila the Hun, since his childhood, when he lost his ... [www.imdb.com](http://www.imdb.com)*
- *LINK 6: Atila | Hello Young Lovers | CD Baby Listen to and buy Atila music on CD Baby. Download Hello Young Lovers by Atila on the independent record store by musicians for musicians. [www.cdbaby.com](http://www.cdbaby.com)*
- *LINK 7: ATILA on MySpace Music - Free Streaming MP3s, Pictures ...MySpace profile for ATILA. Find friends, share photos, keep in touch with classmates, and meet new people on MySpace. [www.myspace.com](http://www.myspace.com)*
- *LINK 8: ATILA - Association of Turicum International Business Law AlumniHomepage der Alumni-Vereinigung der Juristischen Fakultät der Universität Zürich, Schweiz. - Homepage of business law alumni of the University Zurich, Switzerland. [www.atila-zh.ch](http://www.atila-zh.ch)*
- *LINK 9: Atila BiosystemsAtila Biosystems not only provide innovative technology, but also technology to meet ultra high throughput miRNA and SNP ... [atilabiosystems.com](http://atilabiosystems.com)*
- *LINK 10: Flickr: AtilaAbout Atila / atila. Photostream. I AM OLDER THAN 18 YO. way! older!!! Photos of Atila See more... Atila's favorite photos from other Flickr members See more. [www.flickr.com](http://www.flickr.com)*
- *If you want to visit the links, you must to say: "LINK", and the number of link you want to visit. For example, "LINK 13".*

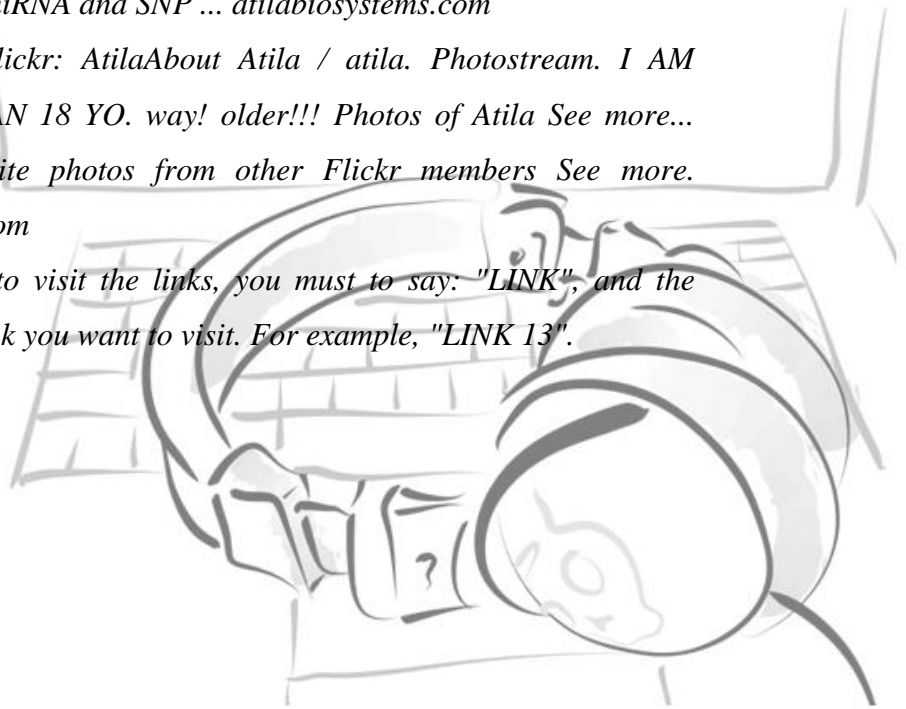




Figura 37. Resultados de la Prueba #1 sobre VG.

**PRUEBA #2:**

- *Welcome to Voice Google. What do you want to search? If you don't know how to use this application, say: "INSTRUCTIONS".*
- TURKEY.
- *Ok.*
- HUNGARY.
- *Ok.*
- EGYPT.
- *Ok.*
- BLANK.

- *Ok.*
- BURKINA FASO.
- *Ok.*
- EGYPT.
- *Ok.*
- AUSTRALIA.
- *Ok.*
- TURKEY.
- *Ok.*
- LUXEMBOURG.
- *Ok.*
- EGYPT.
- *Ok.*
- SPAIN.
- *Ok.*
- SEARCH.
- *Ok.*
- *Results for "the beatles":*
  - *LINK 1: The Beatles 'Red' and 'Blue' have been remastered by the same dedicated team of engineers at EMI Music's Abbey Road Studios responsible for remastering The Beatles' original ...  
[www.thebeatles.com](http://www.thebeatles.com)*
  - *LINK 2: The Beatles - Wikipedia, the free encyclopedia The Beatles were an English rock band, formed in Liverpool in 1960, and one of the most commercially successful and critically acclaimed acts in the history of popular music ...  
[en.wikipedia.org](http://en.wikipedia.org)*
  - *LINK 3: The Beatles: Biography from Answers.com Born: 1962 Birthplace: Liverpool, England Died: 1970 Best Known As: The Fab Four One of the biggest musical acts in history, The Beatles were John Lennon (guitar ...  
[www.answers.com](http://www.answers.com)*
  - *LINK 4: The Beatles (TV series) - Wikipedia, the free encyclopedia The Beatles is an American animated television series featuring the fanciful and musical misadventures of the popular English rock band of the same name.  
[en.wikipedia.org](http://en.wikipedia.org)*



## CAPÍTULO 4: CONCLUSIONES

- *LINK 5: The Beatles - TvWiki, the free encyclopedia. The Beatles were a British pop and rock group from Liverpool, England. The group shattered many sales records and charted more than 50 top 40 hit singles, including 20 #1 ... [twiki.tv](http://twiki.tv)*
- *LINK 6: The Beatles - Simple English Wikipedia, the free encyclopedia. The Beatles were a very popular rock and pop band in the 1960s. They were from Liverpool, England. The members of the band were John Lennon (rhythm guitar, keyboards), Paul ... [simple.wikipedia.org](http://simple.wikipedia.org)*
- *LINK 7: The Beatles - Fan History Wiki: The Fandom History ResourceIntroduction. The Beatles fan fiction community likely traces its roots back to the mid to late 1960s. During this era and during the ensuing thirty years, most of the fan ... [www.fanhistory.com](http://www.fanhistory.com)*
- *LINK 8: The Beatles – Free listening, videos, concerts, stats ...Watch videos amp; listen to The Beatles: Come Together, Let It Be amp; more, plus 865 pictures. The Beatles were an iconic rock group from Liverpool, England. They are ... [www.last.fm](http://www.last.fm)*
- *LINK 9: The Beatles | Music Videos, News, Photos, Tour Dates ...So much has been said and written about the Beatles -- and their story is so mythic in its sweep -- that it's difficult to summarize their career without restating ... [www.mtv.com](http://www.mtv.com)*
- *LINK 10: The Beatles - The Internet Beatles Album A collection of Beatles related informational, sound and picture files, made available for study by Beatles fans and scholars all over the world to explore the behind-the ... [www.beatlesagain.com](http://www.beatlesagain.com)*
- *If you want to visit the links, you must to say: "LINK", and the number of link you want to visit. For example, "LINK 8".*



Figura 38. Resultados de la Prueba #2 sobre VG.

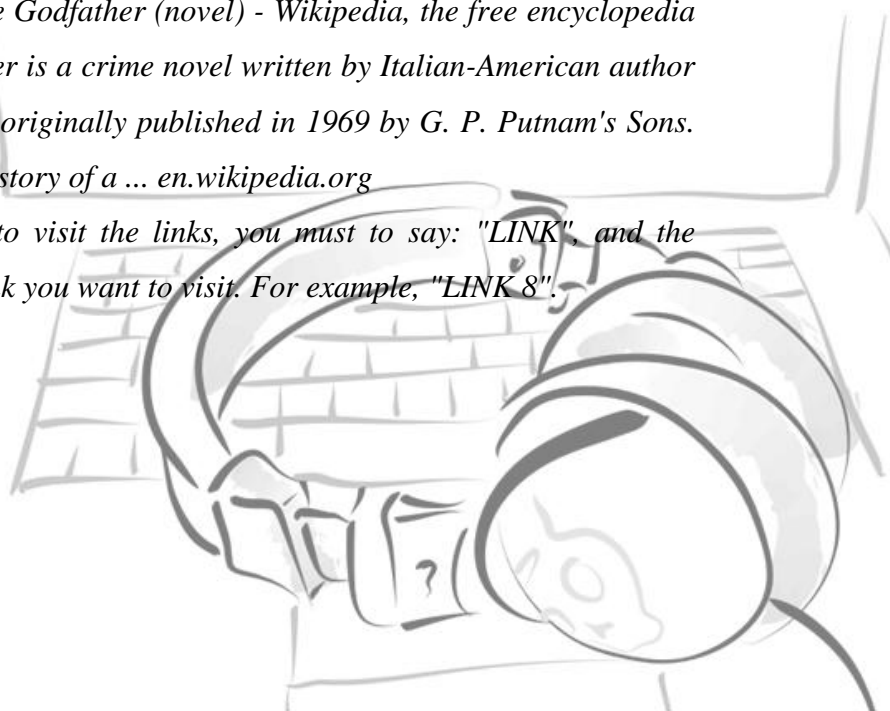
### PRUEBA #3:

- *Welcome to Voice Google. What do you want to search? If you don't know how to use this application, say: "INSTRUCTIONS".*
- MADAGASCAR.
- *Ok.*
- AUSTRALIA.
- *Ok.*
- ROLLING STONES.
- *Ok.*
- IRELAND.

- *Ok.*
- OMAN.
- *Ok.*
- BLANK.
- *Ok.*
- PAKISTAN.
- *Ok.*
- UNITED KINGDOM.
- *Ok.*
- ZIMBABWE.
- *Ok.*
- OMAN.
- *Ok.*
- SEARCH.
- *Ok.*
- *Results for "mario puzo":*
  - *LINK 1: Mario Puzo - The Official Library; Bookstore The Official site for author Mario Puzo, author of The Godfather, The Last Don, and others. Features a comprehensive store. mariopuzo.com*
  - *LINK 2: Mario Puzo - Wikipedia, the free encyclopedia Mario Gianluigi Puzo (October 15, 1920 – July 2, 1999) was an American author and screenwriter, known for his novels about the Mafia, including The Godfather (1969 ... en.wikipedia.org*
  - *LINK 3: Mario Puzo: Information from Answers.com Works by Mario Puzo (1920-1999) 1969 The Godfather. After two previous critically acclaimed but unpopular novels, Puzo produces what has been www.answers.com*
  - *LINK 4: The Official Mario Puzo Library The Official site for author Mario Puzo, author of The Godfather, The Last Don, and others. Features a comprehensive store. www.jgeoff.com*
  - *LINK 5: Mario Puzo - IMDb Mario Puzo was born October 15, 1920, Hell's Kitchenquot; on Manhattan's (NY) West Side and,*

*following military service in World War II, attended New York's New ... [www.imdb.com](http://www.imdb.com)*

- *LINK 6: Mario Puzo BIOGRAPHY Mario Puzo's Biography. Mario Puzo was born October 15, 1920, in Hell's Kitchen on Manhattan's (NY) West Side and, following military service in World War II ... [www.mariopuzo.com](http://www.mariopuzo.com)*
- *LINK 7: Mario Puzo: Facts, Discussion Forum, and Encyclopedia Article The Dark Arena is the first novel by Mario Puzo, published in 1955. The book follows Walter Mosca, an American World War II veteran who is forced to return to Germany for ... [www.absoluteastronomy.com](http://www.absoluteastronomy.com)*
- *LINK 8: Mario Puzo Biography - Mario Puzo Childhood, Life and Timeline Mario Puzo was an Italian-American author and screenwriter. Read this brief biography to find more on his life. [www.thefamouspeople.com](http://www.thefamouspeople.com)*
- *LINK 9: Mario Puzo Achievements, Author Biography, Trivia, and Movies ... Mario Gianluigi Puzo was a two time Academy Award-winning Italian American author and screenwriter, known for his novels about the Mafia, especially The ... [www.juggle.com](http://www.juggle.com)*
- *LINK 10: The Godfather (novel) - Wikipedia, the free encyclopedia The Godfather is a crime novel written by Italian-American author Mario Puzo, originally published in 1969 by G. P. Putnam's Sons. It details the story of a ... [en.wikipedia.org](http://en.wikipedia.org)*
- *If you want to visit the links, you must to say: "LINK", and the number of link you want to visit. For example, "LINK 8".*



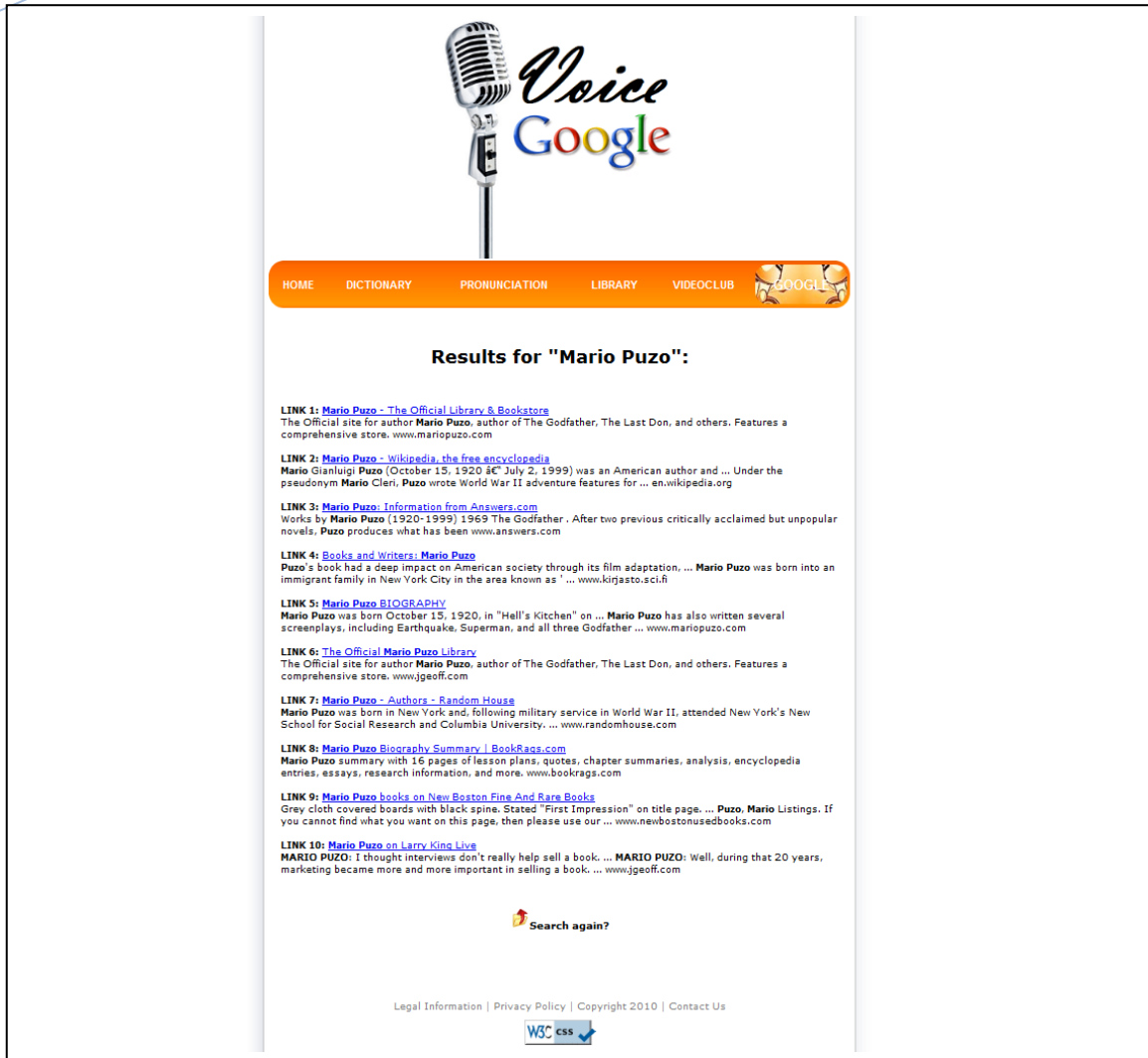
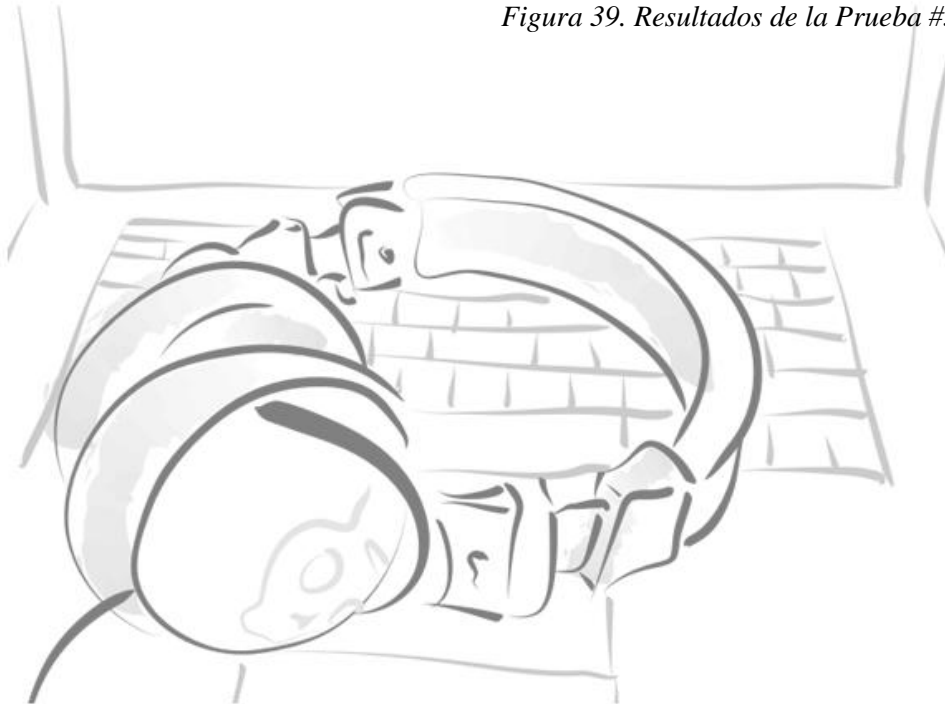
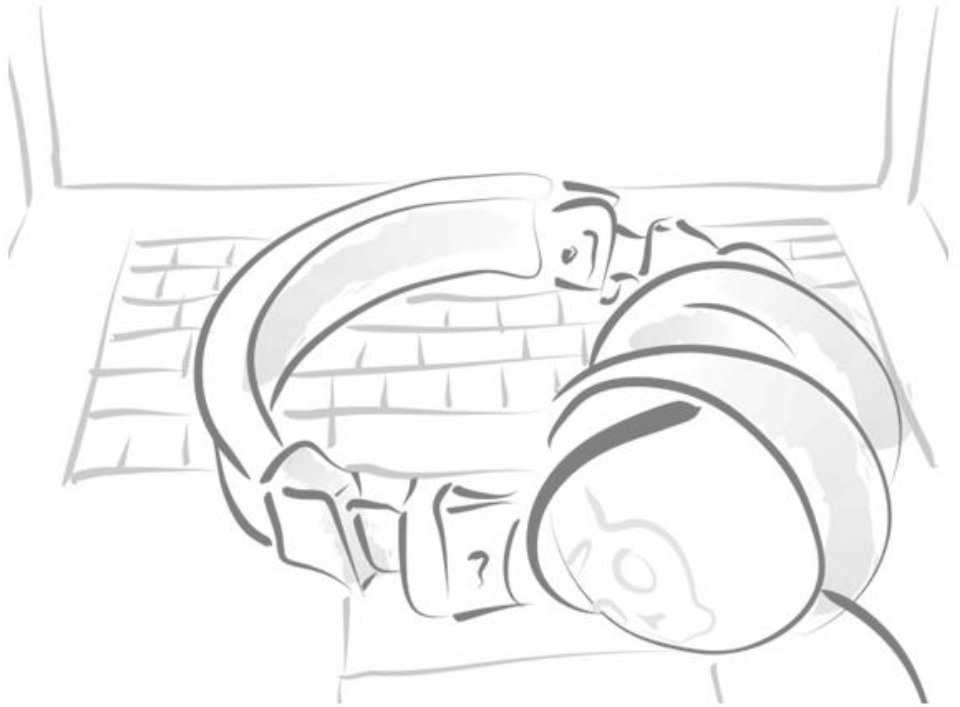


Figura 39. Resultados de la Prueba #3 sobre VG.





# Capítulo 4

## *Conclusiones*

### 4.1 Resumen

El entorno *Voice Applications* (VA), en cada una de las aplicaciones que lo compone, se sitúa como una de las mejores referencias a la hora de realizar el estudio de las tecnologías X+V para la integración de sistemas multimodalidades en unidades unimodales. Sincronizado a través de los eventos de XML, los lenguajes XHTML y VoiceXML forman la tecnología X+V. El primero de los lenguajes se encarga del desarrollo a un nivel más estricto y mejor formado para el diseño visual de entornos Web, mientras que el segundo lenguaje cumple la función de generar sistemas de diálogo a través de interfaces orales que interaccionan con el usuario con el usuario aportando y recogiendo información. De esta forma, integramos en la plataforma VA todo un sistema oral que permite al usuario la navegación y uso de todas y cada una de las aplicaciones únicamente con su voz.

La portada del entorno VA muestra una breve descripción en Inglés de cada una de las aplicaciones que lo componen. El sistema de diálogo oral da la bienvenida al usuario y le invita a navegar por la plataforma. Si el usuario pasa el ratón por encima de cualquiera de los textos descriptivos de las aplicaciones comprobará como estos textos son narrados al usuario instantáneamente. Una vez que el usuario ha elegido la aplicación que desea visitar puede decir su nombre a VA y este redirigirá la página a esa aplicación.

La primera aplicación que nos encontramos es *Voice Dictionary* (VD). Con esta aplicación podemos buscar cualquier contenido de la plataforma Wikipedia con nuestra propia voz y obtener los resultados de manera visual y oral. El usuario debe deletrear el contenido de la búsqueda a través de las instrucciones que se muestran en una etiqueta inferior, o mediante el comando oral “INSTRUCTIONS”. Cada uno de los comandos mostrados, en su mayoría países del mundo, inserta una letra en la barra de búsqueda. Una vez completado el contenido de la búsqueda, y a través del comando oral “SEARCH”, la aplicación envía la información a Wikipedia, recoge los resultados, los procesa y los muestra en una página de resultado. Esta página de resultado narra cada uno de los contenidos encontrados al usuario, pudiendo en cualquier momento ser interrumpido. El usuario puede realizar la función de hipertexto seleccionando cualquiera de los hipervínculos de manera oral, y generando una nueva página de resultado con un nuevo contenido y una nueva narración. En cualquier momento de todo el proceso de búsqueda y resultados de la aplicación VD el usuario puede decir el nombre de cualquier otra aplicación y el entorno redirigirá al usuario a dicha aplicación.

Tras esta aplicación podemos encontrar *Voice Pronunciation* (VP), no sólo una aplicación de diseño multimodal sino todo un divertido juego que ayuda a la correcta pronunciación en lengua inglesa. El usuario accede a una portada dónde puede elegir entre las instrucciones (donde se le mostrarán y narrarán todas y cada una de las instrucciones para el correcto uso de los juegos), el juego de las palabras denominado “*Words*”, y el juego de las imágenes denominado “*Picture*”.



Al acceder al juego de las palabras, ya sea por el enlace visual o a través del comando oral, se muestra al usuario una página con una palabra y definición seleccionada al azar de entre las cerca de doscientas mil palabras almacenadas. La aplicación narra al usuario la definición de la palabra y el usuario debe pronunciarla correctamente. De ser así, obtendrá diez puntos que se suman al total de puntos obtenidos por el usuario, mostrados en la esquina superior izquierda de la página, junto a la puntuación más alta obtenida en este juego. En caso de que el usuario no conozca la pronunciación correcta de la palabra puede acceder a la solución o pasar a la siguiente palabra (todo ello tanto de manera visual como de forma oral). En cualquier momento puede salir del juego volviendo a la página principal de la aplicación VP.

En caso de que el usuario elija el juego de las imágenes se muestra una imagen, en lugar de una palabra, seleccionada de manera aleatoria de una base de datos con más de seiscientas imágenes de todo tipo. El resto de contenidos de este juego se muestran iguales a los del juego de las palabras. Sin embargo, en este caso, además de pronunciar correctamente el nombre de la imagen, el usuario debe adivinarlo, pues este no se muestra en ningún sitio. Igual que en el juego de palabras, el usuario cuenta con las opciones de solución, siguiente, y salida del juego. De una manera sencilla y divertida el usuario puede mejorar su pronunciación y enriquecer su vocabulario en la lengua de Shakespeare.

A continuación, el usuario puede acceder a la aplicación *Voice Library* (VL, dónde se presenta en la página principal una barra de buscador una bienvenida en forma de diálogo. El usuario puede introducir el contenido de la búsqueda en la barra al igual que ocurría en la aplicación VD, y buscar toda la información sobre sus libros preferidos en una base de datos de trescientos libros. Una vez que el usuario recibe los resultados, la aplicación le narra cada una de las características de cada libro. Estos libros son mostrados de forma visual en una tabla con un fondo que aparenta ser una biblioteca, dando un aspecto original y llamativo.

En la aplicación *Voice Videoclub* (VV) tenemos el mismo sistema, pero con una base de datos de más de mil doscientas películas. Al igual que ocurre con las páginas de presentación de VD y VL, aquí el usuario debe deletrear el contenido que desea buscar. La aplicación accede a la base de datos y muestra un listado de cada una de las películas con todos y cada uno de los datos más importantes de esas películas, incluida la imagen del cartel con la que se presentó en los cines, o con la que se publicó para su venta. Todos y cada uno de los datos resultantes son narrados al usuario, pudiendo en cualquier momento realizar una nueva búsqueda, o visitar cualquier otra aplicación.

La última aplicación que forma la plataforma VA se denomina *Voice Google* (VG). Se presenta a modo de buscador, igual que las aplicaciones anteriores. El usuario deletrea de manera oral o teclea el contenido de la búsqueda y VG envía la información al buscador recuperando los contenidos resultantes y mostrándolos en una página de resultado de la aplicación. Los resultados son narrados de forma que se asigna a cada uno de los vínculos hallados un número en el orden en que se encontraron en el buscador. Así, el usuario únicamente debe decir el comando “LINK” y el número del enlace para visitarlo, en lugar de narrarle a la aplicación todo el nombre del enlace, a veces confuso. De esta sencilla forma, obtenemos una aplicación con toda la robustez y eficacia del buscador Google, con la función oral incorporada para poder ser manejada totalmente con la voz.



## 4.2 Trabajo Futuro

Toda la estructura y entorno de la aplicación VA está preparada para dar un paso más, y albergar nuevas funcionalidades. La aplicación VD está diseñada para realizar búsquedas en la plataforma Wikipedia, sin embargo las posibilidades pueden ser ilimitadas al trasladar los métodos y algoritmos a otras plataformas de búsqueda de contenidos, y darles una interfaz oral con la que puedan interactuar un número mayor de internautas. El aspecto visual de la aplicación está diseñado para que el usuario elija la plataforma sobre la que desea realizar la búsqueda, dejando esta funcionalidad para un trabajo futuro. Lo mismo ocurre con la aplicación VG, diseñada para realizar sus búsquedas sobre cualquier buscador online, dando las opciones al usuario para que lo seleccione e integre la voz a su buscador favorito.

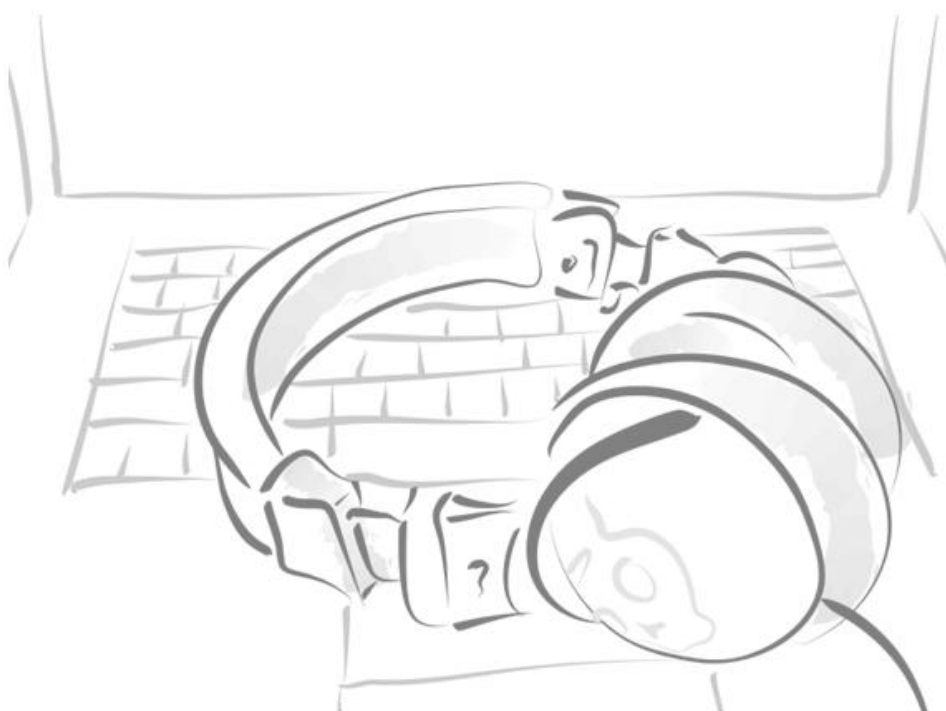
La línea de trabajo futura de la aplicación VP gira en torno a la visión fonética de las palabras y dibujos, y a la ayuda de la interfaz oral. En lugar de responder al usuario con ‘palabra correcta’ o ‘imagen incorrecta’, el sistema debe reconocer la pronunciación exacta del usuario y completar la respuesta con los errores concretos de su pronunciación, indicándole el lugar exacto de la palabra donde ha errado.

Las aplicaciones VL y VV dirigirán sus líneas de trabajo futuras hacia la compra de libros y películas online, a modo de tienda electrónica, en la que puedas ver todas las características de un libro o película, y además puedas comprarlo, viendo y escuchando los comentarios de usuarios que ya los hayan adquirido con anterioridad. Todo este proceso no es nuevo, pero la integración de la funcionalidad de voz desde el inicio de la búsqueda del producto hasta la compra final, sí es innovador. Además de no parar solo en productos como libros y películas, sino en todo tipo de productos, desde una silla de oficina, a unas entradas de teatro, o unos billetes de avión.

Las grandes mejoras en la incorporación de tecnologías X+V vienen determinadas por el desarrollo que ofrezcan los soportes de estas tecnologías. Los navegadores que soportan un sistema multimodal son escasos, contando como única excepción a Opera. Este navegador permite el manejo oral de la interfaz de usuario, sin embargo, pese a su iniciativa, son muchos los errores que deben ser solucionados (desde la dificultad de comunicación con el usuario, hasta la frecuencia con que el programa se cierra sin previo aviso).

Opera ha innovado con la integración de esta funcionalidad, pero las últimas versiones del navegador o bien no permiten la instalación del paquete de datos necesario para utilizarla, o bien no soportan las aplicaciones orales. Actualmente, para el uso oral de internet, es necesaria una versión antigua del navegador Opera, lo que impide el desarrollo puntero de aplicaciones, descartando nuevos avances para los desarrolladores como la utilización de las nuevas plantillas CSS.

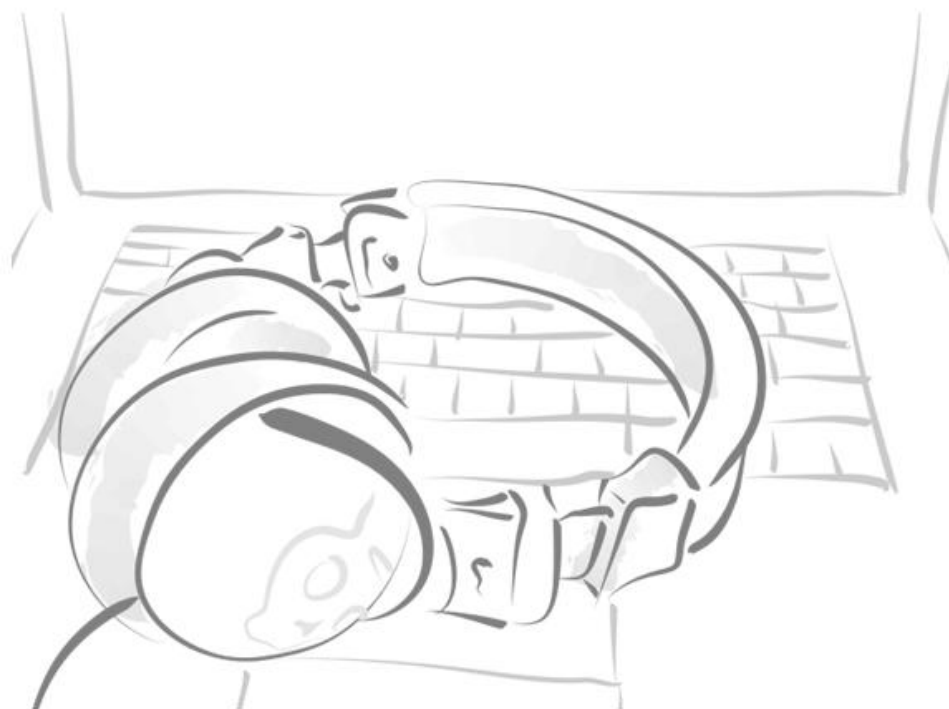
La interfaz oral sólo es soportada por un explorador, siendo un avance futuro la mejora de esa integración en Opera y la inserción de la función oral en el resto de los navegadores, haciendo que esa actualización sea algo útil y necesario. Esta barrera arquitectónica de internet todavía no ha sido saltada, y el desarrollo futuro debe ir en esta línea de trabajo.



## 4.3 Conclusiones Personales

A lo largo del proceso de desarrollo del presente proyecto he disfrutado con todos y cada uno de los retos que se han ido presentando. El afán de superación y las palabras de aliento han logrado que las soluciones no sólo solventasen los retos, sino que fuesen la mejor de las soluciones para realizar un trabajo robusto y eficiente. En cada uno de los retos he aprendido a perseverar sin descanso, logrando además ampliar mis conocimientos.

La creación del entorno me ha enseñado a manejar todos y cada uno de los estándares que se describen en este documento, aportándome nociones imprescindibles para el futuro laboral de un buen desarrollador web.







# Glosario

ABNF	<i>Forma Normal de Backus Aumentada</i>
API	<i>Application Programming Interface</i>
ASCII	<i>American Standard Code for Information Interchange</i>
ATIS	<i>Air Travel Information Services</i>
BBDD	<i>Bases de Datos</i>
BNF	<i>Backus-Naur Form</i>
CAST	<i>Center for Applied Special Technology</i>
CFG	<i>Context-Free Grammar</i>
COM	<i>Component Objetc Model</i>
CORBA	<i>Common Object Request Broker Architecture</i>
CSS	<i>Cascading Style Sheet</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>



DB	<i>Data Base</i>
DOM	<i>Document Object Model</i>
DTD	<i>Definición de Tipos de Documentos</i>
DTMF	<i>Dual Tone Multiple Frequency</i>
ECMA	<i>European Computer Manufacturers Association</i>
FSM	<i>Finite State Machine</i>
GLN	<i>Generación del Lenguaje Natural</i>
HTML	<i>HyperText Markup Language</i>
IETF	<i>Internet Engineering Task Force</i>
IP	<i>Internet Protocol</i>
JSGF	<i>JSpeech Grammar Format</i>
MMI	<i>Multimodal Interaction</i>
MySQL	<i>My Structured Query Language</i>
NLSML	<i>Natural Language Semantics Markup Language</i>
PC	<i>Personal Computer</i>
PDA	<i>Personal Digital Assistant</i>
PHP	<i>Personal Home Page; PHP Hypertext Pre-processor</i>
PLN	<i>Procesamiento de Lenguaje Natural</i>
PML	<i>Phone Markup Language</i>
RAH	<i>Reconocedores Automáticos del Habla</i>
SDO	<i>Sistema de Diálogo Oral</i>
SEM	<i>Semantic Interpretation for Speech Recognition</i>
SGML	<i>Standard Generalized Markup Language</i>
SRGS	<i>Speech Recognition Grammar Specification</i>
SUNDIAL	<i>Speech UNderstanding and DIALog</i>

SVG	<i>Scalable Vector Graphics</i>
TTS	<i>Text-to-Speech</i>
URI	<i>Identificador Uniforme de Recursos</i>
VA	<i>Voice Applications</i>
VD	<i>Voice Dictionary</i>
VG	<i>Voice Google</i>
VL	<i>Voice Library</i>
VoiceXML	<i>Voice eXtensible Markup Language</i>
VP	<i>Voice Pronunciation</i>
VV	<i>Voice Videoclub</i>
W3C	<i>World Wide Web Consortium</i>
X+V	<i>XHTML+Voice</i>
XHTML	<i>eXtensible Hypertext Markup Language</i>
XML	<i>eXtensible Markup Language</i>



# Bibliografía

- [1.1-A] *XHTML+Voice*. Disponible [Internet]:  
<<http://www.w3.org/TR/xhtml+voice/>> [12 de diciembre de 2010]
- [1.1-B] *eXtensible HyperText Markup Language*. Disponible [Internet]:  
<<http://www.w3c.es/divulgacion/guiasbreves/XHTML>> [12 de diciembre de 2010]
- [1.1-C] *Voice eXtensible Markup Language*. Disponible [Internet]:  
<<http://www.voicexml.org/>> [12 de diciembre de 2010]
- [1.1-D] *World Wide Web Consortium*. Disponible [Internet]:  
<<http://www.w3c.es/>> [12 de diciembre de 2010]
- [1.1-E] *Opera*. Disponible [Internet]: <<http://www.opera.com/>>  
[12 de diciembre de 2010]
- [1.1-F] *Speech Recognition Grammar Specification*. Disponible [Internet]:  
<<http://www.w3.org/TR/speech-grammar/>> [12 de diciembre de 2010]
- [1.1-G] Bangalore, S., Hakkani-Tur, D., Tur, G., 2006. Introduction to the Special Issue on Spoken Language Understanding in Conventional Systems. *Speech Communication* 48, 233-238

- [1.1-H] *XHTML+Voice Programmer's Guide*. Disponible [Internet]:  
<[ftp://ftp.software.ibm.com/software/pervasive/info/multimodal/XHTML\\_voice\\_programmers\\_guide.pdf](ftp://ftp.software.ibm.com/software/pervasive/info/multimodal/XHTML_voice_programmers_guide.pdf)> [12 de diciembre de 2010]
- [1.2.3-A] *Web Accesibility Initiative*. Disponible [Internet]:  
<<http://www.w3.org/WAI/GL/>> [12 de diciembre de 2010]
- [1.2.3-B] *Center for Applied Special Technology*. Disponible [Internet]:  
<<http://www.cast.org/learningtools/Bobby/index.html>> [12 de diciembre de 2010]
- [1.3.3-A] *Cascading Style Sheet*. Disponible [Internet]:  
<<http://www.w3.org/Style/CSS/>> [12 de diciembre de 2010]
- [1.3.4-A] *PHP Hypertext Pre-processor*, inicialmente *Personal Home Page Tools*. Disponible [Internet]: <<http://www.php.net>> [12 de diciembre de 2010]
- [1.3.4-B] *My Structured Query Language*. Disponible [Internet]:  
<<http://www.mysql.com>> [12 de diciembre de 2010]
- [1.4.2-A] *What You See Is What You Get*. Disponible [Internet]:  
<<http://es.wikipedia.org/wiki/WYSIWYG>> [12 de diciembre de 2010]
- [1.4.2-B] *Appserv Network*. Disponible [Internet]:  
<<http://www.appservnetwork.com>> [12 de diciembre de 2010]
- [1.4.2-C] *The Apache Software Foundation*. Disponible [Internet]:  
<<http://www.apache.org>> [12 de diciembre de 2010]
- [1.5-A] Larman, C. *Applying UML and Patterns*. Prentice-Hall, 1999.
- [2.1-A] *Desarrollo de Sistemas de Diálogo Oral Adaptativos y Portables: Reconocimiento de Emociones, Adaptación al Idioma y Evaluación de Campo*. Zoraida Callejas Carrión. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Granada. Disponible [Internet]:  
<<http://inf.ucv.ro/~ntand/courses/MMIA222/carrion.pdf>> [12 de diciembre de 2010]
- [2.1.3-A] Traum, D., Larsson, S., 2003. *The Information State Approach to Dialogue Management*. Current and New Directions in Discourse and Dialogue. Kluwer Academic Publisher.
- [2.2-A] Stewart, J., 1922. An electrical analog of the vocal tract. *Nature* 110, 311-312.
- [2.2-B] Turing, A., 1950. Computing machinery and intelligence. *Mind* 236, 433-460.

- [2.2-C] Weizenbaum, J., 1966. ELIZA. A computer program for the study of natural language communication between man and machine. *Communications of the ACM* 9, 36-45.
- [2.2-D] DARPA, 1992. Speech and Natural Language Workshop. Defense Advance Research Projects Agency (DARPA), San Mateo, USA.
- [2.2-E] Kacic, Z., 1999. Advances in spoken dialogue system development. En: Proc. of IEEE International Symposium on Industrial Electronics (ISIE'99). Vil 1. Bled, Slovenia, pg. 169-172.
- [2.2-F] Glass, J. R., 1999. Challenges for spoken dialogue system. En: Proc. of the Automatic Speech Recognition and Understanding Workshop (ASRU'99). Keystone, Colorado, USA.
- [2.2-G] Cheyer, A., Julia, L., 1995. Multimodal maps: An agent based approach. En: Proc. of International Conference on Cooperative Multimodal Cooperation. Eindhoven, Holland, pg. 111-121.
- [2.2-H] Waibel, A., Suhm, B., Vo, M., Yang, J., 1997. Multimodal Interfaces for Multimedia Information Agents. En: Proc. of IEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97). Munich, Germany, pg. 167-170.
- [2.2-I] Wahlster, W. (Ed.), 2006. SmartKom: Foundations of Multimodal Dialogue Systems. Springer. Disponible [Internet]:  
<<http://www.smartkom.org/>> [12 de diciembre de 2010]
- [2.2-J] *MMI Working Group W3C*. Disponible [Internet]:  
<<http://www.w3.org/TR/mmi-arch/>> [12 de diciembre de 2010]
- [2.2-K] Interfaces Conversacionales e Iniciativa Mixta. Cap. 1. Pg. 2. Disponible [Internet]:  
<[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/msp/velazquez\\_s\\_j/capitulo1.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/msp/velazquez_s_j/capitulo1.pdf)> [12 de diciembre de 2010]
- [2.2-L] *W3C "Voice Browser" Working Group*. Disponible [Internet]:  
<<http://www.w3.org/Voice/>> [12 de diciembre de 2010]
- [2.4.1-A] Estándar de VoiceXML del W3C. Disponible [Internet]:  
<<http://www.w3.org/TR/voicexml20/>> [12 de diciembre de 2010]
- [2.4.3-A] Estándar XML *Events* del W3C. Disponible [Internet]:  
<<http://www.w3.org/TR/xml-events/>> [12 de diciembre de 2010]
- [2.5-A] *Spoken Language Interfaces*. Disponible [Internet]:  
<<http://www.cs.cmu.edu/~dbohus/SDS/>> [12 de diciembre de 2010]
- [2.5-B] *Ariadne Spoken Dialogue System*. Disponible [Internet]:  
<<http://www.opendialog.org/>> [12 de diciembre de 2010]

- [2.5-C] *August*. Disponible [Internet]: <<http://www.speech.kth.se/august/>> [12 de diciembre de 2010]
- [2.5-D] *Conquest*. Disponible [Internet]: <<http://www.conquest-dialog.org>> [12 de diciembre de 2010]
- [2.5-E] DINEX. Disponible [Internet]: <<http://groups.csail.mit.edu/sls/applications/dinex.shtml>> [12 de diciembre de 2010]
- [2.5-F] ELVIS. Disponible [Internet]: <<http://www2.research.att.com/~walker/elvis/elvis.html>> [12 de diciembre de 2010]
- [2.5-G] *How may I help you?* Disponible [Internet]: <<http://www2.research.att.com/~algor/hmihy/>> [12 de diciembre de 2010]
- [2.5-H] *Intelligence Procedure Asistance*. Disponible [Internet]: <<http://www.riacs.edu/research/detail/rialist>> [12 de diciembre de 2010]
- [2.5-I] Jupiter. Disponible [Internet]: <<http://groups.csail.mit.edu/sls/research/jupiter.shtml>> [12 de diciembre de 2010]
- [2.5-J] *Language-based Agent for Retrieval of Repair Information*. Disponible [Internet]: <<http://www.cs.cmu.edu/~dbohus/docs/symphony.avi>> [12 de diciembre de 2010]
- [2.5-K] Mercury. Disponible [Internet]: <<http://groups.csail.mit.edu/sls/research/mercury.shtml>> [12 de diciembre de 2010]

