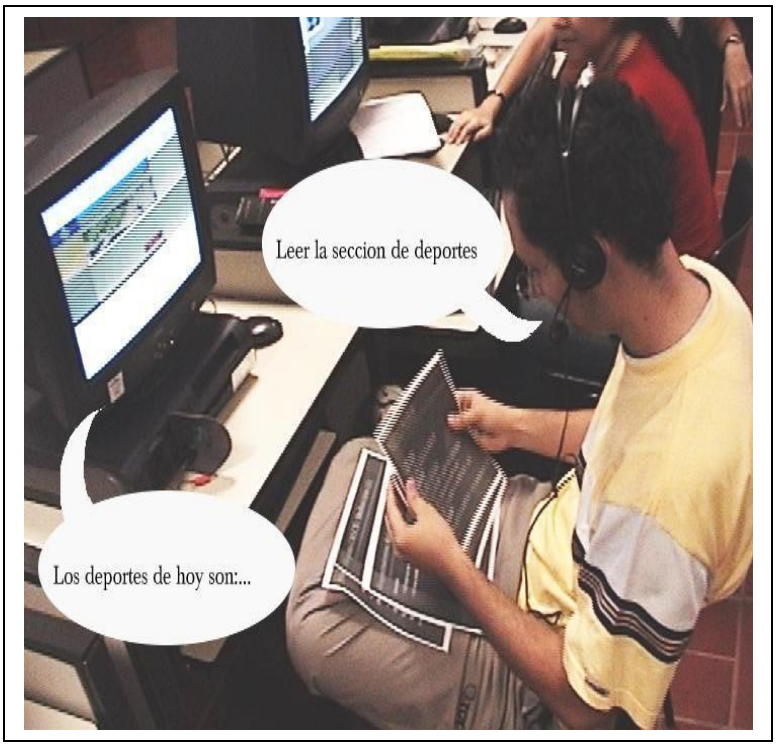
ARQUITECTURA PARA APLICACIONES CON INTERFAZ ORAL



RESUMEN

Este artículo describe una arquitectura para aplicaciones con interfaz oral y hace una evaluación teórica de la misma. También reporta las experiencias obtenidas al implementarla en una aplicación específica: un sistema para navegación básica y lectura de noticias en una página web.

PALABRAS CLAVES

Arquitectura de software, interfaces orales, lingüística computacional, interface humano-computador.

ABSTRACT

This paper presents an architecture for oral interface applications and its evaluation. Besides, it reports the experiences achieved during the implementation of the architecture on a system for basic navegation and news reading from a web page.

KEYWORDS:

Software architecture, oral interfaces, computational linguistics, human - computer interface.

Alina Almira Vázquez Esp.
Profesora de idiomas Español, Inglés y Francés. Traductora e intérprete.
Universidad Javeriana Cali, Universidad del Valle.
Colombia

Gloria Inés Alvarez Vargas. Mag.
Profesora de ciencias de la computación.
Universidad Javeriana Cali
Colombia

Antal A. Buss Molina. Ing.
Profesor de ciencias de la computación.
Universidad Javeriana Cali
Colombia

Diego Luis Linares Ospina. PhD.
Profesor de ciencias de la computación.
Universidad Javeriana Cali
Colombia

Grupo Destino prial@puj.edu.co.
http://prial.puj.edu.co

INTRODUCCIÓN

La estructura interna de las aplicaciones ha venido evolucionando a la par con los paradigmas de programación y las formas de interacción con el computador. En un principio los programas hacían interfaz con el usuario en forma textual, comúnmente basada en menús, ese fue el momento de la programación estructurada y de aplicaciones con arquitectura "serial"; más adelante vinieron las interfaces gráficas, acompañadas de un auge de la programación orientada a objetos y de una arquitectura propia orientada por eventos. Desde el punto de vista de los usuarios finales, este fue un paso adelante en cuanto a facilidad de uso, en particular para las personas novatas en el manejo de aplicaciones computacionales. El siguiente paso puede ser la incorporación del lenguaje natural en la interacción humano-computador; nuevamente, este cambio debe introducir una manera diferente de pensar las aplicaciones de modo que se pueda potenciar el efecto positivo de esta modalidad de interacción. Los beneficios de hacerlo son múltiples: se facilita el acceso al computador de personas con algunas discapacidades, se provee una manera de interacción a distancia sobre sistemas ubicados en sitios remotos, peligrosos o poco saludables, se brinda la posibilidad de utilizar aplicaciones aún teniendo la vista o las manos ocupadas.

Este artículo presenta una propuesta de arquitectura para aplicaciones con interfaz oral en la cual se busca asumir en forma robusta los variados problemas computacionales que se introducen y mostrar una primera aplicación que la implementa. En la sección 2 se presenta una definición de lo que se entiende por aplicación con interfaz oral, en la sección 3 se explican los aspectos más significativos de la arquitectura propuesta, en la sección 4 se reporta el desarrollo de una aplicación para lectura de noticias en una página web que aplica la arquitectura mostrada, en la sección 5 se evalúan tanto la arquitectura como su implementación y finalmente se presentan las conclusiones y trabajos futuros.

2. QUE ES UNA APLICACIÓN CON INTERFAZ ORAL.

Un aplicación con interfaz oral es aquella que realiza entrada/salida en lenguaje natural hablado, es decir, que entiende los comandos que el usuario le dirige oralmente y en lenguaje natural y que emite sus respuestas por el mismo método. Es razonable pensar que este tipo de interacción puede ser más amigable que la tradicional, basada en teclado y pantalla; sin embargo, desde el punto de vista computacional ambas alternativas son significativamente diferentes, como se ilustra en la figura 1.

E/S teclado-pantalla	***E/S oral***
Las operaciones read/write son muy rápidas de ejecutar.	Procesamiento pesado, pues requiere la ejecución de algoritmos complejos.
Las primitivas read/write funcionan para cualquier tipo de dato definido en el lenguaje de programación.	Las primitivas de reconocimiento y entendimiento de habla están limitadas a un vocabulario específico y deben ser entrenadas previamente.
La ejecución de las funciones de entrada salida no introduce ningún error apreciable.	Las primitivas pueden mal-interpretar las elocuciones del usuario.

Figura 1. Paralelo entre entrada salida tradicional y oral

El reto para que las interfaces orales sean percibidas por el usuario como una manera más natural y espontánea de interactuar con el computador, consiste en: asumir y manejar, en forma robusta, las posibles fuentes de error introducidas por los procesos probabilísticos involucrados en el entendimiento de habla; proveer fluidez en el control de diálogo, naturalidad en la generación de lenguaje y claridad de la voz en la síntesis; manteniendo a su vez el tiempo de respuesta cercano al tiempo real.

3. ARQUITECTURA DE UNA APLICACIÓN CON INTERFAZ ORAL.

3.1 Partes del sistema.

La ejecución de una aplicación con interfaz oral se desarrolla como un diálogo entre el usuario y dicha aplicación. Esto implica que a la algorítmica requerida para realizar la tarea del sistema, se deben adicionar otras funcionalidades como: entendimiento de lenguaje natural, generación de lenguaje natural y control de diálogo (ver figura 2). Cada una de estas nuevas funcionalidades incorpora una buena cantidad de retos computacionales a resolver:

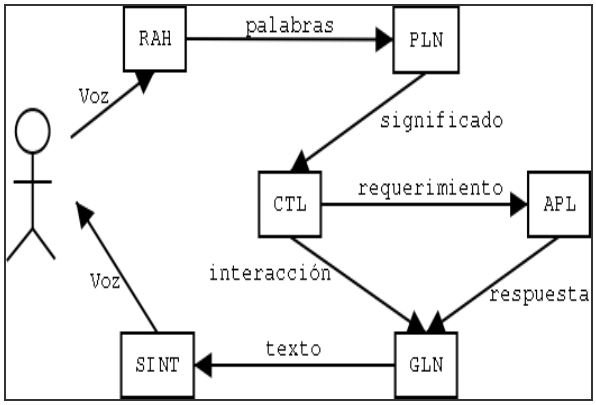


Figura 2. Estructura interna de una aplicación con interfaz oral.

El módulo llamado RAH (Reconocimiento Automático de Habla): se encarga de capturar una señal de voz a través del micrófono y la tarjeta de sonido y produce la secuencia de palabras que con mayor probabilidad pudo dar origen a dicha señal (dentro del vocabulario conocido por el reconocedor). Aquí se debe tratar con una gran cantidad de datos que deben ser procesados para extraer la información relevante que está contenida en ellos. También deben utilizarse modelos sólidos y flexibles, capaces de reconocer la información a pesar de variaciones de volumen, tono, timbre, duración, entonación y acento entre otras y a pesar que los datos pueden llegar incompletos o ruidosos. La técnica por excelencia para resolver este problema son los modelos ocultos de Markov.[1][2][3][4][5][6]

El módulo llamado PLN (Procesamiento de Lenguaje Natural): se encarga de obtener el significado más probable de la secuencia previamente reconocida. Aquí es necesario modelar la inmensa variedad de posibilidades que proporciona el idioma para expresar las ideas y tener la capacidad de distinguir las oraciones con sentido de aquellas que no significan nada. Este problema se hace especialmente interesante cuando se trabaja con habla espontánea, en la cual es común encontrar oraciones gramaticalmente incorrectas que de todos modos transmiten significado. También es necesario establecer un modelo de representación de conocimiento en el cual expresar los significados obtenidos. Las técnicas más comunes para abordar esta tarea son: el análisis sintáctico a partir de gramáticas libres de contexto y los métodos estadísticos basados de nuevo en modelos ocultos de Markov. En cuanto a modelos de representación de conocimiento, es común el uso de estructuras de características (también llamadas "frames").[3][4][5][6][7]

El módulo de Control del diálogo: se encarga de coordinar las elocuciones del sistema con las del usuario de manera que la conversación avance hacia el logro del objetivo del sistema. El reto en este caso consiste en permitir la mayor naturalidad posible en el intercambio oral, para ello es deseable no limitar al usuario en cuanto a las palabras o las estructuras gramaticales a utilizar, tampoco en cuanto al orden en el cual brinde la información. Desafortunadamente, el sistema requiere un esquema de interacción que de alguna manera anticipe las posibles rutas que va tomando el diálogo, esto con el fin de reaccionar adecuadamente en cada contexto. La técnica más común utilizada en esta etapa es el uso de autómatas de estado finito para modelar las posibilidades de interacción y de estructuras de características para modelar el estado del diálogo.[3][4][6][8][9]

El módulo llamado GLN (Generación de Lenguaje Natural) se encarga de redactar un conjunto de párrafos y oraciones que transmitan en forma natural la información que el sistema va a entregar al usuario como respuesta a su solicitud. Esta tarea debe optimizar la selección de estructuras gramaticales y palabras de manera que transmita en forma nítida y sin ambigüedades el objetivo de comunicación. Normalmente se utilizan diversos tipos de gramáticas para ir transformando paulatinamente una secuencia de informaciones representadas en términos computacionales en su equivalente en términos lingüísticos, por ejemplo: gramáticas funcionales o gramáticas de unificación.[4][5][6][10]

El módulo de Síntesis de voz (SINT): recibe un texto en español y lo transforma en una señal acústica generada a través de la tarjeta de sonido y emitida por los parlantes del computador. Los retos más interesantes en esta etapa tienen que ver con el tema de la prosodia; es decir, con la forma de generar una señal acústica que tenga ritmo, acento y que sea capaz de transmitir información como: estados de ánimo, énfasis, ironía y otros aspectos que los humanos incluimos en nuestras emisiones de voz. La técnica más utilizada en la actualidad es el método de concatenación de sonidos.[4][5][6][11][12]

El módulo de Aplicación: implementa la funcionalidad de la aplicación, es decir, los servicios que el sistema va a prestar.

3.2 INTERACCIÓN ENTRE COMPONENTES.

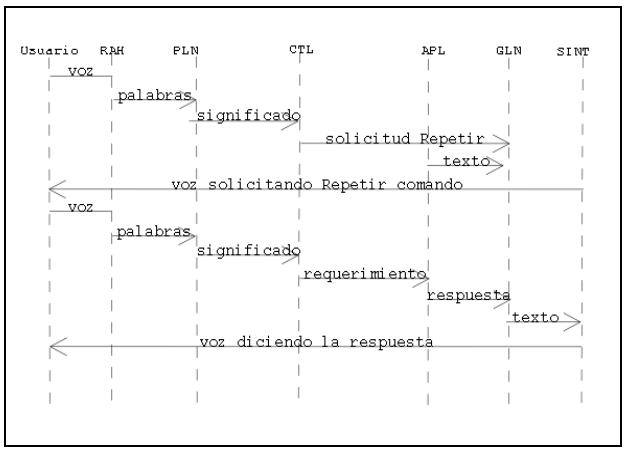


Figura 3.Diagrama de secuencia para un escenario en el cual el sistema requiere repetición por parte del usuario.

La figura 3 ilustra el escenario en el cual el usuario realiza una solicitud al sistema. En esta situación el usuario expresa su solicitud en forma oral, el sistema escucha y entiende la señal acústica emitida por el usuario, evalúa el significado de lo recibido y detecta que no ha entendido la solicitud formulada por el usuario; el sistema solicita que éste le repita, posteriormente el usuario repite la solicitud, el sistema evalúa nuevamente, determina que ha entendido el requerimiento y realiza la acción solicitada, finalmente produce una respuesta oral.

Esta arquitectura refleja, aunque de un modo burdo, los procesos que ocurren dentro de una persona que está interactuando oralmente con otra. Para continuar con esta analogía, se espera que el sistema esté en capacidad de realizar varias operaciones simultáneamente, por ejemplo: escuchar y al mismo tiempo ir analizando la información que se va recibiendo o hablar y escuchar al mismo tiempo. Es por este motivo que se plantea la concurrencia como una característica importante para su implementación; gracias a la concurrencia es posible, por ejemplo, que el usuario interrumpa al sistema mientras éste habla, o que cancele requerimientos antes que ellos sean realizados por el sistema. Básicamente se considera la existencia de tres hilos de ejecución: uno que se encarga de escuchar constantemente y detectar cuando se ha percibido una señal acústica; otro encargado de realizar el procesamiento de lenguaje natural y el control de diálogo, y otro encargado de la síntesis de voz.

4. APLICACIÓN DE LA ARQUITECTURA .

Se ha aplicado esta arquitectura a un caso concreto que es el de una aplicación con interfaz oral para lectura de noticias en una página web. Se ha utilizado para este experimento la página de noticias del diario colombiano El Tiempo (http://eltiempo.terra.com.co) excluyendo las secciones de contenido netamente visual y las revistas adicionales y separatas. La aplicación está en capacidad de entrar al diario, desplazarse entre las secciones y leer los textos completos de las noticias.

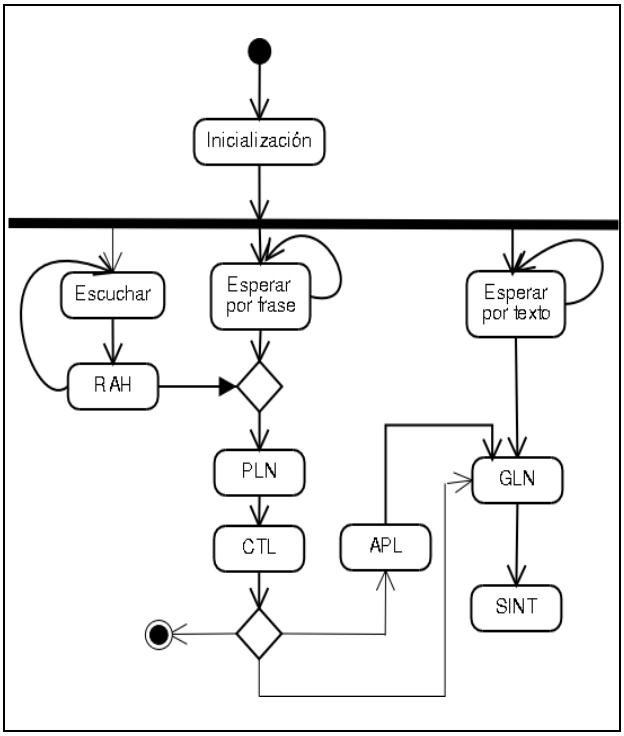


Figura 4. Diagrama de actividad del sistema.

La figura 4 muestra el Diagrama de actividad del sistema. Allí se aprecian tres hilos de ejecución: el de la izquierda se encarga de escuchar y cuando recibe una onda de entrada ejecuta sobre ella el proceso de reconocimiento automático de habla. El hilo del centro se activa cuando se ha reconocido una oración y procede a extraer su significado y evaluarlo, en caso que el significado sea claro, se ejecuta la acción solicitada, en otro caso se interactúa con el usuario para aclarar la solicitud. El hilo de la derecha se encarga de realizar la generación y síntesis de voz.

Para el desarrollo de las tareas propias de la interfaz, se hizo uso de aplicaciones de dominio público, o cuya licencia permite utilizarlas con fines investigativos en forma gratuita. Las herramientas que se utilizaron fueron: para la tarea de reconocimiento automático de habla se usó HTK [14], para hacer análisis sintáctico se usó NLTK [15] y para hacer síntesis de voz se usó Festival [16]. El control de diálogo se implementó desde cero así como el módulo de aplicación. Esto permitió construir en mucho menos tiempo un prototipo funcional.

El control de diálogo le permite a la aplicación reaccionar ante diferentes situaciones como: no haber entendido nada en absoluto o haber entendido parcialmente, en ambos casos el sistema interactúa con el usuario para tratar de aclarar cuál es la solicitud y llevarla a cabo; considerar este tipo de escenarios aumenta la robustez del sistema.

En cuanto a la extracción del contenido de la página web del periódico, se ha optado por traducir la información que está en formato html a un formato rss (RDF Site Summary) que es utilizado en la actualidad por numerosos diarios en el mundo para facilitar la extracción de información de sus páginas (RDF significa Resource Description Framework). Dado que ningún periódico en el país ofrece la información en este formato, se debe pre-procesar la página del diario antes que pueda ser leída por esta aplicación.

5. EVALUACIÓN DE LA ARQUITECTURA

Fortalezas:

- La arquitectura mantiene cada tarea claramente separada de las demás, esto da claridad al código y permite que la arquitectura sea bastante pedagógica en el sentido que ayuda a quien la utiliza a comprender cómo está estructurado un sistema con interfaz oral.
- La arquitectura propuesta permite desarrollar aplicaciones para diversos fines, ya que no ha sido diseñada para un dominio en particular.
- No fue necesario modificar la arquitectura al aplicarla a una tarea específica; aunque esto no demuestra su completitud si es un buen indicio sobre su pertinencia.
- La arquitectura facilita el mantenimiento de las funcionalidades de la aplicación, ya que mantiene esta parte del código claramente separada de la interfaz.
- La arquitectura facilita la evolución del sistema: en caso que se obtengan nuevas formas de realizar cualquiera de las tareas, es fácil reemplazar el módulo siempre y cuando se respete el protocolo de comunicación. Por ejemplo: si se desarrolla un módulo de reconocimiento de habla más perfeccionado, éste puede reemplazar al actual sin afectar el resto de la aplicación.
- Gracias a la facilidad para intercambiar módulos, la arquitectura es una base apropiada para desarrollar tareas de comparación entre métodos para resolver diferentes tareas de la interfaz, es decir, que se pueden llegar a tener varios métodos de control de diálogo disponibles, varios métodos de reconocimiento de habla, etc. y experimentar construyendo sistemas que los combinen de diversas formas.

Debilidades:

- Al mantener cada tarea claramente separada de las demás se pierde la posibilidad de realizar optimizaciones; por ejemplo, hay reportes de mejoras considerables cuando se realizan RAH y PLN en un mismo paso [6]. Esto obviamente causa una mayor complejidad del programa pero mejora la exactitud del reconocimiento.
- Dado que no es posible realizar toda la optimización posible, no se espera un buen desempeño en tareas medianas o grandes (se han obtenido buenos resultados trabajado con aplicaciones de entre 200 y 500 palabras de vocabulario).

Evaluación de la arquitectura desde el punto de vista de la aplicación para lectura de noticias en una página web.

Fortalezas

- Fue posible construir algunos módulos aprovechando herramientas de dominio público disponibles para tareas puntuales como: reconocimiento automático de habla, procesamiento de lenguaje natural y síntesis de voz.
- Existen interfaces claras entre los módulos del sistema, lo cual facilitó la detección del sitio donde ocurrían fallas y por lo tanto su corrección.
- La arquitectura prácticamente sugiere una implementación orientada a objetos, la cual mostró ser muy apropiada para el sistema desarrollado.

Debilidades

- Aunque la aplicación es multilocutor, la tasa de desempeño no es la misma con todos los locutores, esto se debe a que el sistema se ha entrenado con la voz de 6 personas, que son muy pocas para una tarea de este tipo.
- No es fácil reunir en el control de diálogo las mediciones de confiabilidad de la información capturada, ya que esos datos se generan en otros módulos diferentes. En esta versión del sistema se optó por no incluir este tipo de medidas; para poder incluirlas es necesario que las tareas de RAH y PLN interactúen y aporten mediciones cuantitativas al módulo de control.

CONCLUSIONES

Se ha presentado una propuesta de arquitectura para el desarrollo de aplicaciones con interfaz oral, la cual ha sido diseñada en forma independiente de un dominio específico de aplicación.

Entre las principales características de la arquitectura se encuentran: su modularidad y la clara separación que se establece entre la interfaz propiamente dicha y la aplicación. En virtud de esto, el esquema no limita la posibilidad de utilizar diversas técnicas o estrategias para la implementación de los módulos requeridos.

En la arquitectura propuesta es claro que existen algunas tareas independientes del objetivo del sistema, por ejemplo: RAH, PLN y la ejecución del control de diálogo; estas tareas pueden ser reutilizadas en otras aplicaciones siempre y cuando se implementen en forma que sea posible su parametrización sobre un dominio específico. Lo anterior hace pensar que es posible construir una herramienta tipo shell que facilite el desarrollo de aplicaciones con interfaz oral en dominios específicos.

El modelo presentado se adapta a un conjunto importante de aplicaciones con interfaz oral de tamaño pequeño; es decir que aplicaciones que requieran estrategias de control de diálogo altamente especializadas o trabajen sobre volúmenes grandes de vocabulario quedarían por fuera del alcance de esta arquitectura.

Comparando esta arquitectura con el esquema propuesto por VoiceXML [13], se encuentra que en el fondo ambos hacen uso de los mismos elementos, sólo que haciendo énfasis en distintos aspectos: VoiceXML hace explícito el control de diálogo y las funciones propias del dominio de la aplicación, en tanto que oculta las funciones de entendimiento y generación de lenguaje natural, al ubicarlas como primitivas del lenguaje. Por su parte esta arquitectura mantiene todos los elementos en un mismo nivel. Ambas aproximaciones son compatibles, en la medida que un sistema diseñado en la arquitectura puede ser transladado a VoiceXML para su implementación. Para exploración de combinaciones entre técnicas, optimización de métodos para reconocimiento, procesamiento de lenguaje natural y otras tareas de la interfaz, la arquitectura se muestra más apropiada que VoiceXML debido precisamente a que todos los módulos están a un mismo nivel y son explícitas las interfaces entro ellos.

El desarrollo de aplicaciones con interfaz oral no se ha extendido como se esperaría debido, en parte, a las competencias de tipo lingüístico que se requieren para incursionar en este tipo de sistemas.

TRABAJOS FUTUROS

A partir de esta arquitectura es posible diseñar y construir una plantilla de software (“shell”) que facilite el desarrollo de aplicaciones con interfaz oral. Dicha herramienta permitiría la reutilización de buena parte del código necesario para la implementación de la interfaz. Con un diseño cuidadoso, pueden parametrizarse las diferentes etapas de modo que se pueda trazar una frontera clara entre aquello que es propio de la aplicación y el esquema de funcionamiento general. En la actualidad se está trabajando en el diseño e implementación de esta herramienta.

Yendo un paso más adelante se podría llegar a tener un ambiente de desarrollo que permita ensamblar módulos correspondientes a las diferentes partes del sistema; teniendo una apropiada variedad de implementaciones, el ambiente serviría no sólo para crear aplicaciones con interfaz oral sino también para explorar y comparar métodos de solución de las diferentes tareas.

El modelo aún debe ser mejorado, si se le desea utilizar sobre grandes vocabularios; especialmente se debe buscar una forma de integrar módulos para poder realizar optimizaciones. Lo ideal sería poder hacerlo sin perder generalidad, es decir, sin que la

estructura resultante dependa directamente de un dominio de trabajo en particular.

AGRADECIMIENTO

Se agradece a la Facultad de Ingeniería de la Pontificia Universidad Javeriana Seccional Cali, al Fondo Central de Investigación de la Pontificia Universidad Javeriana Seccional Cali, al Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología Francisco José de Caldas COLCIENCIAS y a CINTEL por el apoyo que han brindado para la realización del proyecto *Herramientas para el desarrollo de aplicaciones con interfaz oral, enfocadas al español colombiano*, dentro del cual se ha realizado el presente trabajo.

BIBLIOGRAFIA

[1] RABINER, Lawrence. A tutorial on hidden markov models and selected applications on speech recognition. En IEEE Proceedings Vol 77 Número 2. 1989.
[2] JELINEK, Frederick. Statistical methods for speech recognition. The MIT Press. Ed 3a. 2001. 283p
[3] JURAFSKY, Daniel. MARTIN, James. Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition. Prentice Hall. 2000. 934p.
[4] HUANG, Xuedong. et. al. Spoken language processing: a guide to theory, algorithm and system development. Prentice Hall. 2001. 980p.
[5] COLE, Ronald A. et. al. Survey of the state of the art in human language technology. Center for spoken language understanding. Oregon Graduate Insitute. 1995
[6] LLISTERRI, Joaquim. Memorias del curso: Conversar con el ordenador: el procesamiento del lenguaje y del habla en los sistemas de diálogo. Soria España. 2003.
[7] ALLEN, James Lamar. Natural language understanding. The Benjamin Cummings Company.1995. 645p.
[8] LÖMKER, Frank. http://www.techfak.uni-bielefeld.de/ags/ai/projects/systemDemo/dialog
[9] ALLEN,Rob. Workflow: an introduction. Open Image Systems Inc. United Kingdom Chair, WFMC External relations Committee. 38p.
[10] EHUD,Reiter. Building natural language generation systems. Cambridge Univesity Press. 2000. 248p.
[11] DUTOIT, Thierry. An introduction to text to speech synthesis. Kluwer Academic Publishers. 1997. 285p.
[12] SAGISAKA, Yoshinori. Computing prosody: computational models for processing spontaneous speech. Springer-Verlag. 1997. 401p.
[13] LARSON, James. VoiceXML: introduction to developing speech applications. Pearson education. 2003. 324p
[14] HTK V 3.1 Cambridge Univerity.
[15] LOPPER, Edward. NLTK V 1.0 Universidad de Pennsylvania. 2002
[16] Festival V 1.4.2 Universidad de Edimburgo.

AUTORES

Alina Almira Vazquez. Traductora e intérprete y especialista en Lengua Francesa del Instituto Superior de Lenguas Extranjeras Pablo Lafargue de la Universidad de la Habana, Cuba. 1977. aalmira@puj.edu.co

Gloria Inés Alvarez Vargas. Ingeniera de sistemas de la Universidad Autónoma de Manizales (1990), Magíster en sistemas y computación de la Universidad de Los Andes (Bogotá 1994). Docente investigadora Pontificia Universidad Javeriana Cali. galvarez@puj.edu.co

Antal Alexander Buss Molina. Ingeniero de Sistemas Pontificia Universidad Javeriana Cali (1999) Docente investigador Pontificia Universidad Javeriana Cali. abuss@puj.edu.co

Diego Luis Linares Ospina. Ingeniero de Sistemas Pontificia Universidad Javeriana Cali (19??) Doctor en Reconocimiento de formas e inteligencia artificial. Universidad Politécnica de Valencia (2003) Docente investigador Pontificia Universidad Javeriana Cali. dlinares@puj.edu.co