Apéndice A. Descripción de los Servidores Vocales Interactivos utilizados.

En este apéndice describiremos los Servidores Vocales Interactivos utilizados en la experimentación llevada a cabo en la presente tesis. Para cada uno de los sistemas descritos, comentaremos los detalles más importantes tanto de su estructura e implementación como de las últimas evaluaciones realizadas sobre ellos acerca de la satisfacción de los usuarios.

Los servicios utilizados son los siguientes: el sistema CU Communicator que ofrece información y reserva de billetes de avión, de hotel y de alquiler de coches (Ward y Pellom, 1999; Pellom et al, 2000; Zhang et al, 2001; http://communicator.colorado.edu), un servicio de páginas blancas (San-Segundo et al, 1999; Córdoba et al, 2000; Córdoba et al, 2001), y un servicio de información y reserva de billetes de tren (San-Segundo et al, 2001c; San-Segundo et al, 2001d; San-Segundo et al, 2001e). El primero de ellos en inglés y los dos últimos en castellano desarrollados en el Grupo de Tecnología del Habla (GTH). Para la realización de estos dos últimos sistemas se ha utilizado el entorno de desarrollo de aplicaciones telefónicas TADE (Telephone Application Development Enviroment) implementado íntegramente en nuestro grupo. En el apartado A.2 se describirá en detalle este entorno.

A.1 El sistema Communicator de la Universidad de Colorado: CU Communicator.

Este sistema ha sido desarrollado por el CSLR (The Center for Spoken Language Research) de la Universidad de Colorado, dentro del proyecto DARPA Communicator (http://fofoca.mitre.org). El sistema combina reconocimiento de habla continua, compresión de lenguaje natural y control de diálogo flexible para ofrecer, mediante una interacción natural con el usuario a través del teléfono, información y reserva de billetes de avión, hoteles y coches de alquiler. El sistema se conecta a la página web de una agencia de viajes, de donde extrae la información actualizada.

Uno de los mayores objetivos en el desarrollo de este servicio ha sido su portabilidad a otros dominios. Es decir, la posibilidad de desarrollar nuevos servicios en otros dominios de aplicación, con bajo coste de implementación y sin la necesidad de conocer en profundidad la arquitectura utilizada. Esta portabilidad ha obligado a implementar herramientas adicionales que faciliten la creación rápida de aplicaciones con lenguaje natural. En esta línea cabe comentar el gran esfuerzo que se ha realizado en el desarrollo de herramientas para representar el conocimiento dependiente de la tarea, en ficheros de texto externos al código fuente. Esta representación hace uso de sencillos lenguajes descriptivos, desarrollados también, en el marco de este proyecto.

Este sistema utiliza como estructura base la arquitectura GALAXY-II, desarrollada en MIT, Massachusetts Institute of Technology (Seneff et al, 1998). En el siguiente apartado se describirá brevemente esta plataforma.

A.1.1 Arquitectura GALAXY-II.

En la fase de diseño de esta arquitectura se llevaron a cabo varias reuniones entre expertos en plataformas para la implementación de gestores de diálogos. En estas reuniones se hizo especial hincapié tanto en la estrategia de control del diálogo como en la necesidad de descomponer el sistema en varios servidores. Servidores con gestión independiente pero con una interfaz de comunicaciones muy bien definida. En cuanto a la estrategia de control hubo dos tendencias: en primer lugar los partidarios de una estrategia secuencial hacían valer la mayor facilidad de implementación de esta opción, y por otro lado, se consideraba la posibilidad de una estrategia distribuida basada en tecnología de agentes inteligentes. La solución adoptaba fue una estrategia distribuida en varios módulos (lo que facilitaría su extensión a la tecnología de agentes), en la que las reglas de cada módulo se describieran de forma secuencial. Los principales módulos definidos fueron los siguientes: reconocimiento de voz, comprensión de lenguaje natural, gestión de diálogo, generación de respuesta y conversión texto-voz.

Además de estos módulos se consideraron módulos independientes para la gestión de la información de depuración, el acceso a bases de datos, y el control de los mensajes entre los diferentes módulos (HUB). En la figura A-1 podemos ver la arquitectura de módulos (Seneff et al, 1998). Al lado de cada módulo (en cursiva), aparecen los nombres de los servidores desarrollados en MIT.

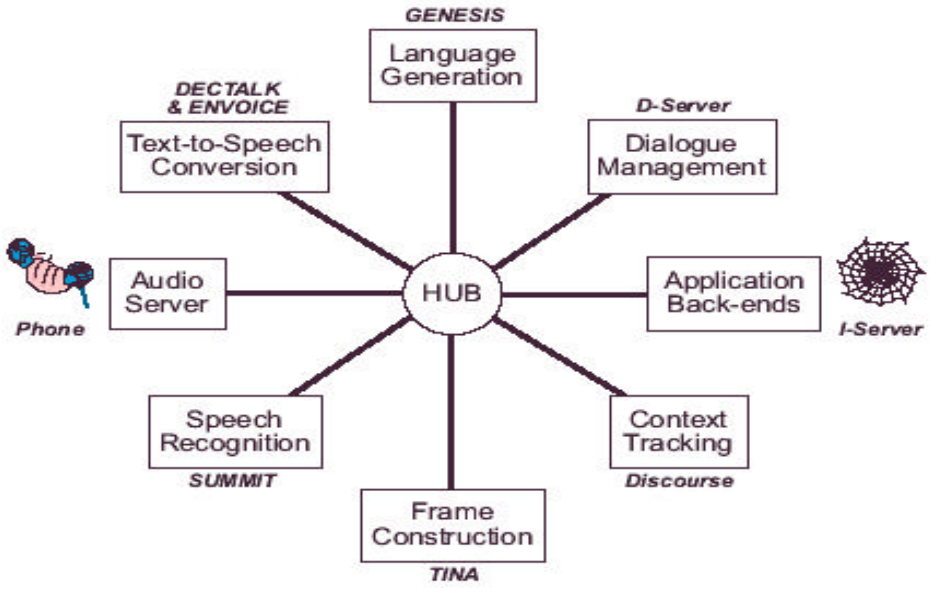

Figura A-1: Módulos de la arquitectura GALAXY-II (Seneff et al, 1998).

La interacción entre el HUB y los módulos se controla mediante un lenguaje descriptivo de alto nivel. El HUB dispone de una descripción acerca de los módulos. En esta descripción se detalla una lista de los módulos/servidores existentes, el nombre de la máquina donde se ejecuta cada servidor, un puerto de comunicaciones y el conjunto de acciones que se pueden hacer en cada servidor. Además se deben especificar,

también en el HUB, una serie de reglas que activan la ejecución de las funciones de cada módulo o servidor. La ejecución del sistema se puede activar desde una interfaz visual o desde un servidor concreto, como por ejemplo cuando se detecte una llamada telefónica en el servidor de audio (Audio Server).

Para la comunicación del HUB con los servidores se utiliza un protocolo de comunicaciones sencillo. En la inicialización, el HUB envía un mensaje de bienvenida a todos los servidores para comprobar que están funcionando correctamente. Después de esta inicialización, el HUB entra en un bucle de espera. En esta situación, los servidores pueden enviar comandos para activar las reglas definidas en el HUB. Estos comandos pueden ir acompañados de parámetros. En la primera versión, un comando sólo puede activar una regla, limitación que se podría extender en el futuro, permitiendo que se ejecuten varias reglas para realizar procesamiento en paralelo. La ejecución de una regla puede llevar asociada la necesidad de ejecutar alguno de los servidores, lo que obliga al HUB a emitir un comando para solicitar el servicio necesario. Este comando también puede tener ciertos parámetros que serán un subconjunto de los parámetros pertenecientes al comando que activó la regla. De igual forma, el servidor solicitado emitirá un comando de respuesta con ciertos parámetros de salida, como por ejemplo la frase reconocida si se tratase del servidor de reconocimiento de habla.

A parte de la arquitectura y de los módulos comentados, también están disponibles una serie de herramientas complementarias de depuración y monitorización que facilitan el desarrollo de nuevos servicios. En primer lugar es posible la ejecución paso a paso, permitiendo zonas de parada entre cada una de las reglas que se van ejecutando. Al mismo tiempo se va almacenado, en un fichero de depuración, los valores de las diferentes variables/parámetros involucrados en cada paso de la ejecución. Además se van grabando las locuciones pronunciadas tanto por el sistema como por el usuario. Por otro lado, el HUB dispone de la posibilidad de manipular su evolución mediante funciones de entrada y salida asíncronas. Como ya hemos comentado, la comunicación entre servidores se realiza principalmente a través del HUB pero no siempre es así. El caso típico ocurre entre el servidor de audio y el sistema de reconocimiento. El servidor de audio, en primera instancia, informa al HUB de la existencia de un conjunto de muestras de voz que se deben decodificar. Es el HUB el que se encarga de buscar un reconocedor libre e informarle del puntero al buffer donde se están grabando las muestras. A partir de ese instante la comunicación entre ambos servidores es directa y se realiza través de ese buffer.

Por último, un detalle importante que comentan sus autores (Seneff et al, 1998) es la consistencia semántica entre los diferentes servidores. Cuando se trabaja con una arquitectura común, de la que se pueden reutilizar ciertos módulos, los nuevos servidores introducidos deben utilizar las mismas representaciones semánticas que se utilizaban en los módulos reutilizados, para que el sistema en su conjunto funcione correctamente. Por ejemplo, el gestor de diálogo (Dialogue Manager) debe prever una serie de conceptos que le llegarán del analizador semántico (Frame Construction) para definir la estrategia del diálogo. Esta representación de los conceptos semánticos debe ser la misma para que su interpretación sea coherente.

A.1.2 Servidores del sistema CU Communicator.

El sistema CU Communicator está formado por un HUB y siete módulos o servidores que pasamos a comentar en detalle:

- **Servidor de Audio:** este módulo es el encargado de gestionar tanto la tarjeta de sonido para la grabación y reproducción de voz, como la tarjeta de interfaz con la línea telefónica para realizar la gestión de la llamada; detección de llamada entrante, descolgar, colgar,... En una primera versión se utilizó el servidor desarrollado en MIT pero en la actualidad se está trabajando en el desarrollo de un nuevo módulo que permita la interrupción del sistema por voz (barge-in) y utilice un hardware comercial en el que estén integradas las tarjetas de audio y de interfaz de línea en una sola tarjeta. En la actualidad se está trabajando sobre una tarjeta Dialogic.

- **Reconocimiento de voz:** en este módulo se utiliza el reconocedor de la Universidad Carnegie Mellon (Ravishankar, 1996). Este reconocedor está basado en modelos ocultos de Markov continuos para modelar trifonemas. El modelo de lenguaje utilizado es una gramática probabilística de tipo 3-gram basada en clases semánticas. El servidor de reconocimiento recibe la voz directamente del módulo de audio. El decodificador ofrece a su salida un grafo de palabras de cuál se extrae la mejor hipótesis para ser analizada por el módulo de comprensión. En futuros desarrollos se ha cambiado esta estructura para que se realice un análisis semántico sobre el propio grafo de palabras directamente con el fin de aumentar la robustez del parser (Hacioglu y Ward, 2001).

  Otro cambio importante en este módulo ha sido la adaptación de los modelos acústicos a la tarea concreta y el entrenamiento de modelos acústicos diferentes para hombre, mujer y señal procedente de teléfonos móviles. Cada uno de estos tipos de modelos acústicos, da lugar a un reconocedor diferente. En el sistema CU Communicator se activan todos ellos de forma paralela. La selección de la mejor hipótesis se puede hacer siguiendo varios criterios: mejor verosimilitud por trama o mediante una clasificación previa del sexo o del tipo de teléfono. En la presente tesis se propone la utilización de medidas de confianza para la combinación de hipótesis provenientes de diferentes reconocedores.

- **Analizador semántico:** en este servidor se utiliza una versión modificada del sistema Phoenix (Ward, 1994). La labor de este módulo es traducir la salida del reconocedor en una secuencia de conceptos semánticos o plantillas. Una plantilla no es más que un conjunto de slots que representan trozos de información relacionados entre sí. Cada slot tiene asociada una gramática de contexto libre que determina el conjunto de patrones de palabras que hacen referencia o rellenan ese slot. Estas gramáticas se compilan en forma de redes de transición recursivas (RTN: Recursive Transition Networks). El sistema Phoenix ha sido modificado para que produzca, además de los slots y sus respectivos valores, la estructura conjunta que los relaciona.

- **Gestor de diálogo:** es un gestor guiado por eventos en el cual se utiliza fundamentalmente la información sobre el estado actual del sistema para decidir la siguiente acción a realizar. No se necesita un script para definir el flujo de estados, sino que el gestor actúa sobre la representación semántica de la tarea y la información de la situación actual para decidir el camino a tomar. El gestor de diálogo recibe la información semántica del analizador y la interpreta según el estado actual. El estado actual del diálogo queda especificado por un conjunto de datos o estructuras en C que se generan automáticamente a partir de formularios descritos en ficheros de texto. Estos formularios deben ser diseñados por el desarrollador del gestor. Cada formulario está formado por varios campos que contienen información sobre un dato concreto de la tarea ej: ciudad origen o destino de una viaje. Los campos pueden ser: nombre del dato, valor, slot asociado del analizador semántico y punteros a diferentes templates para la generación de respuesta. En estos templates se representa un trozo de la frase en la que iría insertada el dato correspondiente. Mediante la concatenación de estos templates se pueden generar las frases de lenguaje natural que dice el sistema al usuario, o las consultas SQL para acceder a la información de la base de datos.

  Esta arquitectura funciona como si fuera un sistema de producción automático que sigue un conjunto de reglas muy sencillo. Los slots obtenidos de la frase del usuario analizada semánticamente, hacen que se modifiquen las variables de estado del diálogo. Una vez modificado este estado, el gestor lo interpreta y decide la siguiente acción a realizar. El conjunto de acciones posibles, ordenadas por prioridad, son las siguientes:

  - Clarificación: si la interpretación semántica es incierta o ambigua, la siguiente acción será pedir al usuario que concrete o deshaga la ambigüedad.

  - Marcar la tarea como completada: en ese caso ofrece un resumen de la información ofrecida y cuelga.

  - Búsqueda de la información solicitada y presentación al usuario: si hay suficiente información para hacer una consulta a la base de datos, se realizará la consulta y se presentará al usuario los resultados obtenidos.

  - Petición al usuario de más información: esta acción se realiza cuando no se dispone de la información suficiente para hacer un acceso a la base de datos.

  Las reglas para decidir qué dato preguntar, son bastante sencillas. Asociando cada objetivo o parte del servicio a una plantilla, podemos comprobar fácilmente los slots que no han sido aún rellenos e ir preguntando al usuario de mayor a menor prioridad.

  Este sistema de gestión de diálogo permite iniciativa mixta: el sistema va haciendo preguntas para completar los slots de la plantilla u objetivo activo en cada momento, pero el usuario puede contestar e ir rellenado esos slots al ritmo y en el orden que mejor le parezca.

- **Servidor de Base de Datos y acceso a Web:** la interfaz con la información del servicio se realiza a través de una base de datos SQL y un conjunto de scripts en Perl para la obtención de la información de Internet. Las consultas en lenguaje SQL se generan por el gestor de diálogo mediante la concatenación de templates obtenidos de los formularios de los datos/slots. Los datos involucrados en cada consulta serán aquellos datos necesarios para satisfacer el objetivo o plantilla solicitada por el usuario.

  El acceso por Internet puede llevar asociado cierto retardo en la obtención de la información, lo que obliga a imponer una limitación del tiempo, superada la cual, se debe informar al usuario.

- **Conversor Texto-Voz:** en este proyecto se han desarrollado dos servidores para realizar la conversión texto-voz. El primero de ellos está basado en el sistema de conversión denominado FESTIVAL (Taylor et al, 1998). Debido a las posibilidades de este conversor texto-voz, el servidor permite configurar diferentes parámetros de síntesis como la calidad de la voz, la velocidad de locución e incluso el idioma (inglés, español y alemán).

  Por otro lado, se ha desarrollado otro sintetizador basado en concatenación de unidades con tamaños muy variables. Este sintetizador puede trabajar con unidades desde difonemas hasta frases completas. En el proceso de síntesis de la voz, el texto se divide automáticamente en frases individuales, haciendo la conversión de grafema a fonema para cada una de ellas. La selección de unidades a concatenar se decide mediante un algoritmo de búsqueda, Viterbi, realizado sobre el espacio formado por todas las unidades. La función de coste considerada incluye información del contexto fonético, el pitch, duración y amplitud de la señal. Se ha incorporado una distancia espectral basada en los parámetros LSF (Line Spectral Frequency) (Pellom y Hansen, 1998). Los segmentos de voz obtenidos del alineamiento, se concatenan para generar la señal de voz que se reproducirá.

- **Servidor de preferencias del usuario:** se ha desarrollado un servidor para la captura de características o preferencias del usuario. Este servidor permite, a través de una página web, que el usuario pueda definir sus criterios preferidos a la hora de ordenar las diferentes posibilidades de viaje en avión o de reserva de hotel y coche (precio, duración, confort,...), el tipo de asiento o comida preferida, la compañía aérea o empresa de alquiler de coches preferida, e introducir información personal como la dirección de correo electrónico o el número de teléfono. En este servidor, el usuario también puede particularizar parámetros del generador de respuesta o del sintetizador; velocidad y pitch.

A.1.3 Captura de datos.

El centro CSLR (The Center for Spoken Language Research) mantiene un sistema de demostración activado constantemente. Desde Octubre de 1999 hasta Junio del 2001 se han capturado más de 3.000 llamadas telefónicas de alrededor de 1.000 locutores

diferentes dando lugar a un número de frases superior a las 30.000. Estas frases, así como los diálogos generados en las llamadas, son grabadas y transcritas manualmente de forma casi paralela. Con estos nuevos datos se van ajustando tanto los modelos acústicos como los modelos de lenguaje utilizados en el reconocedor, para ir realizando un proceso iterativo de mejora y test del sistema. Para registrarse como usuario y acceder al sistema se puede consultar la página web http://communicator.colorado.edu.

A.1.4 Evaluación por NIST en Junio 2000

Aunque el sistema está en constante evolución y mejora, presentaremos los resultados de la evaluación realizada en Junio de 2000 a cargo de NIST (National Institute of Standards) para tener una idea de su funcionamiento. En esta evaluación 72 llamadas realizadas por usuarios diferentes, fueron recogidas a lo largo de este mes. Del total de llamadas, 44 fueron realizadas por mujeres y 28 por hombres. Tras el análisis de las conversaciones se obtuvo que 53, de las 72 llamadas (73,6%), se completaron correctamente ofreciendo la información solicitada por el usuario. Los errores más frecuentes se produjeron debido a fallos en el servidor de audio y a problemas en el reconocimiento de ciudades con nombres muy parecidos (en inglés) como Austin/Boston o Asheville/Nashville.

La tasa de error (Word Error Rate) obtenida a lo largo de las 1264 frases de evaluación recopiladas fue de 26,2%. En estos casos los mayores problemas surgieron porque los usuarios hablaban antes de que el sistema comenzase a grabar.

El tiempo medio por llamada fue de 260 segundos (4 minutos y 20 segundos) dando lugar a 20 interacciones sistema-usuario (pregunta-respuesta) en media. Las medidas de evaluación subjetiva por parte de los usuarios se recogen en la tabla A-1.

Pregunta realizada el usuario	Media	CU
Fue fácil conseguir la información deseada	3.1	3.9
Fue fácil comprender lo que el sistema dice	3.8	4.5
Sabía lo que podía decir o hacer en cada punto del diálogo	3.5	4.2
El sistema funcionó como me esperaba	3.1	3.7
Me gustaría utilizar este sistema regularmente	3.6	3.4
Valor medio obtenido	**3.2**	**3.9**

Tabla A-1: Representación de los valores de subjetividad obtenidos para la media de los grupos involucrados en este proyecto (Media) y para el sistema de CU (CU). La satisfacción del usuario se codifica con un valor entre 1 (completamente en desacuerdo) y 5 (completamente de acuerdo).

Como se puede ver de la evaluación, el sistema desarrollado en la Universidad de Colorado (en el CSLR) fue considerado como uno de los mejores sistemas obtenidos en el marco del proyecto DARPA Communicator.

El trabajo desarrollado en esta tesis en el marco de este Servidor Vocal Interactivo ha estado relacionado con la obtención de medidas de confianza al nivel de palabra, concepto semántico y frase (ver capítulo 5). Este análisis de las medidas de confianza tiene el objetivo de mejorar la gestión del diálogo por un lado, y por otro, considerar estas medidas como heurístico para combinar hipótesis de diferentes reconocedores (con modelos acústicos diferentes).

A.2 TADE (Telephone Application Development Enviroment)

El sistema TADE es un entorno para el desarrollo de Servidores Vocales Interactivos. Este entorno proporciona un lenguaje descriptivo, con ciertas primitivas de alto nivel, para el diseño de aplicaciones telefónicas. Además incluye diversas utilidades necesarias para cubrir todo el ciclo de vida de una aplicación: diseño, compilación y ejecución (Casas, 1997; Valín, 2000; Martínez, 2000; López, 2001; Heras, 2002).

Este entorno funciona sobre el sistema operativo Windows 95 o superior, y permite la ejecución simultánea de dos aplicaciones en líneas telefónicas independientes. Cada una de las líneas necesita de una tarjeta de sonido, que puede ser cualquier tarjeta comercial compatible con el sistema operativo Windows, y una tarjeta de interfaz con la línea telefónica para lo que se debe utilizar la tarjeta desarrollada en nuestro departamento. Para ver más detalles acerca de su instalación se puede consultar los proyectos fin de carrera de J. Martínez y J. López (Martínez, 2000; López, 2001).

A.2.1 El entorno TADE

A continuación se describen las principales utilidades disponibles en el entorno de desarrollo:

- **Edición de texto:** el entorno dispone de un editor en el que se escribe la aplicación a desarrollar en el lenguaje de alto nivel. Esta herramienta dispone de todas las utilidades necesarias en cualquier editor de texto: cortar, pegar o eliminar texto seleccionado, cambiar el formato del texto o el tipo de letra, opciones de búsqueda, impresión, etc. Por otro lado, la herramienta de edición permite gestionar ficheros de texto con formato RTF (Riched Text Format): creación de un fichero nuevo, apertura de varios ficheros simultáneamente, y diferentes opciones para guardar o salvar el contenido de los ficheros.

- **Compilación de la aplicación:** una vez escrita la aplicación es necesario compilarla. Con esta utilidad el entorno hace una comprobación de la sintaxis del lenguaje escrito y genera un fichero binario con la máquina de estados que representa la aplicación desarrollada. En caso de detectar algún error, el sistema avisa al usuario describiendo el tipo de error y la línea de código en la que se produjo. Esta herramienta es muy importante para evitar muchos de los errores que aparecen en la fase de aprendizaje del lenguaje.

- **Ejecución de la aplicación:** una vez compilada correctamente la aplicación se puede ejecutar con el entorno. Al seleccionar esta opción, en el menú principal aparece una ventana dividida en dos partes, cada una de ellas referida a una línea telefónica. Para cada línea se dispone de una pantalla en la que van apareciendo las instrucciones que se van ejecutando. Además, se dispone de tres botones con los que se puede cargar una aplicación determinada, ejecutarla y pararla. Otros dos botones adiciones ofrecen la funcionalidad de simular el colgado o descolgado de la línea telefónica lo que es de gran ayuda en el período de depuración.

- **Entorno de depuración:** otra utilidad muy importante es el entorno de depuración. En este entorno se pueden introducir puntos de ruptura en el código y ejecutar la aplicación hasta ellos, se puede ejecutar hasta el cursor, realizar una ejecución instrucción a instrucción o ejecutar normalmente. En cualquiera de los modos comentados es posible visualizar e incluso modificar las variables del sistema en tiempo de ejecución. Este hecho permite modificar el comportamiento de la aplicación sin necesidad de compilarla de nuevo.

- **Grabación de mensajes de voz:** cualquier aplicación telefónica necesita de la reproducción de mensajes de voz, bien con preguntas a formular al usuario, o bien con información útil para el usuario. Para la grabación de estos mensajes, el entorno ofrece una herramienta que analiza la aplicación desarrollada y extrae los ficheros que deben ser grabados para su correcto funcionamiento. Esta herramienta mantiene dos listas activas de ficheros (ficheros grabados y no grabados) que va gestionando automáticamente. Una vez grabado un fichero se puede escuchar para comprobar su calidad pudiendo ser borrado en caso de mala grabación.

- **Generación de diccionarios:** las instrucciones de reconocimiento de voz, que comentaremos más adelante, hacen uso de diccionarios de reconocimiento con las palabras o expresiones que se quieren reconocer. Estos diccionarios contienen tanto las palabras como su transcripción alofónica. Esta herramienta permite, a partir de un texto cualquiera, generar un diccionario con todas las palabras contenidas en él. A la hora de generar el diccionario se debe elegir el formato deseado según el reconocedor que se vaya a utilizar.

- **Reconocimiento desde fichero:** esta utilidad permite hacer un reconocimiento de voz off-line de un fichero grabado anteriormente. De esta forma nos permite depurar el proceso de reconocimiento de voz. En esta herramienta se debe seleccionar tanto el reconocedor a utilizar como el diccionario considerado.

A.2.2 Funcionalidad del Lenguaje en el entorno TADE

De forma general, una aplicación telefónica especificada en el lenguaje consta de las siguientes partes:

- *Definición e inicialización de variables globales*: donde se definen e inicializan las variables globales a toda la aplicación. Estas variables pueden ser accedidas desde cualquier trozo de código, incluyendo el cuerpo de una subrutina.

- *Tratamiento de errores*: en esta zona se indican las acciones a realizar por el sistema cuando se produzcan algunos de los errores considerados: máximo tiempo excedido sin que el usuario conteste, número máximo de intentos fallidos por parte del usuario, el usuario cuelga antes de finalizar la aplicación o cuando se produce algún error general del sistema.

- *Subrutinas*: donde se definen e implementan las subrutinas de las cuales se hará uso en la aplicación. En la última versión del lenguaje (López, 2001), estas subrutinas permiten la definición de variables locales a su entorno, así como el paso de parámetros de entrada y de salida.

- *Aplicación*: que constituye el conjunto de instrucciones que se irán ejecutando de forma secuencial. El lenguaje diseñado ofrece principalmente sentencias de gestión de la línea telefónica (colgar, descolgar, marcar o esperar llamada) y sentencias de voz (reconocimiento, conversión texto a voz, reproducción y grabación). También ofrece primitivas para el acceso a bases de datos (abrir/cerrar bases de datos y ejecutar consultas), para el envío de correo electrónico, la gestión de directorios y archivos, el manejo de cadenas y operaciones aritméticas básicas. Otro tipo de funciones disponibles son aquellas desarrolladas para la intervención de un operador humano y para el manejo de matrices de datos.

```
SECCION_VARIABLES:
n_max_duracion_llamada = 40;    /*En minutos */
s_variable_de_tipo_string = "...";
SECCION_ERRORES:
 TRATAMIENTO NO_RECONOCIDO     sintetizar("Opción errónea";);   reintentar;
   FIN_TRATAMIENTO
 TRATAMIENTO LONGITUD_CORTA   sintetizar("Cadena corta.";);      reintentar;
  FIN_TRATAMIENTO
 TRATAMIENTO TIMEOUT                  sintetizar("Vuelva a repetir";);   reintentar;
 ...
SECCION_SUBRUTINAS:
 SUBRUTINA Bienvenida(s_frase_a_pronunciar;):
 sintetizar(s_frase_a_pronunciar;);
 retornar:
 FIN_SUBRUTINA
SECCION_APLICACION:
INICIO: /* Comienzo de la aplicación */
          esperar_llamada(;);
            s_variable_de_tipo_string = "hola a todos";
          gosub Bienvenida(s_variable_de_tipo_string;):;
          colgar(;);
FIN: /* Fin de la aplicación */
```

Figura A-2: Ejemplo de aplicación en el lenguaje del entorno TADE.

Además de las instrucciones comentadas anteriormente, el sistema ofrece análisis del progreso de llamada (*CPA: Call Progress Analysis*) configurable, redirección de llamadas dentro de una misma centralita (Alcatel o Ibercom) y la activación de un hilo musical en caso de accesos largos a bases de datos. En la figura A-2, podemos ver un ejemplo sencillo de aplicación.

El entorno dispone de tres primitivas para el reconocimiento de voz: una para reconocer los dígitos y algunos comandos de control, otra para reconocer habla aislada o expresiones cortas, y otra primitiva para el reconocimiento de nombres deletreados de forma continua:

- **Reconocimiento de dígitos:** esta función permite reconocer los dígitos pronunciados de forma aislada y algunos comandos de control como "sí", "no" "ayuda" y "cancelar". Este reconocedor utiliza modelos acústicos de palabra HMM discretos y el algoritmo de Viterbi para la comparación de los modelos acústicos con el dígito o palabra pronunciada (San-Segundo, 1997).

- **Reconocimiento de expresiones cortas:** esta función realiza el reconocimiento de la palabra o expresión pronunciada por el usuario de entre las pertenecientes a un diccionario determinado (hasta 10.000 palabras o expresiones). Como resultado del reconocimiento se ofrece una lista ordenada de candidatos. El reconocimiento se realiza en dos etapas: hipótesis (preselección) y verificación (reconocimiento más detallado) (Macías-Guarasa et al, 1996b; Ferreiros et al, 1998; Macías-Guarasa et al, 2000a; Córdoba et al, 2001).

  - *Hipótesis o preselección:* el módulo de preselección consiste en un conjunto de algoritmos de reconocimiento poco costosos a través de los cuales se permite reducir la lista de candidatos a reconocer. Esta reducción en los candidatos permite aplicar después, técnicas de reconocimiento más potentes sin aumentar excesivamente el tiempo de proceso. El módulo de preselección consta de dos partes: *Extracción de la cadena fonética (decodificador acústico)* en la que se utiliza el algoritmo de One-pass para obtener la secuencia de alófonos que mejor se ajusta a la secuencia de observaciones acústicas recibidas. Los modelos probabilísticos utilizados para la decodificación acústica son modelos de alófono HMM semicontinuos (SCHMM). La segunda parte es el *Acceso léxico*, que se trata de un algoritmo de programación dinámica que recibe a su entrada una cadena de unidades fonéticas (alófonos), y genera a su salida una serie de palabras candidato, ordenadas de menor a mayor distancia con los nombres o expresiones del diccionario.

  - *Verificación:* en esta etapa se escoge un número reducido de candidatos (10%-15% del tamaño del diccionario) que obtuvieron una menor distancia en la etapa anterior, y se realiza un proceso de reconocimiento, basado en el algoritmo de Viterbi, utilizando modelos más potentes, modelos de palabra HMM-continuos (resultado de la concatenación de los modelos de alófono contextuales correspondientes).

La finalidad de utilizar estas dos etapas es la de acelerar el proceso de reconocimiento. Esta característica es más importante cuanto mayor sea el tamaño del diccionario a tratar.

- **Reconocimiento de nombres deletreados:** esta función realiza el reconocimiento de una secuencia de letras correspondiente a un nombre de un diccionario. La estructura de reconocimiento es parecida a la comentada anteriormente para el caso de habla aislada o expresiones cortas. Este reconocedor es la versión en tiempo real del desarrollado en la presente tesis y cuya descripción detallada viene recogida en el capítulo tres o en los trabajos realizados por San-Segundo (San-Segundo et al, 2000b; San-Segundo et al, 2002).

La base de datos de voz utilizada para entrenar los modelos acústicos en el caso del reconocedor de dígitos fue grabada en el GTH, mientras que para el caso del reconocedor de expresiones y el de nombres deletreados se ha utilizado la base de datos SpeechDat (Moreno, 1997).

Para realizar el proceso de conversión de texto a voz se dispone de un sistema de síntesis para el que se han desarrollado una voz masculina (Pardo et al, 1995) y otra femenina (Montero et al, 2000). La conversión de texto a voz es una herramienta muy útil para realizar un prototipo rápido de la aplicación sin necesidad de grabar las frases por un locutor. Además, permite modificar dichas frases sin más que modificar el texto que se le pasa como parámetro al conversor. Una característica importante en este sistema de síntesis es la posibilidad de introducir en el texto, etiquetas o comandos específicos para cambiar las características de locución: velocidad y pith.

Un conjunto de instrucciones disponibles en el entorno TADE, muy útiles para el desarrollo de Servidores Vocales Interactivos, son aquellas que permiten el acceso a Bases de Datos. En particular el protocolo de acceso utilizado en el entorno es ODBC, proporcionado por Microsoft. Gracias a estas funciones se puede almacenar de forma organizada la información que se va a ofrecer/recoger en el servicio desarrollado. Además, estas funciones, utilizadas en combinación con aquellas que realizan la gestión de arrays y matrices de valores, permiten almacenar en una base de datos la información de depuración del diálogo. Esta información nos permitirá más adelante comparar las diferentes posibilidades consideradas en la definición del diálogo de la aplicación. Ejemplos de esta información pueden ser el número de interacciones necesarias para recoger un dato, el número de preguntas para realizar una consulta, etc. El hecho de almacenar esta información en una base de datos permite utilizar las herramientas disponibles (consultas en SQL, enlaces con otros formatos, etc.) para el análisis y obtención de estadísticas.

Por otro lado, a través de dos instrucciones del lenguaje, se permite que un operador humano pueda modificar el valor de una variable de la aplicación mientras se está ejecutando. El sistema presenta al operador una pantalla con el nombre de la variable y un cuadro donde escribir el nuevo valor. En esta pantalla, también se presenta una lista de posibles valores a asignar a la variable para que el operador no tenga que taquigrafiar el valor completo, y un botón donde se permite al operador escuchar un fichero de voz.

Fichero que puede contener una palabra o expresión pronunciada por el usuario. Esta utilidad permite que una persona pueda modificar el flujo del diálogo o puede cambiar el resultado de un reconocimiento sin que el usuario se dé cuenta de tal hecho. De esta forma, el usuario piensa que está interactuando con un sistema completamente automático pero con una calidad del servicio muy elevada puesto que prácticamente no hay error en el reconocimiento. Esta herramienta nos permitirá implementar la estrategia de Mago de Oz (WOZ: Wizard of Oz) que como veremos en el capítulo seis, es una etapa importante en el diseño del gestor de diálogo en un SVI.

Otra utilidad importante es el envío de correo electrónico. Mediante esta instrucción, el sistema puede enviar cierta información al usuario, como un resumen de la reserva de viaje realizada, o puede informar de su funcionamiento al administrador de la aplicación: posibles errores, actualización de datos o ciertas estadísticas sobre su funcionamiento.

Por último, comentar que en los últimos proyectos (Martínez, 2000; Heras, 2002) se han incorporado nuevas primitivas para la gestión de elementos multimedia como la presentación de imágenes, la proyección de vídeos o la incorporación de agentes animados disponibles en el SDK (Speech Development Kit) de Microsoft. Esta nueva funcionalidad ha dado una nueva dimensión al entorno, ofreciendo la posibilidad de desarrollar no sólo aplicaciones telefónicas sino también aplicaciones de entretenimiento o educación.

Para el desarrollo de los SVIs utilizados en esta tesis, se utilizará la última versión del sistema TADE desarrollada por Javier López y Juan Manuel Montero Martínez, alumno y tutor respectivamente del proyecto fin de carrera (López, 2001). En esta nueva versión se ha realizado una reestructuración muy importante de los módulos, orientada hacia una arquitectura distribuida similar a la arquitectura GALAXY II que fue presentada en el apartado A.1.1.

A.2.3 Servicios comerciales desarrollados con el entorno TADE

Desde 1992 el entorno TADE ha sido la plataforma sobre la que se han desarrollado la mayoría de las aplicaciones telefónicas o Servidores Vocales Interactivos diseñados en el GTH. Como ejemplos de SVIs comerciales desarrollados en este entorno cabe destacar los siguientes (todos ellos con anterioridad al desarrollo de la presente tesis):

- Sistema de atención al cliente de Hewlett Packard. Este sistema identifica al usuario a través del reconocimiento de su código de cliente (por reconocimiento de voz o DTMF, Detección de Tonos Multi-Frecuencia), y redirige la llamada hacia alguno de los ingenieros que esté libre en esos momentos o hacia el ingeniero especializado en el trato con ese cliente.

- Servicio de notas por teléfono del Dpto. Ingeniería Electrónica. Este servicio ofrece a los alumnos la posibilidad de consultar sus calificaciones por teléfono, tanto provisionales como definitivas, sin más que identificarse con su DNI. El

sistema accederá a una base de datos e informará de las calificaciones disponibles para todas las asignaturas en las que estén matriculados.

- El servidor de calificaciones del Rectorado de la Universidad Politécnica de Madrid. Este servicio es similar al del Dpto. Ingeniería Electrónica, que ofrece las calificaciones definitivas de todas las carreras de la Universidad Politécnica de Madrid. En este caso, nuestro sistema debe interactuar con un HOST IBM mediante un procedimiento de llamada remoto (RPC: Remote Procedure Call).

- El servicio de buzón vocal instalado en nuestro departamento, que permite atender las llamadas de cualquier despacho en los casos de teléfono comunicando o cuando no contesta. El sistema ofrece la opción de grabación de mensajes, enviando un e-mail de aviso al destinatario de la llamada.

A.3 Servicio de páginas blancas

El servicio de páginas blancas utilizado, ha sido el desarrollado en el proyecto IDAS (Interactive Directory Assistance Service) (San-Segundo et al, 1999; Lehtinen et al, 2000; Córdoba et al, 2000; Córdoba et al, 2001). El objetivo del proyecto fue desarrollar un demostrador capaz de dar un servicio de páginas blancas por teléfono, proporcionando números de teléfono o fax, tanto de particulares como de empresas. Un número de teléfono particular queda determinado, para el caso de nuestro demostrador, al conocer la ciudad donde vive la persona, su nombre y su primer apellido. Por otro lado, un número de teléfono de una empresa se puede obtener conociendo la ciudad donde está localizada y su nombre. Algunas características del servicio son:

- Base de datos simulada con un millón de registros de datos.

- Cuatro vocabularios de reconocimiento diferentes: ciudades, nombres de pila, apellidos y nombres de empresas. Los vocabularios de ciudades, nombres de pila y empresas tienen 1.000 palabras y los de apellidos 10.000 palabras.

- El sistema ofrece números de teléfono y fax tanto de personas particulares como de empresas u organismos oficiales.

- Permite la intervención de un operador humano cuando el reconocimiento falla.

La aplicación desarrollada es la que se describe en los siguientes pasos:

1) En primer lugar, se descuelga, se da un mensaje de bienvenida al usuario y se le pregunta al usuario si desea un teléfono particular o de empresa. Una vez realizada la selección se procede al reconocimiento de la ciudad de la que desea conocer el teléfono, pidiendo confirmación al usuario del primer candidato (y si fuese necesario, del segundo) resultado del reconocimiento. En el caso de que el usuario no confirme lo reconocido, se le pide que deletree el nombre y se le solicita confirmación del resultado. En caso de que vuelva a fallar, se anota esta circunstancia para la intervención posterior de un operador.

2) Si la opción elegida es la de empresa, se le pide al usuario el nombre de la empresa, realizando un reconocimiento análogo al visto para el caso de la ciudad (pidiendo confirmación al usuario de los dos primeros candidatos y activando el reconocimiento de nombres deletreados si fuese necesario).

3) En el caso de teléfono particular, se procede al reconocimiento del apellido y nombre, en turnos de diálogo independientes.

4) Si ha habido algún reconocimiento no confirmado por parte del usuario se le ofrece al operador un cuadro de diálogo donde se le permite escuchar lo dicho por el usuario y rellenar un cuadro de texto con el dato correcto. En ningún momento hay comunicación directa con el cliente de modo que la intervención del operador humano es transparente para el usuario. Una vez obtenidos los datos correctos, se accede a la base de datos y se proporciona el teléfono solicitado.

Con la posibilidad de intervención de un operador se garantiza que la totalidad de las consultas serán atendidas correctamente. Lo que variará será la tasa de intervención del operador, dependiendo de la calidad del reconocimiento.

En la evaluación final, publicada en (Córdoba et al, 2001), 58 personas (39 hombres y 19 mujeres) accedieron a 20 números de teléfonos: 10 de personas particulares y 10 de empresas completando 1160 consultas. La tasa de llamadas completadas automáticamente (sin intervención del operador) fue del 58,8%. El tiempo medio de llamada fue de 65 segundos para el caso de teléfono de empresa y de 84 segundos para un teléfono particular (74,5 segundos en media). Las medidas de satisfacción del usuario se reflejan en la tabla A-2.

Pregunta realizada el usuario	Valor medio
El sistema entiende lo que le dices	3,0
Las respuestas del sistema son claras y precisas	3,5
Entiendo lo que dice el sistema	3,3
Se accede rápidamente a la información	3,7
Es fácil de usar y aprender	4,3
El sistema me ayuda durante la interacción	3,7
En caso de error la corrección fue fácil	2,5
Prefiero llamar al sistema que consultar las páginas blancas	3,0
Valor medio obtenido	**3,4**

Tabla A-2: Representación de los valores de subjetividad obtenidos. La satisfacción del usuario se codifica con un valor entre 1 (completamente en desacuerdo) y 5 (completamente de acuerdo).

De esta evaluación se puede deducir que el sistema es fácil de usar, y en general, es rápido para conseguir la información deseada.

La principal aportación de la presente tesis en este sistema, ha sido el desarrollo del sistema de reconocimiento de nombres deletreados por línea telefónica y su incorporación en el sistema final. Este reconocedor se utiliza como última alternativa antes de solicitar la intervención del operador. Como veremos en el capítulo correspondiente, en más del 33% de las llamadas servidas automáticamente fue necesaria la intervención del sistema de reconocimiento de nombres deletreados, lo que pone de manifiesto su utilidad en la automatización del servicio.

A.4 Servicio de información y reserva de billetes de tren

Otro Servidor Vocal Interactivo desarrollado con el entorno TADE ha sido un servicio de información de horarios y precios de viajes de tren con posibilidad de hacer la reserva del viaje (San-Segundo et al, 2001c; San-Segundo et al, 2001e; San-Segundo et al, 2001f). Algunas características del servicio son:

- Utilización de información real obtenida de la empresa RENFE (Área de negocio de Estaciones Comerciales), dando cobertura a más del 80% de las grandes ciudades españolas con estación de tren, entre las que se encuentran los grandes núcleos urbanos.

- Utilización de viajes reales con varios transbordos haciendo uso del nuevo Motor de Búsqueda desarrollado para la empresa RENFE (Ledesma, 2000).

- Servicio de reserva simulada.

- Servicio completamente automático con posibilidad de incorporar la redirección de la llamada a un operador humano en caso de dificultades.

La aplicación desarrollada es la que se describe en los siguientes pasos:

1) Después de un mensaje de bienvenida, se le pide al usuario las ciudades origen y destino de su viaje en turnos de diálogo diferentes. Si ocurriera un fallo de reconocimiento, se solicita al usuario que deletree el nombre de la ciudad. En el caso de un nuevo fallo, se volvería a solicitar el nombre de la ciudad. Este proceso se repetiría hasta que el sistema reconozca la ciudad correctamente o hasta que el usuario cuelgue. En este servicio, no se considera la intervención de un operador humano. En el reconocedor de nombres deletreados hemos incorporado un módulo para la detección de nombres fuera del vocabulario de reconocimiento. En este caso el sistema informa al usuario y le da la posibilidad de hacer una nueva consulta.

2) Posteriormente se le pregunta al usuario sobre la fecha del viaje. Al disponer únicamente de reconocimiento de habla aislada o expresiones, la obtención de una

fecha requiere de varias interacciones. El diseño de estas interacciones se puede consultar en el capítulo 6.

3) Obtenida la fecha se pregunta al usuario el periodo del día en el que se desea viajar: por la mañana, por la tarde o por la noche. Pasando a ofrecer la información de las diferentes alternativas de viaje agrupadas en grupos de 2 ó 3 opciones El sistema va ofreciendo al usuario la posibilidad de elegir alguna de estas alternativas. Una vez elegida una opción de viaje se le da al usuario la posibilidad de seleccionar un viaje para la vuelta. En caso afirmativo el proceso sería análogo: petición de fecha y periodo del día, y selección de la alternativa elegida.

4) Una vez obtenido un viaje de ida o de ida y vuelta se le informa de los precios del viaje para las diferentes clases del tren y se le da la posibilidad de hacer la reserva. En caso afirmativo se le pregunta al usuario el número de plazas a reservar y la clase, pasando a emitir un código de reserva que se debe presentar en la estación para retirar los billetes.

Aunque en el capítulo seis se presentarán los resultados finales de la evaluación, en este apéndice pasamos a comentar brevemente los resultados sobre satisfacción del usuario obtenidos. En esta evaluación, 105 personas (55 estudiantes de la universidad y 50 empleados de la empresa RENFE) llamaron al sistema para completar 4 escenarios posibles de viaje obteniendo un total de 375 consultas (no todos los usuarios completaron los 4 escenarios). El tiempo medio por consulta fue de 195 segundos con un número medio de preguntas de 18,8. Las medidas de satisfacción del usuario obtenidas se reflejan en la tabla A-3.

Pregunta realizada el usuario	Valor medio
El sistema comprende lo que le dices	3,1
Las respuestas del sistema con claras y concisas	3,4
Entiendo lo que el sistema me dice	3,5
Se accede a la información de trenes rápidamente	2,9
El sistema es fácil de usar y de aprender	3,6
El sistema me ayuda durante la interacción	3,1
En caso de error la corrección fue fácil	2,9
El sistema me pregunta en un orden lógico	3,6
Valor medio obtenido	**3,0**

Tabla A-3: representación de los valores de subjetividad obtenidos. La satisfacción del usuario se codifica con un valor entre 1 (completamente en desacuerdo) y 5 (completamente de acuerdo).

Como se puede observar, a pesar de utilizar voz sintética, las preguntas referidas a la inteligibilidad del sistema obtienen muy buena puntuación lo que pone de manifiesto la

calidad de la voz femenina utilizada (Montero et al, 2000). Además, la estructura del diálogo también obtiene muy buena puntuación 3,6, gracias al extenso estudio realizado en esta tesis para diseñar el gestor de diálogo. Por otro lado, los mecanismos de corrección y la velocidad del sistema para obtener la información, son los puntos con la peor puntuación obtenida.

La aportación de esta tesis a este servicio ha sido fundamental y ha consistido en el diseño e implementación del gestor de diálogo. En el desarrollo de este gestor se ha definido y propuesto una metodología de diseño consistente en 5 fases: análisis de la base de datos, diseño por intuición, diseño por observación, simulación del sistema y mejora iterativa. Esta metodología tiene como finalidad generar un diálogo lo más adaptado posible a las necesidades o preferencias de los usuarios, teniendo en cuenta las limitaciones de la tecnología del habla utilizada. Por otro lado, se han incorporado medidas de confianza en los sistemas de reconocimiento disponibles en el entorno TADE, con el fin de permitir una mejor gestión de las confirmaciones de los datos. Los detalles sobre su implementación se recogen en el capítulo 6.