

Capítulo 6. Análisis y diseño del sistema

En este capítulo explicaremos de manera detallada como implementamos el reconocedor de voz al sistema MexVox, así como los problemas a los que nos enfrentamos y como los resolvimos y el diseño final del sistema en base al análisis previo usando herramientas UML para este trabajo.

6.1 Descripción general del sistema

Al principio se contaba con el código fuente y ejecutable del DosVox en portugués, pero no se conocía el funcionamiento del sistema por lo que fue necesario empezar con un estudio del DosVox.

Primero leímos documentos acerca del DosVox y de la importancia que tiene para los invidentes, esto para documentarnos acerca del sistema. Al mismo tiempo que hacíamos este trabajo, empezamos a probar el DosVox en portugués y hacer un documento de especificaciones del mismo.

Una vez concluido este trabajo, empezamos a trabajar con el código fuente del DosVox, el primer obstáculo con el que nos enfrentamos fue que el código fuente estaba escrito en lenguaje delphi, del cual no teníamos conocimiento, por lo que fue necesario aprender este lenguaje.

Un problema al que nos enfrentamos fue que los códigos del DosVox son códigos escritos por varios programadores a lo largo de 10 años, por lo que los estilos de programación son diversos, y los primeros códigos utilizan programación con “goto” por lo que fue muy difícil entender el funcionamiento de los códigos. Para resolver este problema y como ayuda al servicio social, el autor del DosVox, el Dr. Antonio Borges, nos hizo una visita a la Universidad de las Américas. Durante esta visita, el equipo del servicio social (entre ellos los autores de esta tesis) asesorados por el Dr. Borges realizamos lo que sería la primera versión en español del DosVox. Gracias a esto pudimos entender el funcionamiento de los códigos del DosVox, obtener el MexVox para a partir de este empezar a trabajar con el reconocedor.

A la vez que estudiamos el DosVox, empezamos a investigar que reconocedor utilizaríamos para nuestra aplicación.

Para añadirle reconocimiento de voz al MexVox, necesitábamos un reconocedor que trabajara sobre la plataforma de Windows y que fuera completamente gratuito. Entre los reconocedores que encontramos y consideramos que fuera posible implementarlos en el MexVox fueron los siguientes:

- CSLU TOOLKIT: esta herramienta permite hacer reconocedores de voz en español y es completamente gratuito, se puede adquirir a través de la web. Sin embargo había que construir todo el reconocedor, desde crear el diccionario de datos, etiquetar, alinear y crear la aplicación y dentro de nuestros objetivos no se encuentra

el construir un reconocedor de voz, si no implementar uno que ya se encuentre hecho.

- Sphinx: Al igual que el toolkit esta herramienta permite crear un reconocedor de voz. Esta herramienta al igual que el Toolkit es gratuita, pero las aplicaciones que pueden ser creadas son para el idioma inglés.

Existe en la Universidad Autónoma de Tlaxcala un profesor, el Mc. Heriberto Cuayáhuatl que está trabajando con la creación de reconocedores de propósito específico para palabras en español utilizando el Sphinx, por lo que lo contactamos y nos dijo que el construiría un reconocedor con las palabras de nuestro dominio en español para que pudiéramos trabajar con éste, pero por cuestiones de tiempo en su trabajo no le fue posible construir el reconocedor, por lo que deberíamos buscar otro.

- Speech Microsoft Recognition: éste es un reconocedor gratuito ya construido, está desarrollado para el idioma inglés. Una de las ventajas que encontramos sobre este reconocedor es que trabaja bajo SAPI que es el mismo API utilizado por el sintetizador de voz para el MexVox. Experimentando soluciones factibles dado el problema de que el reconocedor esta desarrollado para el idioma ingles, encontramos una solución interesante. Ésta es entrenar el reconocedor leyendo las palabras (que trae de origen para entrenar y que están escritas en ingles), pero pronunciándolas en español. Los resultados obtenidos se muestran en el capítulo 7.

Al darnos cuenta de los buenos resultados que arrojaron las pruebas, optamos por utilizar el Speech Microsoft Recognition, pero nos encontramos con otro problema, que los componentes Active_X que se utilizan para programar en Delphi, no son soportados por la aplicación Mexvox, ya que dichos componentes son para aplicaciones gráficas (frame, panel) y MexVox trabaja sobre una consola.

Para darle solución a este problema decidimos usar una arquitectura cliente-servidor para la implementación del reconocedor de voz al sistema MexVox. Dentro de este esquema, el sistema MexVox hace las veces de Cliente y el Reconocedor es el servidor.

Dentro del apéndice A, podemos ver que en el sistema MexVox existen varios programas, entre ellos están el programa principal MexVox, el editor de textos, el lector de documentos y el programa de salida del sistema. Estos programas son los que pueden hacer uso del reconocedor. Estos se conectan con el reconocedor de voz mediante una conexión TCP.

Al entrar al MexVox; éste llama al programa servidor (servidorvoz). Servidorvoz es un programa en Delphi en el que se implementó el reconocedor; éste programa se ejecuta y espera a que algún cliente se conecte. Después de esto, el programa del MexVox hace la conexión TCP, es decir, llama al programa servidor y establece el puente de comunicación entre el MexVox y el servidor. Para implementar el programa Servidorvoz y adaptar los

programas del MexVox a la arquitectura Cliente/Servidor fue necesario instalar en Delphi los componentes TClientSocket y TServerSocket (Ver Apéndice F).

Una vez que la conexión con el servidor fue hecha, el sistema pregunta al usuario que le informe si es hombre o mujer (que presione “h” para hombre, “m” para mujer), esto porque se hicieron dos entrenamientos, uno con voces de hombres y otro de mujeres (3 voces para cada entrenamiento), así el sistema sabrá con cual trabajar para hacer que las probabilidades de reconocimiento sean mayores. El usuario tendrá la opción de elegir si quiere o no utilizar el reconocedor. Si desea utilizar el reconocedor, tendrá que activarlo para hacer la petición al servidor. Para activarlo simplemente tendrá que presionar la tecla F11 y en ese momento el reconocedor estará listo para reconocer los comandos que el usuario diga y enviarlos al MexVox. Es importante mencionar que mientras el reconocedor esté activo, los comandos solo podrán ser ejecutados por medio de la voz y el teclado quedará inactivo y será activado nuevamente al presionar F11, con lo cual desactivará al reconocedor (no se cierra la conexión, solo no hay comunicación con el reconocedor, hasta que sea activado nuevamente con F11).

Cuando el MexVox llame al reconocedor, éste activará el menú correspondiente al MexVox, es decir, solamente se podrán reconocer los comandos del MexVox. Esto es una ventaja, ya que si por error el usuario dice una palabra de otro programa, del editor por ejemplo, ésta será ignorada y no afectará el funcionamiento del sistema.

Cada vez que se invoque alguno de los programas MexVox, editor, lector o salida de MexVox, la conexión del programa anterior con el reconocedor, se cerrará y se abrirá la

del nuevo programa con el reconocedor; se activará el menú correspondiente al nuevo programa (Ver Apéndice G) y funcionará como se explicó anteriormente. Por ejemplo, si estamos en el MexVox e invocamos al editor, la conexión del MexVox con el reconocedor se cerrará y se abrirá la conexión del editor con el reconocedor, ya no se podrán reconocer palabras del menú del MexVox, solamente del menú del editor y de la misma manera como se mencionó antes, con F11 se invoca y se duerme al reconocedor y. Cuando se sale del editor se cierra la conexión con el servidor.

Con el programa de salida del MexVox (Fechawin), sucede lo mismo, solamente que aquí al salir del programa, éste no solo cierra la conexión con el servidor sino que cierra la ejecución del programa servidorvoz, dando así fin al sistema.

6.2 Análisis del Sistema

6.2.1 Casos de uso

Id: 01

Caso de uso: MexVox por medio del habla de comandos

Autores: José Manuel Zulaica Rugarcía y Bertha Mariel Márquez Avendaño

Actores: Invidentes (usuarios).

Propósito: Dictar comandos de voz al MexVox y que ejecute las acciones correspondientes.

Resumen: El usuario entra al sistema MexVox dice palabras que el sistema reconoce y ejecuta las acciones correspondientes a esas palabras. Al terminar las tareas el usuario sale del sistema.

Tipo: Primario-esencial

Acción del actor	Respuesta del sistema
<p>1. El caso de uso comienza cuando el usuario invidente entra al sistema MexVox y se coloca el micrófono con la intención de realizar cierta tarea.</p> <p>3. El usuario presiona la tecla “F11” y dice por el micrófono alguna palabra.</p> <p>5. El usuario escucha al sistema y confirma que la palabra reconocida es correcta.</p> <p>6. Presiona la tecla “F11” para desactivar el reconocedor.</p> <p>8. El usuario realiza la tarea</p> <p>9. El usuario sale del sistema con una tarea realizada</p> <p>Cursos alternos: Línea 5: El usuario indica que no es la palabra. Vuelve a la línea 3 Línea 7: Ejecuta la acción correspondiente a la palabra reconocida, es decir, entra al módulo elegido (Probar teclado, lector de textos, imprimir, archivos, discos o juegos</p> <p>Línea 9: El usuario regresa a la línea 3</p>	<p>2. Recibe al usuario con un saludo de bienvenida a MexVox. Pregunta al usuario que tarea desea realizar.</p> <p>4. Reconoce que la palabra hablada por el usuario es un comando válido y la “dice” para confirmar.</p> <p>7. Ejecuta la acción correspondiente a la palabra reconocida, es decir, entra al caso de uso 02.</p>

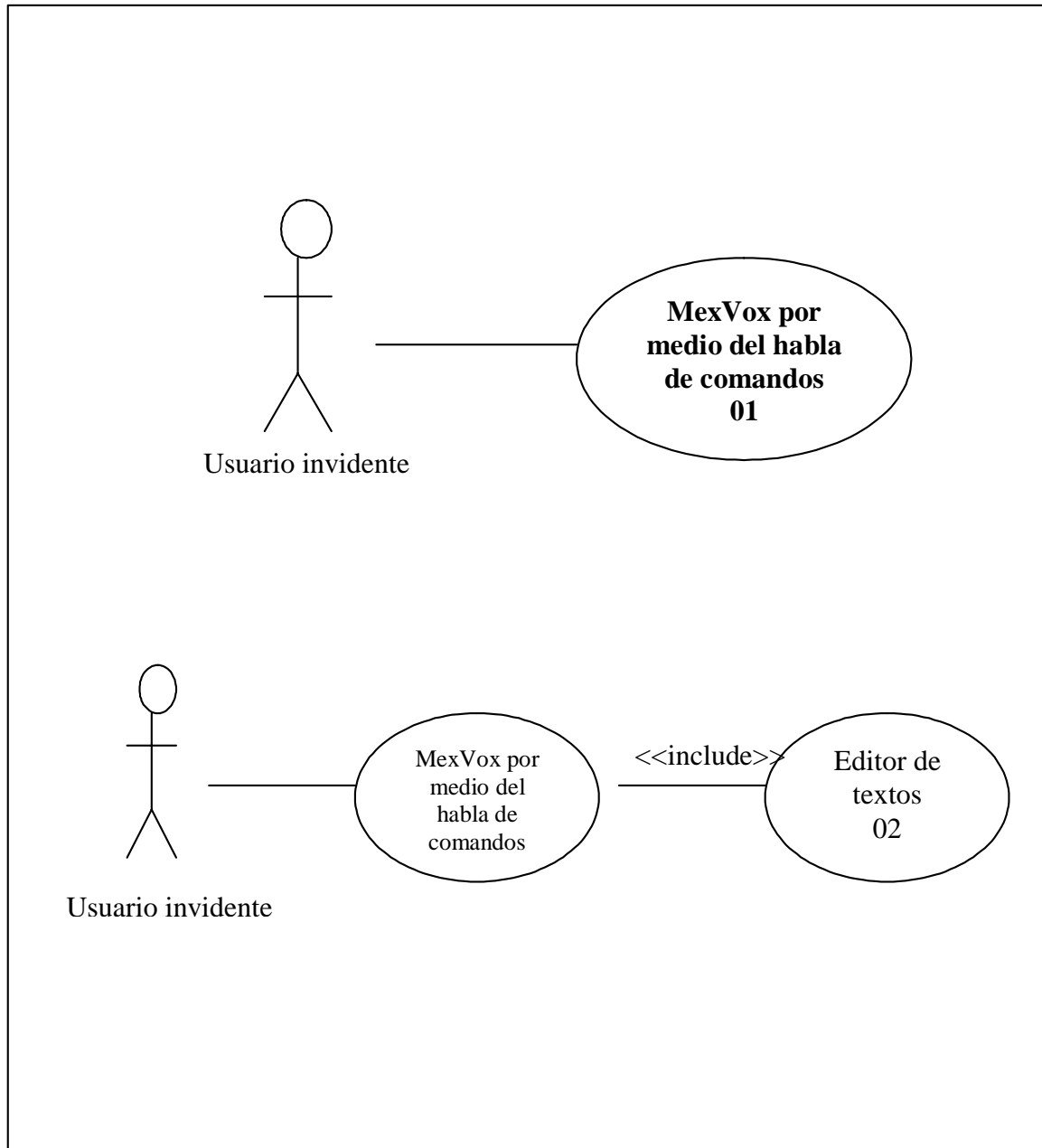
Id: 02
 Caso de uso: Editor de textos
 Autores: José Manuel Zulaica Rugarcía y Berta Mariel Márquez Avendaño
 Actores: Invidentes (usuarios)
 Propósito: Accesar las funciones del editor de textos mediante comandos de voz

Resumen: El usuario entra al editor de textos, crea textos, les da formato y los guarda. Hace estas funciones por medio del habla de comandos que el sistema reconoce

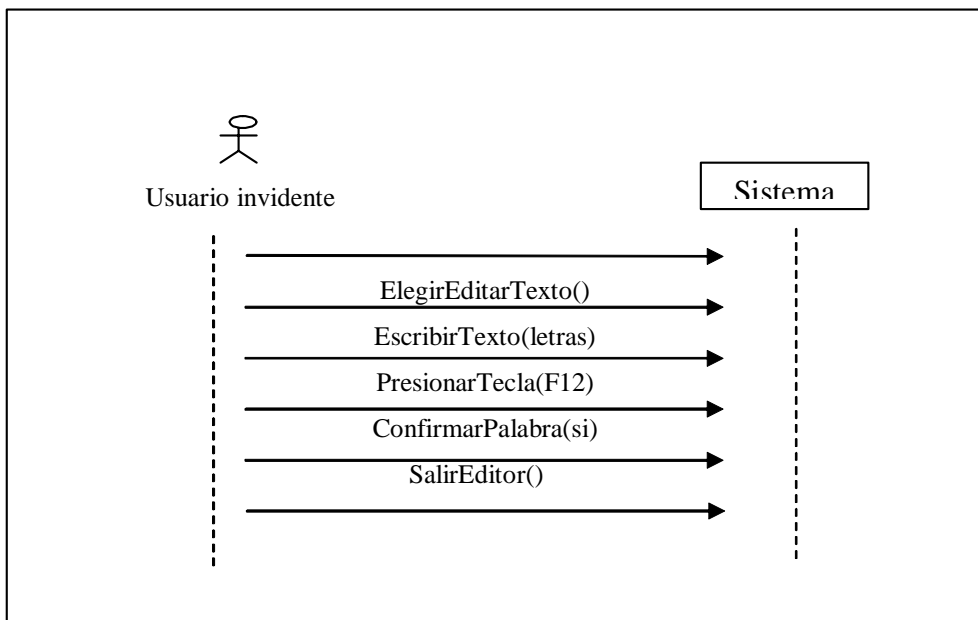
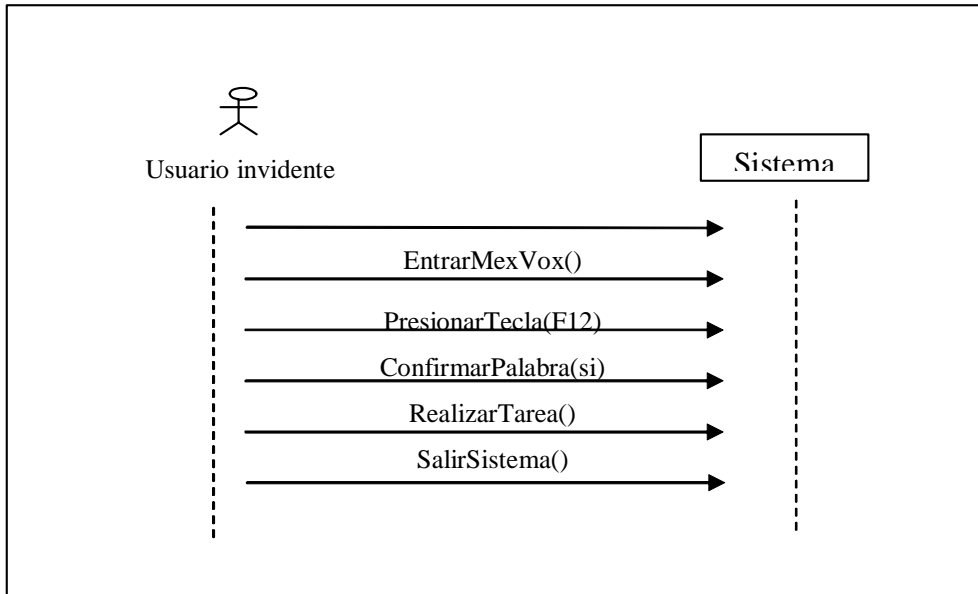
Tipo: Primario-esencial

Acción del actor	Respuesta del sistema
1. Comienza cuando el usuario entra al editor de textos .	2. Recibe al usuario con un saludo de bienvenida al editor de textos. Pregunta al usuario el nombre del archivo.
4. El usuario escribe un texto	3. Crea un archivo con el nombre que el usuario le dio.
5. El usuario puede elegir herramientas del editor presionando la tecla “F11” y hablando por el micrófono alguna palabra.	6. Reconoce la palabra hablada por el usuario y la “dice” (sintetiza) para confirmar.
7. El usuario escucha al sistema y confirma que la palabra reconocida es correcta.	
8. Presiona la tecla “F11” para desactivar el reconocedor.	9. Ejecuta la opción elegida para el documento
10. El usuario elige salir del editor.	
11. Regresa al caso de uso 01.	
Cursos alternos: Línea 3: Abre el archivo guardado con el nombre que el usuario le dió	
Línea 7: El usuario indica que no es la palabra. Vuelve a la línea 5.	
Línea10: El usuario realiza la tarea y regresa a la línea 5	

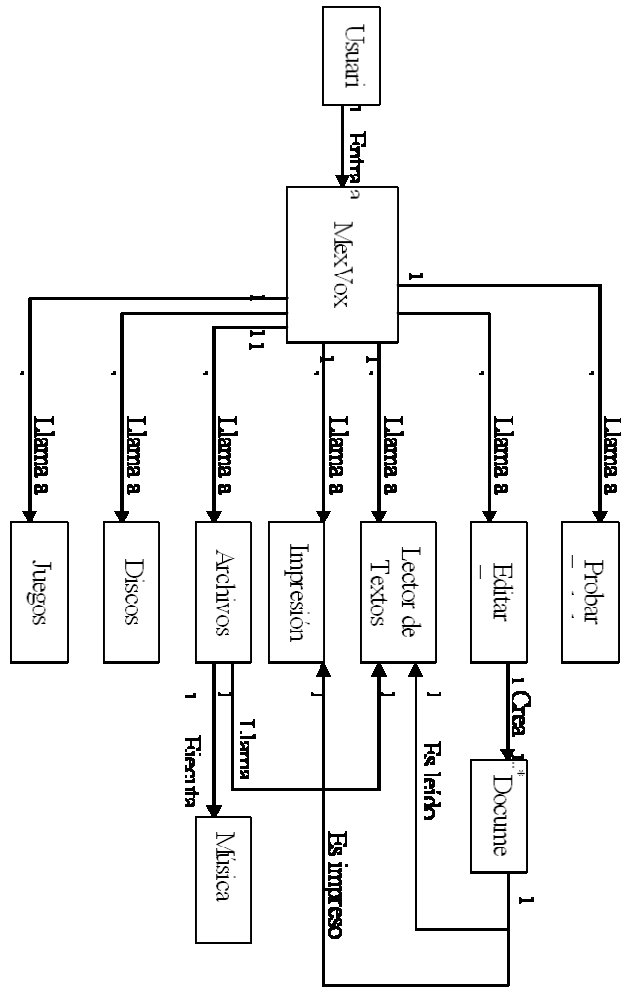
6.2.2 Diagrama de Casos de uso



6.2.3 Diagramas de Secuencia



6.2.4 Diagramas de Secuencia



6.3 Diseño del sistema

6.3.1 Clases

Mex Vox
Comando : String Activado : boolean
ConexionTcp() PeticiónServidor() Lector() Editar() ProbarTeclado() Impresión() Archivos() Discos() Juegos() Salir()

Lector
comando : String activado : boolean
ConexionTcp() PeticiónServidor() Salir()

Editar
comando : String activado : boolean
ConexionTcp() PeticiónServidor() DecirPalabra() Guardar() LineaActual() Deletrear() BuscarIntervalo() OrdenarArchivo() RemoverLinea() Hora() Ayuda() Salir()

ServidorVoz
socket : TserverSocket comandos : TVcommand comando : String
ClienteSocketConnect() ServerSocketClientConnect() ServerSocketClientDisconnect() ServerSocketClientRead() VcommandRecognize()

6.3.2 Diagrama de Clases

