

Capítulo VI. GetME: Módulo de Software de Lectura de Mensajes de Texto para un Ambiente de Aprendizaje.

6.1. Arquitectura General de GetME.

La inclusión de tecnología de voz en teléfonos celulares y/o agendas personales, permitirá al estudiante, una interacción mucho más amigable y sencilla, cuando esté interactuando en un ambiente de aprendizaje. El módulo de software GetME, desarrollado por el presente tesista, tiene como propósito lograr la lectura de mensajes de texto para el estudiante en ambiente de aprendizaje inalámbrico, utilizando como base el lenguaje VoiceXML.

En la figura 6.1, se muestra el modelo arquitectónico, de cómo funciona el modulo de software GetME, así como una breve explicación del seguimiento que realiza.

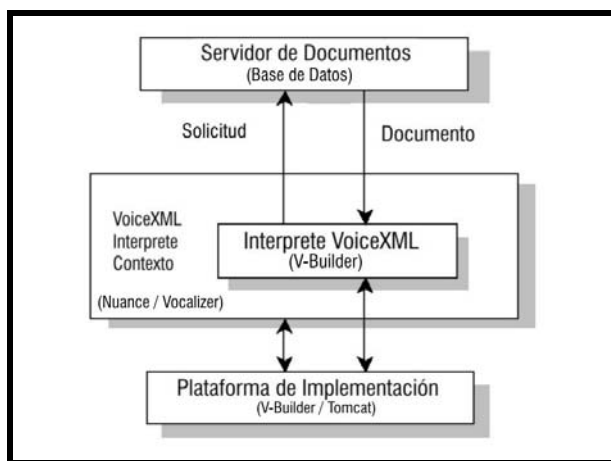


Figura 6.1. Modelo Arquitectónico de GetME.

A continuación, se explican los pasos de cómo funciona el módulo GetME:

- El Servidor de Documentos (Base de Datos), procesará las solicitudes provenientes de la interfaz del estudiante.
- El servidor producirá documentos VoiceXML en respuesta, los cuales son procesados por el intérprete VoiceXML (V-Builder).

- El intérprete de contexto VoiceXML (Nuance / Vocalizer), monitoreara las solicitudes del estudiante, de manera paralela con el intérprete VoiceXML (V-Builder).
- La plataforma de implementación (V-Builder / Tomcat), es el interprete de aplicación en el que se desarrollo el módulo GetME, la cual es controlada por el intérprete de contexto VoiceXML (Nuance / Vocalizer) y por el intérprete VoiceXML (V-Builder), permitiendo así, tener un control de las solicitudes por parte del estudiante.
- La plataforma de implementación (V-Builder / Tomcat), se encarga de generar los eventos de acuerdo a las interacciones del estudiante, en el ambiente de aprendizaje. Como pueden ser órdenes habladas, desconexión, salida del ambiente; así como eventos propios del sistema (expiración del tiempo de conexión). Algunos de estos eventos, son actos que por si solo, los realiza el intérprete VoiceXML (V-Builder), y que el estudiante no es notificado.

6.2. El Interprete V-Builder 1.2.

Para el desarrollo del módulo GetME, para la lectura de mensajes de texto, se empleo como herramienta de programación y diseño de diálogos, el V-Builder 1.2., por presentar mejores opciones al momento de la concepción del módulo GetME. Lo que otros interpretes para aplicaciones en VoiceXML, no tienen disponibles o bien habilitadas.

Así como también la realización de las pruebas iniciales al módulo GetME, por ser una IDE grafico, sirvió de apoyo para la visualización de errores en el código de VoiceXML.

6.2.1. Limitantes del V-Builder 1.2.

Como todo software gratuito para el desarrollo de aplicaciones en VoiceXML. V-Builder necesariamente realiza una o varias conexiones a su servidor principal. Originando que cada código que se elabore en VoiceXML, se tenga que hacer referencia a la ruta de acceso del servidor de Nuance. Lo que origina en la mayoría de los casos, una restricción

si se desea hacer un acceso a un archivo de manera local, con herramientas como Tomcat de Jakarta, debido a que solicita se encuentre en el servidor de Nuance, por lo que se deben buscar otras opciones para el acceso a un archivo y evitar así algún mensaje de error.

6.3. Análisis y Diseño del Módulo GetME.

De acuerdo a la metodología sugerida por [VoiceXML, 2002], para el diseño de aplicaciones en VoiceXML, se utilizó la herramienta de diseño diagramas de flujo, por aportar al desarrollador de VoiceXML un alto nivel en el diseño, así como un fácil uso y entendimiento en las aplicaciones en dicho lenguaje, permitiendo así un reconocimiento más objetivo sobre la usabilidad del módulo GetME, dando al desarrollador una retroalimentación adecuada, a lo largo de todo el ciclo de vida del módulo GetME. A continuación en la figura 6.3, se ilustra el diagrama de flujo del módulo GetME para lectura de mensajes de texto, y cómo está constituido actualmente.

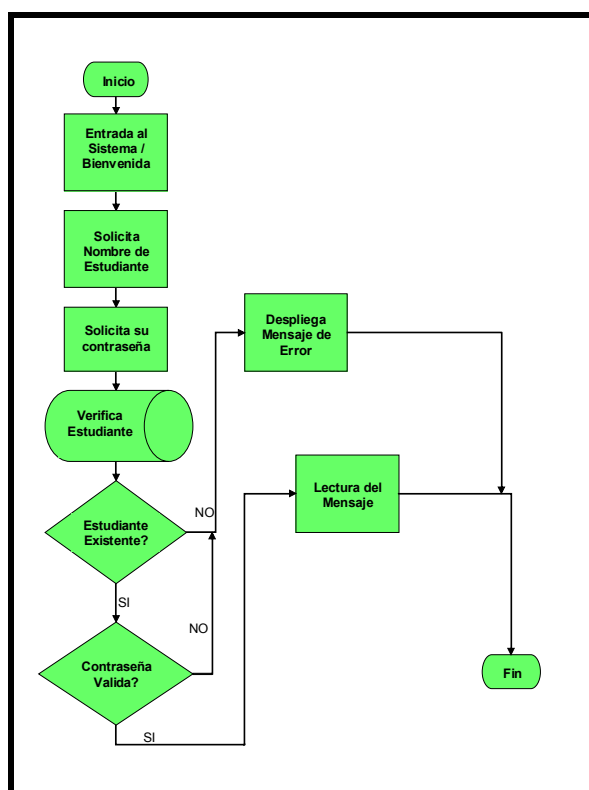


Figura 6.3. Diagrama de Flujo del módulo GetME para la lectura de mensajes de texto en el interprete V-Builder.

6.3.1. Diseño del Diálogo del Módulo GetME.

En el Ambiente de Integración de Desarrollo V-Builder, el dialogo principal de GetME, ilustra de manera clara como es el diseño del módulo GetME, con la opción para el desarrollador de la visualización misma del código fuente en VoiceXML; a continuación se ilustra en la figura 6.3.1, el dialogo principal de GetME.

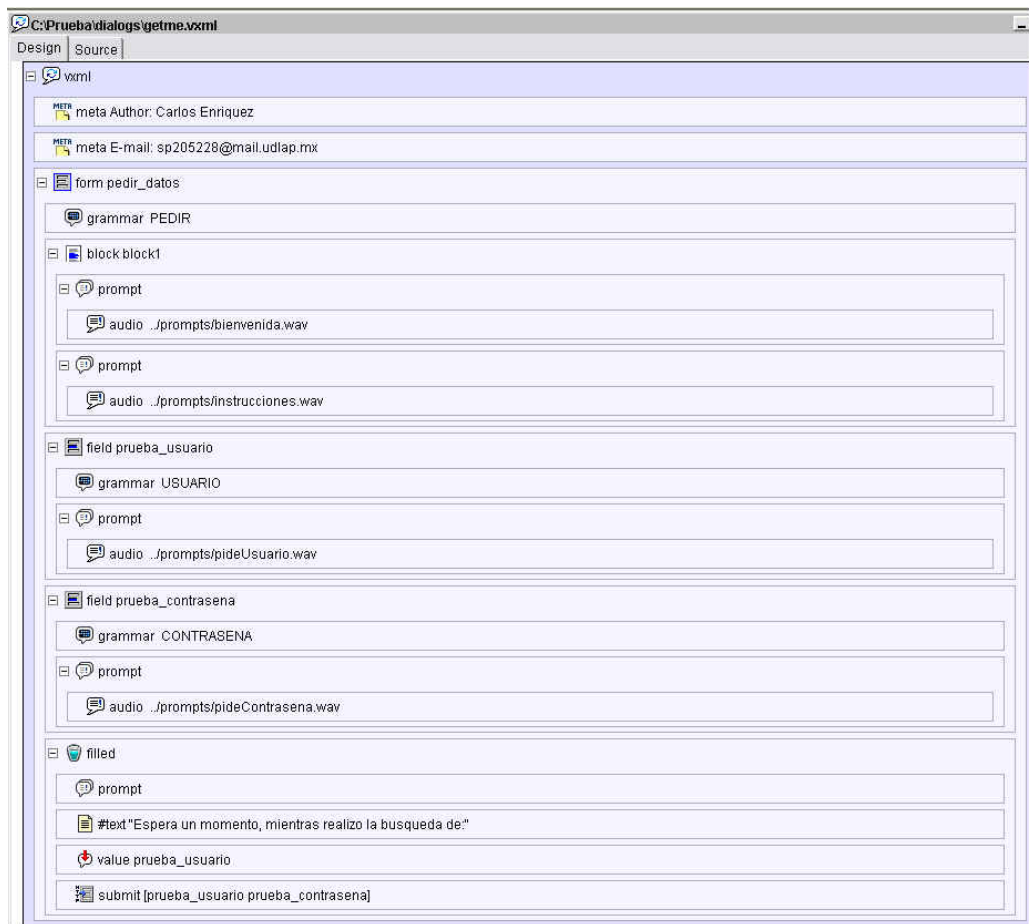


Figura 6.3.1. Diseño del dialogo utilizado en GetME en V-Builder.

6.4. Funcionamiento del módulo GetME con el interprete V-Builder.

GetME, es un módulo prototipo para la lectura de mensajes de texto, a través de Cassiel-WAP, dándole la facilidad a los estudiantes con capacidades especiales y “normales”, de escuchar los mensajes que, otros miembros de su comunidad le envíen al estudiante.

El módulo GetME, funciona de la siguiente manera:

1. El estudiante es recibido por un mensaje de bienvenida. Y se le da una breve explicación del uso de GetME.
2. GetME, solicita al estudiante su nombre de estudiante.
3. GetME, solicita al estudiante su contraseña.
4. GetME, realiza la verificación de los datos proporcionados por el estudiante.
5. En caso de que, GetME, encuentre una diferencia entre alguno de los datos emitidos por el estudiante, le será informado, para que los vuelva a capturar.
6. Si GetME, encuentra una vez más, que los datos no coinciden, se despide del estudiante y termina la aplicación.
7. Si el estudiante dio correctamente sus datos a GetME, procederá a la lectura de los mensajes almacenados en la base de datos.
8. Una vez que termina la lectura de mensajes, GetME, se despide y termina la aplicación.

A continuación en la figura 6.4, la ejecución de dialogo que muestra el módulo GetME al estudiante, de ahí en adelante, todo es a través del modulo de reconocimiento proveído por V-Builder, por lo que el estudiante no vera reflejado, a partir de este momento, ningún texto, por parte de GetME.

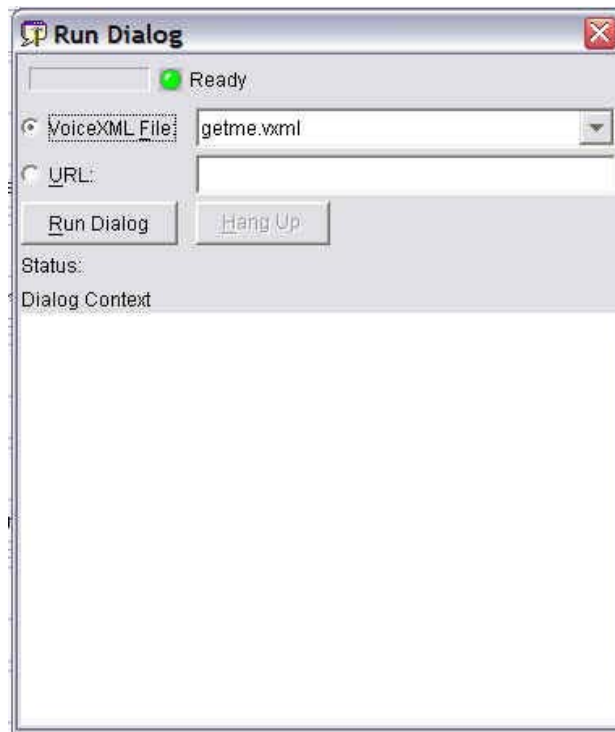


Figura 6.4. Caja de la Ejecución del Dialogo del módulo GetME, en el interprete V-Builder.

6.5. Funcionamiento del Intérprete VoiceXML (V-Builder).

Es la aplicación residente en el servidor VoiceXML, la que interpreta los documentos de VoiceXML. El Ambiente de Integración de Desarrollo V-Builder 1.2, permitirá la programación y el seguimiento mismo del módulo GetME, a través de toda la aplicación, dándole la opción al desarrollador, de un mejor control, gracias a la amigabilidad del entorno de V-Builder.

V-Builder, es un ambiente de integración, para el desarrollo de aplicaciones basadas en voz, usando VoiceXML, y el paquete de Nuance SpeechObjects. V-Builder ofrece un alto rango de funciones, en el manejo de aspectos de desarrollo de aplicaciones, basadas en voz, incluyendo administración de proyectos, edición del código, manejo de gramática, grabación y reproducción de archivos de sonidos, y un componente que servirá para las pruebas de la aplicación.

6.5.1. Funcionamiento del Intérprete de Contexto VoiceXML (Nuance / Vocalizer).

Es la aplicación residente en el Servidor VoiceXML, que realiza la depuración y ejecución de los documentos en VoiceXML. Verificando lo referente al idioma utilizado, así como su gramática misma.

6.5.2. Funcionamiento del Servidor de Documentos (Base de Datos).

Será la base de datos que puede estar en el servidor Cassiel-WAP, que se encarga de proporcionar toda la información necesaria, solicitada por el estudiante a través de la interfaz del teléfono celular.

6.5.3. Funcionamiento de la Plataforma de Implementación.

En nuestro proyecto utilizamos V-Builder 1.2, para lo referente al lenguaje VoiceXML y la gramática e idioma respectivo; y Tomcat de Jakarta, para el acceso a la base de datos.

6.6. Requerimientos de Software y Hardware para la Ejecución de GetME.

La explicación de los requerimientos utilizados para la creación del módulo GetME, fueron tratados en el capítulo 1, por lo que aquí solo se hace mención al hardware y software utilizados, durante las pruebas y concepción del mismo, para garantizar el buen funcionamiento.

6.6.1. Requerimientos de Software.

- Especificaciones de VoiceXML 2.0.
- Nuance V/Builder 1.2.
- Sintetizador de Voz en español Vocalizar 1.0.
- Jakarta Tomcat 3.3.1.
- Microsoft Access.

- IBM Websphere Voice Server SDK.
- Microsoft Windows XP Profesional.
- Teléfono Celular con capacidad WAP y Reconocimiento de Voz (Sony Ericsson T68i).
- Conexión vía teléfono USA, para pruebas.

6.6.2. Requerimientos de Hardware.

- Computadora Pentium 4 a 1.4 GHz.
- 512 MB de Memoria.
- 40 GB de Disco Duro.
- Tarjeta de Sonido Sound Blaster 5.1. Live!
- Conexión a Internet.
- Micrófono y Altavoces.

6.7. El API de Cassiel, como apoyo para el módulo GetME.

Para fines de desarrollo de GetME, se empleo como parte complementaria, parte del API del Ambiente de Aprendizaje de Cassiel, desarrollado por [Saito, 2001] y [Paredes, 2001], para entender el funcionamiento de Cassiel, a continuación se anexan los módulos del API, que sirvieron de ayuda para el módulo GetME.

mx.udlap.centia.cassiel.user
Interface RemoteMessagesManager

All Superinterfaces:

java.rmi.Remote

All Known Implementing Classes:

[MessagesManager](#), [MessagesManager_Stub](#)

```
public interface RemoteMessagesManager
extends java.rmi.Remote
```

Esta clase, es la interfaz de la entidad encargada de recuperar los mensajes de los usuarios dentro del ambiente de CASSIEL.

Version:

1.0

Author:

Nobuji A. Saito y Rosa G. Paredes.

Method Summary

| | |
|----------------------------|---|
| Message [] | getMessages (java.lang.String login) Método que permite recuperar todos los mensajes para un usuario dado, dentro del ambiente de CASSIEL. |
| Message [] | getMessages (java.lang.String login, java.lang.String type) Este método recupera los mensajes de un usuario, de acuerdo al tipo de mensaje. |
| int | getNumberOfMessages (java.lang.String login) Este método para obtener el número de mensajes de un usuario. |
| int | getNumberOfMessages (java.lang.String login, java.lang.String type) Este método para obtener el número de mensajes de un usuario, de acuerdo al tipo de mensaje. |

Method Detail

getMessages

public [Message](#)[] **getMessages**(java.lang.String login)

throws java.rmi.RemoteException

Método que permite recuperar todos los mensajes para un usuario dado, dentro del ambiente de CASSIEL.

Parameters:

login - El login del usuario para el que se van a recuperar los mensajes.

Returns:

El arreglo de mensajes que se recuperaron para ese usuario.

java.rmi.RemoteException

getMessages

public [Message](#)[] **getMessages**(java.lang.String login,
java.lang.String type)

throws java.rmi.RemoteException

Este método recupera los mensajes de un usuario, de acuerdo al tipo de mensaje.

Parameters:

login - La clave del usuario para el que recupera los mensajes.

type - El tipo de mensajes que se desea contar.

Returns:

El conjunto de mensajes del usuario, null en caso de no tener ningún mensaje de ese tipo.

java.rmi.RemoteException

getNumberOfMessages

```
public int getNumberOfMessages(java.lang.String login)
```

```
    throws java.rmi.RemoteException
```

Este método para obtener el número de mensajes de un usuario.

Parameters:

login - La clave del usuario para el que cuenta los mensajes.

Returns:

El número de mensajes para ese usuario.

java.rmi.RemoteException

getNumberOfMessages

```
public int getNumberOfMessages(java.lang.String login,
```

```
    java.lang.String type)
```

```
    throws java.rmi.RemoteException
```

Este método para obtener el número de mensajes de un usuario, de acuerdo al tipo de mensaje.

Parameters:

login - La clave del usuario para el que cuenta los mensajes.

type - El tipo de mensajes que se desea contar.

Returns:

El número de mensajes del usuario, null en caso de no tener ningún mensaje de este tipo.

java.rmi.RemoteException

6.8. Pruebas realizadas al Módulo GetME.

Dentro de las diversas pruebas que se hicieron al modulo GetME, se logro obtener una visión clara de cuales son los alcances del módulo de GetME, así como las limitaciones que contamos en cuestión de hardware.

- Las pruebas realizadas al módulo GetME, con el interprete V-Builder, presentaron dificultades durante la realización de las conexiones a los mensajes de texto,

debido a que requería de enlaces que no localizaba en el equipo residente donde se estaba ejecutando.

- Para la solución del problema de la conexión, se realizaron diversos contactos con el soporte técnico de Nuance, logrando que se nos enviarán algunos módulos de actualización, que solucionaban las fallas en los enlaces al servidor de Nuance, y los cuales fueron instalados en el directorio residente del V-Builder.
- Aunque el modulo de idioma español estaba cargado en el V-Builder y declarado en el mismo, en ciertas lecturas de los mensajes de texto, el sintetizador sonaba muy robotizado y con el idioma en inglés, lo que dificultaba en algunas ocasiones la comprensión de lo que decía el sintetizador al estudiante. Se realizaron pruebas con mensajes de texto en idioma ingles, y se obtuvieron buenos resultados, en lo referente al sintetizador.
- Se realizo un módulo para la lectura de mensajes de voz, donde se empleo como herramienta de programación en VoiceXML, el IBM Websphere Voice Server SDK, obteniéndose una buena calidad en el módulo de síntesis de voz en español, debido a que es mucho más entendible para el estudiante, que el presentado por el V-Builder 1.2. de Nuance.
- Igualmente como V-Builder, el interprete IBM Websphere Voice Server SDK, requiere de hacer una conexión a su servidor residente. En lo referente a la interfase desplegada al estudiante, se presenta muy pobre, debido a que no es grafica, lo que pudiera ocasionar para estudiantes “normales” un desinterés mientras se interactúa con el intérprete.
- El interprete IBM Websphere Voice Server SDK, es un ambiente de integración, para el desarrollo de aplicaciones basadas en voz, usando VoiceXML, Ofreciendo al desarrollador la elección de módulos de idiomas, de acuerdo al tipo de aplicación de VoiceXML, que desee desarrollar. El manejo de la gramática, se

encuentra incluido en los módulos de idioma instalados en conjunto con el intérprete.

- El funcionamiento de dicha versión beta, busco seguir los pasos básicos del módulo GetME, desarrollado en V-Builder, por lo que, los procesos de visualización de pantallas se resumen a una sola pantalla, debido a que todo el proceso es realizado por el sintetizador de voz, incluido en el intérprete.

En la figura 6.8, se ilustra la ventana de ejecución del módulo beta de GetME, para la lectura de mensajes de voz, que le muestra al estudiante, en el intérprete de IBM Websphere Voice Server SDK, en donde la interacción es a través de instrucciones vía voz, por lo que, el estudiante solo vera desplegarse texto, por parte del módulo beta de GetME, esperando las indicaciones que le diga el módulo GetME.



```
GetMEAX Software
A: Too loud (1.0)
A: Audio level (0.6)
?: Acabe (too short)
A: Audio level (0.3)
A: Too loud (1.0)
A: Too loud (1.0)
A: Audio level (0.5)
A: Too loud (0.9)
A: Audio level (0.4)
?: Acabe (too short)
A: Audio level (0.3)
A: Audio level (0.3)
A: Audio level (0.3)
A: Too loud (1.0)
A: Too loud (0.9)
?: Acabe (too short)
A: Too loud (0.8)
A: Audio level (0.4)
A: Too loud (0.8)
A: Too loud (1.0)
A: Too quiet (0.2)
A: Too quiet (0.1)
H: Acabe
A: recording stopped
```

Figura 6.8. Ventana de ejecución de la versión beta del módulo GetME, para la lectura de mensajes de voz, en el intérprete IBM Websphere Voice Server SDK.

6.9. Resultados de la interacción con el módulo GetME.

El desarrollo del módulo GetME se incluye como parte complementaria a este trabajo de investigación, para tenerla reflejada en un módulo de software que sirviera a la UDLA,

como el comienzo para futuros desarrollos e investigaciones en el campo de las tecnologías inalámbricas.

El desarrollo del módulo GetME en un lenguaje de programación como VoiceXML, a diferencia de otros lenguajes como Java, C, Tcl, permite a los desarrolladores de aplicaciones inalámbricas una fácil transportabilidad al teléfono celular, debido a que es un lenguaje meramente específico para este tipo de teléfonos celulares. Sin bien existe recientemente en el mercado un gran interés por parte de las compañías productoras de teléfonos celulares en buscar un lenguaje uniforme que permita la programación y desarrollo de aplicaciones inalámbricas, son solo algunas compañías como Nokia las que ofrecen utilerías como el J2ME para los desarrolladores de lenguaje Java, siempre y cuando estos cuenten con en el producto de desarrollo de Java que ellos requieren. El resto de las compañías productoras de teléfonos celulares solamente ofrecen herramientas para desarrollo SDK, que pueden ser usadas en conjuntos con herramientas como el de Nuance V-Builder e IBM Websphere Voice Server SDK.

La Interacción obtenida con el módulo GetME, ofrece expectativas sobre el acceso a la información, a los miembros de comunidades amplias, haciendo hincapié en personas con capacidades especiales dándoles así, la oportunidad de aprovechar los avances tecnológicos, sin importar el lugar donde se encuentre.

De las pruebas que se realizaron con el modulo GetME, en su versión beta de lectura de mensajes de voz obtuvimos una mejor amigabilidad con IBM WebSphere Voice Server, por contar con un módulo de idioma en español más extenso y menos robotizado, que el mostrado por el V-Builder, pero con la limitante en la conexión, debido a que solicitaba de módulos alternos, que no se obtuvieron a diferencia de V-Builder, a los que si se tuvo acceso, y se logro pasar el problema de la conexión.

En el modulo de mensajes de texto con V-Builder, consideramos obtuvimos lo resultados esperados a diferencia, de los obtenidos en la opción de mensajes por voz con IBM WebSphere Voice Server, por lo que propone se realicen futuras investigaciones que

permitan la conjunción de diversas tecnologías de información, para un mejor desarrollo de aplicaciones en VoiceXML en ambos intérpretes.

Ambas pruebas mostraron una restricción en el tipo de interfaz utilizada, debido a que no se muestra ninguna al estudiante, *pero no debemos perder de vista el desarrollo de un software para estudiantes con capacidades especiales, donde la interfaz visual no es de importancia.*

Con ambos intérpretes el V-Builder y IBM WebSphere Voice Server, enfrentamos limitantes similares, debido a que todo código en lenguaje VoiceXML, que se ejecute en dichos intérpretes, forzosamente necesitan realizar diversos enlaces a sus respectivos servidores. Resultando ser una barrera tecnológica para intentar acceder a algún repositorio de información de manera local en nuestra computadora debido a que, por características mismas de los intérpretes, necesita que los archivos o repositorios de información, se encuentren en sus servidores, y no en la ruta que uno como desarrollador indica.

Después de realizar pruebas de la aplicación, en diferentes intérpretes y navegadores de VoiceXML como el IBM WebSphere Voice Server y Nuance V-Builder, encontramos que, existe en la actualidad una gran restricción en el aspecto de la robustez de las aplicaciones, por el hecho de que dichos intérpretes son aún versiones de desarrollo (SDK), que requieren de una aportación económica a dichas compañías, para brindarnos el soporte técnico necesario en caso de requerirlo, así como de las versiones completas de los mismos. Además hay que incluir los costos por enlaces de comunicación para las pruebas, por parte de las compañías celulares.

En lo referente a los teléfonos celulares, la implementación de sistemas de reconocimiento de voz es cada vez más necesario, aprovechando nuevos algoritmos y herramientas que permitan una interacción mas completa con el módulo de GetME, y el teléfono celular donde se vaya a ejecutar el módulo de GetME.

6.10. Recomendaciones Futuras para la Migración del Módulo GetME.

- El tamaño de la aplicación, dependerá de los requerimientos mismo del sistema, y de la capacidad del teléfono celular, para poder almacenarla. Se recomienda contar con una tarjeta SIM, que permite tener almacenada la aplicación, y cargarla cuando sea necesario.
- La portabilidad de los mensajes por parte del módulo GetME, debe seguir los lineamientos proveídos por Cassiel-WAP.
- El módulo GetME, trabaja como un emulador de una interfaz de un teléfono celular (Motorola, Ericsson o Nokia), en un ambiente de integración de desarrollo como lo es V-Builder.
- El módulo GetME, puede ser actualizado, gracias a la transportabilidad del lenguaje VoiceXML.
- El funcionamiento del Servidor VoiceXML Gateway, es independiente del Servidor de Cassiel. Este realizará las consultas al Servidor de Cassiel, de acuerdo a las solicitudes realizadas vía el teléfono celular del estudiante, debido a que, utiliza solicitudes HTTP. El único requisito, es que se entiendan en un mismo lenguaje, como lo es JAVA, VXML y WML.