

Pontificia Universidad Católica de Valparaíso  
Facultad de Ingeniería  
Escuela de Ingeniería Informática

**Sistema para el Aprendizaje del Mapudungun.  
Incluyendo características de reconocimiento de voz  
y bot conversacional.**

**MARCO ANTONIO ALVARADO TRONCOSO**

**INFORME FINAL  
PARA OPTAR AL TÍTULO PROFESIONAL DE  
INGENIERO DE EJECUCIÓN EN INFORMÁTICA**

**DICIEMBRE 2012**

Pontificia Universidad Católica de Valparaíso  
Facultad de Ingeniería  
Escuela de Ingeniería Informática

**Sistema para el Aprendizaje del Mapudungun.  
Incluyendo características de reconocimiento de voz  
y bot conversacional.**

**MARCO ANTONIO ALVARADO TRONCOSO**

Profesor Guía: **Silvana Roncagliolo De La Horra.**

Profesor Co-referente: **Nibaldo Rodriguez Agurto.**

**Carrera: Ingeniería de Ejecución en Informática.**

**DICIEMBRE 2012**

*Dedicado a quienes hicieron de  
este viaje y los que vendrán  
una realidad.*

En primer lugar, agradecer a Dios por Su Compañía, Dirección y Fortaleza durante todo el proceso de formación. A mi familia y amigos por su apoyo incondicional, inmensurable e inmarcesible. Y a la Universidad por los conocimientos impartidos.

## Resumen

Aprendiendo con MAPU, es un sistema de escritorio para el aprendizaje autodidacta de elementos básicos de la lengua del pueblo Mapuche o Mapudungun. Enfocado, de forma no excluyente, a la población infanto-juvenil.

El presente proyecto consiste en agregar nuevas formas de interacción, por parte del usuario hacia el sistema. Estas formas incluyen el manejo del sistema y la obtención de información vía estímulo – respuesta. Para ello se hace uso de las tecnologías de reconocimiento de voz, para el control de la aplicación, y la tecnología de chatbots, para la obtención de información a través de estímulo – respuesta.

**Palabras claves:** Reconocimiento de voz, chatbot, Mapudungun, Aprendiendo con MAPU.

## Abstract

*Aprendiendo con MAPU*, is a desktop system for autodidact learning about basic elements of the Mapuche people language or Mapudungun. Focused, but no in an exclusionary way, to children and adolescents.

This project involves adding new interplay forms, from user to system. These forms include the system handling and obtaining information by stimulus-response. For this, it uses speech recognition technology to control the application, and chatbots technology to obtain information by stimulus-response.

**Key words:** Speech recognition, chatbot, Mapudungun, *Aprendiendo con MAPU*.

# Tabla de Contenidos

<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
<b>2. OBJETIVOS .....</b>	<b>2</b>
<b>2.1 Objetivo General .....</b>	<b>2</b>
<b>2.2 Objetivos Específicos.....</b>	<b>2</b>
<b>3. ESTADO DEL ARTE .....</b>	<b>3</b>
<b>4. APRENDIENDO CON MAPU .....</b>	<b>7</b>
<b>5. MAPUDUNGUN .....</b>	<b>8</b>
<b>5.1 Introducción al Mapudungun .....</b>	<b>8</b>
<b>5.2 Características básicas de los principales Grafemarios o alfabetos .....</b>	<b>8</b>
Grafemario Raguileo.....	10
Alfabeto Unificado .....	10
Grafemario AZÜMCHEFI.....	10
<b>5.3 Elección del Grafemario .....</b>	<b>10</b>
<b>6. SISTEMAS DE RECONOCIMIENTO DE VOZ .....</b>	<b>12</b>
<b>6.1 Introducción a los Sistemas de Reconocimiento de Voz. ....</b>	<b>12</b>
<b>6.2 Historia de los Sistemas de Reconocimiento de Voz.....</b>	<b>13</b>
Años 1950 y 1960 .....	13
Años 70s .....	14
Años 80s .....	14
Años 90s .....	14
2000 .....	15
El Futuro .....	16
<b>6.3 Características Básicas de un Sistema de Reconocimiento de Voz.....</b>	<b>16</b>
6.3.1 ¿Qué es un sistema de reconocimiento de voz?.....	16
Procesamiento:.....	16

Reconocimiento: .....	17
Comunicación: .....	17
6.3.2 Características que definen la complejidad de un Sistema de Reconocimiento de Voz	18
6.3.2.1 Tipos de Reconocimiento .....	18
Reconocimiento de palabras aisladas: .....	18
Reconocimiento de palabras conectadas .....	18
Reconocimiento de habla continua .....	19
6.3.2.2 Dependencia del locutor .....	19
Sistemas dependientes del locutor: .....	19
Sistemas independientes del locutor: .....	20
6.3.2.3 Características del Vocabulario .....	20
Vocabulario: .....	20
6.3.2.4 Otros Factores .....	21
Ruido: 21	
6.3.3 Enfoques de Reconocimiento de Voz .....	21
6.3.3.1 Enfoque Acústico-Fonético .....	21
6.3.3.2 Enfoque de Reconocimiento de patrones .....	22
6.3.3.3 Enfoque de Inteligencia Artificial .....	22
<b>6.4 Principales Tecnologías de Voz .....</b>	<b>22</b>
Control por comandos .....	22
Reconocimiento del hablante .....	22
Reconocimiento de voz .....	22
Síntesis de voz .....	23
Grabación de voz .....	23
<b>7. SYSTEM.SPEECH .NET FRAMEWORK 4.0 .....</b>	<b>24</b>
<b>7.1 Introducción a System.Speech Namespace .....</b>	<b>24</b>
<b>7.2 Formas de reconocimiento .....</b>	<b>24</b>
7.2.1 Shared .....	24
7.2.2 In-Process .....	25
<b>7.3 Gramáticas .....</b>	<b>25</b>
<b>8. CHATBOTS .....</b>	<b>27</b>
<b>8.1 Introducción a los chatbots .....</b>	<b>27</b>

<b>8.2</b>	<b>ALICE .....</b>	<b>28</b>
<b>8.3</b>	<b>AIML .....</b>	<b>30</b>
<b>8.4</b>	<b>El porqué de los bots conversacionales.....</b>	<b>30</b>
8.4.1	Bots conversacionales en la enseñanza de idiomas.....	31
<b>9.</b>	<b>PRIMERA ETAPA DEL DESARROLLO DEL PROYECTO: IMPLEMENTACIÓN RECONOCEDOR DE VOZ. ....</b>	<b>33</b>
<b>9.1</b>	<b>Características técnicas de Aprendiendo con MAPU .....</b>	<b>33</b>
<b>9.2</b>	<b>Reconocimiento de voz y Control por comandos.....</b>	<b>35</b>
<b>10.</b>	<b>SEGUNDA ETAPA DEL DESARROLLO DEL PROYECTO: DESARROLLO ENTORNO CHATBOT.....</b>	<b>40</b>
<b>10.1</b>	<b>Integración chatbot .....</b>	<b>40</b>
<b>10.2</b>	<b>Interfaz .....</b>	<b>41</b>
10.2.1	Imagen.....	42
10.2.1.1	Descripción.....	42
10.2.1.2	Desarrollo .....	42
10.2.2	Generador Texto.....	43
10.2.2.1	Descripción.....	43
10.2.2.2	Desarrollo .....	43
<b>10.3</b>	<b>Registros .....</b>	<b>43</b>
10.3.1.1	Descripción.....	43
10.3.1.2	Desarrollo .....	44
<b>10.4</b>	<b>Upload Registro .....</b>	<b>44</b>
10.4.1.1	Descripción.....	44
10.4.1.2	Desarrollo .....	44
<b>10.5</b>	<b>Diálogos .....</b>	<b>46</b>
10.5.1.1	Descripción.....	46
10.5.1.2	Desarrollo .....	46
<b>10.6</b>	<b>Análisis Informes.....</b>	<b>46</b>
10.6.1.1	Descripción.....	46
10.6.1.2	Desarrollo .....	46



<b>10.7 Update Base de Conocimiento .....</b>	<b>46</b>
10.7.1.1 Descripción.....	46
10.7.1.2 Desarrollo .....	47
<b>11. PRUEBAS DEL RECONOCEDOR DE VOZ.....</b>	<b>48</b>
<b>11.1 Desarrollo .....</b>	<b>48</b>
<b>11.2 Entorno.....</b>	<b>48</b>
<b>11.3 Pruebas .....</b>	<b>48</b>
<b>11.4 Conclusiones Prueba .....</b>	<b>50</b>
<b>12. CONCLUSIONES.....</b>	<b>51</b>
<b>13. BIBLIOGRAFÍA.....</b>	<b>52</b>

**ANEXO A *AZÜMCHEFI: CARACTERÍSTICAS BÁSICAS***

**ANEXO B *GRAMÁTICA MAPU***

## Índice de Figuras

Figura 1: Siri.....	3
Figura 2: Aprendiendo con MAPU. ....	7
Figura 3: Comparación vocales en los distintos Grafemarios .....	9
Figura 4: Comparación de Consonantes en los distintos Grafemarios.....	9
Figura 5: Polera de Apple que muestra lo complejo de los Sistemas de Reconocimiento.....	13
Figura 6: Modelo simplificado del funcionamiento de un SRV.....	17
Figura 7: Reconocimiento de Palabras Aisladas .....	18
Figura 8: Diagrama Secuencia Mapu - Normal.....	35
Figura 9: Diagrama de Secuencia Mapu - Voz.....	39

## Índice de Tablas

Tabla 1: Formularios Mapu .....	33
---------------------------------	----

# 1. Introducción

*El habla es la interfaz "ideal". "Es como deberíamos poder interactuar con las computadoras. Pero ¿cuándo debería empezar a sustituir el mouse y el teclado?" (1)*

La cita anterior plantea la interrogante, de cuando debería el reconocimiento de voz tomar el lugar que hoy en día tienen el teclado y el mouse, o últimamente los dispositivos táctiles.

De forma natural el habla ha sido la forma predominante en la interacción entre personas, pero no así entre persona y computador. Esta interacción, las tecnologías que manejan el habla a nivel informático, se llaman, sistemas de reconocimiento automático del habla o simplemente sistemas de reconocimiento de voz. La historia de esta tecnología es muy amplia, llena de contratiempos y éxitos, actualmente se encuentra en un estado de desarrollo bien avanzado, encontrándose disponible masivamente, desde dispositivos móviles inteligentes a nuestros computadores de uso diario. Con áreas de trabajo tan diversas como la medicina, el manejo del computador por voz, programas de accesibilidad y la enseñanza de idiomas.

Tomando como base el software Aprendiendo con MAPU, se realizaran cambios a su estructura, y se le añadirán características de control por comandos (el control de la aplicación a través de la voz) y la interacción del usuario con la aplicación a través de la implementación de un chatbot.

La disposición de los contenidos a lo largo de este informe es la siguiente:

Capítulo 2: Se presentan los objetivos del proyecto, tanto general, como específicos.

Capítulo 3: El estado del arte de las tecnologías involucradas en este proyecto.

Capítulo 4: Una introducción a Aprendiendo con MAPU, enfocada a sus objetivos.

Capítulo 5: Una breve introducción al Mapudungun, sus características básicas y la elección del Grafemario, que será la base idiomática del proyecto.

Capítulo 6: Enfocado a conocer sobre la tecnología de reconocimiento de voz, cómo es implementada, y sus usos.

Capítulo 7: Descripción de entorno de trabajo System.Speech.

Capítulo 8: Introducción a los chatbos, en especial a Alicebot y el lenguaje de programación AIML.

Capítulo 9: Primera etapa del desarrollo del Proyecto: Implementación reconocedor de voz.

Capítulo 10: Segunda etapa del desarrollo del Proyecto: Desarrollo entorno chatbot.

Capítulo 11: Pruebas del Reconocedor de voz.

Capítulo 12: Conclusiones.

## 2. Objetivos

### 2.1 Objetivo General

Implementar en Aprendiendo con MAPU características de reconocimiento de voz, para el control por comandos, y desarrollar un sistema de bot conversacional.

### 2.2 Objetivos Específicos

- ❖ Estudiar las características de los sistemas de reconocimiento de voz.
- ❖ Conocer la lengua Mapuche, mediante el estudio y la posterior selección de uno de sus Grafemarios.
- ❖ Analizar distintas técnicas para el desarrollo de un sistema de bot conversacional para *Aprendiendo con MAPU*.
- ❖ Habilitar el control por comandos, vía reconocimiento de voz para *Aprendiendo con MAPU*.
- ❖ Desarrollar un sistema de bot conversacional que sea capaz de poder mantener una conversación con el usuario, y que ésta le permita ampliar su comprensión del Mapudungun.

### 3. Estado del Arte

Últimamente el tema del reconocimiento de voz está muy de moda: el lanzamiento del iPhone 4S y de iOS 5 ha destacado por la inclusión de Siri -Figura 1, la asistente disponible en inglés que permite dar órdenes de voz y tener casi una conversación inteligente con estos Smartphones.



Figura 1: Siri

Lo cierto es que las tecnologías de reconocimiento de voz llevan ya mucho tiempo con nosotros, pero su uso ha estado limitado por la precisión de dichos desarrollos software, que a menudo no era la deseada. Esta tecnología ha mejorado en los últimos tiempos, pero parecía que los programas de reconocimiento de voz sólo servían para dictar largos textos de cuando en cuando. Siri ha demostrado que la aplicación de dichos sistemas puede ser muy útil. Junto con otras grandes empresas de la informática como Google que ya habían hecho algo muy similar con las búsquedas por voz y las acciones de voz de Android que también permiten ofrecer funciones similares, Microsoft con su sistema integrado de reconocimiento de voz funcionando ya desde Windows Vista, Simón en Linux, especialmente para entornos KDE y a nivel de investigación también se encuentran importantes proyectos como son HTK y Sphinx.

El estado de los bots conversacionales ha visto su punto máximo hasta el momento con el proyecto A.L.I.C.E y hasta la actualidad existen más de 166.00 bots conversacionales, sólo en uno de los tantos sitios de alojamiento web para estos bots.

Cada vez se está haciendo más cercana la posibilidad de entablar una comunicación oral, con un computador, donde éste sea capaz de entender y responder solicitudes de la misma forma que se haría con otro ser humano, pero para llegar a eso aún queda camino por recorrer; es más, la historia de los sistemas de reconocimiento de voz, es bastante amplia y con una -desarrollos van por un buen camino. A continuación se presenta una lista junto a una breve descripción, de los principales software, ya sean comerciales, libres o de investigación disponibles en el mercado.

**CMU Sphinx** CMU Sphinx (acortado como Sphinx), es el término general para describir un grupo de sistemas de reconocimiento de voz desarrollado en la Universidad de Carnegie Mellon. Incluye una serie de programas para reconocimiento de voz (Sphinx 2 - 4) y un entrenador de modelo acústico (SphinxTrain).

**Julius** Julius es una aplicación para reconocimiento de voz publicado como software libre con licencia BSD.1 Es un programa de reconocimiento de habla continua y en doble vía de lenguaje amplio. Está basado en trigramas (3gram) y en el Modelo oculto de Markov. Puede decodificar casi en tiempo real en la mayoría de computadores actuales. Su principal plataforma es Linux y otros sistemas Unix, también Windows. Julius ha sido desarrollado como parte de un kit de software libre para investigación en reconocimiento de habla continua de amplio vocabulario (LVCSR) desde 1977 y el trabajo ha sido continuado por el consorcio de reconocimiento de habla continua (CSRC) en Japón de 2000 a 2003. La versión más reciente de Julius es la 4.2, disponible desde el primero de mayo de 2011.

**Simon** Es un sistema de reconocimiento de voz que puede utilizarse para controlar la interfaz de usuario y que funciona especialmente bien con KDE, ya que ha sido escrito en C++ usando Qt y KDE para la interfaz de usuario, lo que garantiza su integración con KDE 4. No obstante, es posible usar Simon con otros entornos de escritorio y gestores de ventanas.

**VoxForge** VoxForge es un proyecto que tiene como objetivo recoger transcripciones de textos mediante voz, para ser usados en software de reconocimiento de voz (SRE) como pueden ser ISIP, HTK, Julius o Sphinx. El proyecto se encargará de categorizar y poner disponibles todos los archivos de voz que se envíen (también llamados "Speech Corpus") y los modelos acústicos bajo licencia GPL.

**MacSpeech Dictate** MacSpeech Dictate es un programa de reconocimiento de voz desarrollado para Mac OS X por MacSpeech. La primera versión de MacSpeech Dictate fue lanzada en marzo del 2008. El 20 de Septiembre de 2010, Nuance Communication, la cual adquirió MacSpeech en Febrero del 2010, lanzó una nueva versión al mercado, cambiando su nombre a "Dragon Dictate for Mac". MacSpeech Dictate Medical, una versión con vocabularios especializados para médicos y dentistas, fue lanzado en junio de 2009. MacSpeech Dictate Legal, con el vocabulario especializado para abogados, fue lanzado en julio de 2009. MacSpeech Dictate Internacional, con el apoyo para el habla reconocimiento en Inglés, francés, alemán e italiano, fue lanzado en septiembre de 2009.

**ViaVoice** Es un programa de reconocimiento de voz comercializado por IBM que permite que el computador transforme en texto las palabras dictadas por una persona. Es necesario un micrófono para realizar la transformación.

**Google Voice Search** El 14 de junio de 2011, Google hizo público que lanzaría esta aplicación y que ofrecería a los usuarios la búsqueda con reconocimiento de voz utilizando Google Chrome, con un simple clic en el ícono del micrófono puedes hacer una búsqueda con la voz.

**Google Maps con búsqueda por voz** En el verano de 2008, Google añadió la búsqueda de voz para móviles de Google Maps, permitiendo a los usuarios decir sus búsquedas, además de escribirlas.

**Google Mobile App** Esta aplicación permite a los usuarios buscar en Google por voz, aplicación muy parecida a la de Google Voice Search pero para móviles.

**Windows 7 built-in speech recognition** El reconocimiento de voz integrado de Microsoft, viene funcionando desde hace muchas versiones pero es a partir de Windows Vista que se le da más importancia. Consiste en dos módulos, uno de reconocimiento de voz y el otro de transcripción de texto a voz.

**Microsoft Kinect** La consola de videojuegos Kinect permite ser controlada a través del reconocimiento de voz.

**AT&T Watson** desarrollado por AT&T es un motor que integra una variedad de tecnologías del habla, incluyendo basados en red, reconocimiento automático del habla independiente del locutor, conversión de texto a voz, comprensión de lenguaje natural (incluye machine learning) y manejo de tareas a través de diálogos.

**Loquendo ASR** Loquendo ASR es un software de reconocimiento de voz, extremadamente preciso y veloz. Gracias al empleo de las tecnologías neurales más avanzadas y a su gran resistencia al ruido garantiza óptimas prestaciones incluso con vocabularios extensos, utilizando tanto gramáticas como modelos estadísticos del lenguaje. El resultado es un diálogo sencillo y natural, en una palabra "humano".

**TellMeMore** Sistema web para la enseñanza de idiomas que ofrece características de reconocimiento de voz para comprobar la correcta pronunciación de las palabras. Esta tecnología presenta muchas ventajas en diversos campos, como en la medicina, donde se le permite a los médicos, la transcripción automática de los procedimientos realizados. También en el ámbito de la enseñanza de nuevos idiomas, sistemas como TellMeMore, hacen uso de esta tecnología, tomando sus características para comparar patrones y permitirle al usuario saber en qué palabra está fallando.

Como fue dicho al comienzo de este capítulo, la tecnología crece y crece, cada vez con más efectividad en el reconocimiento, pero solo son soportados los principales idiomas del mundo, ya sea español, inglés, japonés, alemán, etc. Es debido a esto que surge la motivación de realizar este proyecto, el poder continuar en el desarrollo de un reconocimiento de voz para la lengua Mapudungun, enfocada principalmente a niños, para aumentar su interacción con el sistema e incentivarlos al aprendizaje de la cultura y el idioma Mapudungun.

El Mapudungun también ha tenido un gran crecimiento en las últimas décadas, impulsado por las demandas de los pueblos indígenas de Chile, las cuales estaban enfocadas a la conservación y desarrollo de su patrimonio cultural material e inmaterial; y el establecimiento de las bases educacionales en beneficio de las nuevas generaciones y sus proyecciones. A raíz de esta necesidad en el año 1995 es creado en Fondo de Cultura y Educación de la CONADI (Corporación Nacional de Desarrollo Indígena), y uno de sus programas, el que guarda mayor relación con los objetivos de este proyecto es el programa de Recuperación y Revitalización de las Lenguas Indígenas, cuyos objetivos incluyen:

- ❖ Contribuir a la recuperación de las lenguas vernáculas en la población infanto-juvenil indígena del país.
- ❖ Obtener lenguas vernáculas planificadas, en el ámbito oral y escrito, proporcionadas a población infanto-juvenil en la escuela.

- ❖ Desarrollar recursos educativos para la enseñanza/aprendizaje de las lenguas vernáculas proporcionadas a población infanto-juvenil en la escuela.
- ❖ Generar instancias de participación indígena en el desarrollo, promoción y valoración de las lenguas indígenas.

La combinación de ambas técnicas (Mapudungun + Reconocimiento de Voz) aún es muy escasa incluso a nivel informático en sí, no son muchos los proyectos en desarrollo o ya completados, la mayoría de ellos consiste en páginas web con funciones de diccionarios Español - Mapudungun o vice-versa, Traductores, páginas dedicadas a la promoción del idioma y la cultura Mapuche o inclusive una versión del sistema operativo de Microsoft Windows XP, el cual generó mucha polémica y discusión en las comunidades Mapuches llegando a entablar una demanda contra Microsoft. Pero los proyectos o sistemas más cercanos a los objetivos de este proyecto son:

**Mapulafkenche** (gente de lago y de la tierra) El cual es un software educativo que enseña a contar, reconocer los sonidos de los instrumentos y decir el nombre de los colores y las estaciones del año en Mapudungun, desarrollado por Estela Nahuelpán.

**Prototipo de traductor voz-texto Mapudungun-español (2):** Es un prototipo de software cuya función es la de traducir palabras en Mapudungun, ingresadas por voz, al idioma Español, usando técnicas de reconocimiento de voz. Para ello se hace uso de los Modelos Ocultos de Markov. Consta de un total de 150 palabras las cuales fueron grabadas durante tres meses en distintos establecimientos educacionales de la zona sur de Chile, desarrollado por David Cuminao Calluil, Universidad de La Frontera y financiado por CONICYT.

**ALICE (Artificial Linguistic Internet Computer Entity)** es un proyecto de Internet, que forma parte del Proyecto Pandora. Este proyecto consiste en la creación de bots de todo tipo, en especial los de chat. Desde la página de ALICE, el usuario puede entablar una conversación con un programa inteligente de conversación, que simula una charla real, de manera tal que el usuario puede difícilmente percatarse de que está hablando con un robot.



## 4. Aprendiendo con MAPU

Aprendido con MAPU nació como parte de un proyecto de asignatura, en particular Ingeniería de Software de la Carrera de Ingeniería de Ejecución Informática de la Pontificia Universidad Católica de Valparaíso (PUCV), fue desarrollado por cuatro estudiantes de esta carrera, de los cuales el autor de este proyecto formó parte, junto con la colaboración de distintas personas, entre ellas personal de la Dirección de Asuntos Indígenas de Villa Alemana. Fue desarrollado en el transcurso del segundo semestre académico del año 2011, año en el que participó en la feria de software de la Escuela de Ingeniería Informática de la PUCV, obteniendo el segundo lugar. La descripción y los objetivos, en el tiempo que fue creado son los siguientes:

**APRENDIENDO CON MAPU** -Figura 2- “es un software de auto aprendizaje, en forma didáctica y llamativa de la lengua Mapuche, enfocado principalmente a niños de 6 a 12 años de edad. Está dividido en 3 niveles, el primero de ellos está subdividido en 10 temas en los que se trata de enfocar lo más básico de la lengua. Además de esto cada tema cuenta con un mini cuestionario que permitirá al usuario ver si está progresando o no con el aprendizaje. El nivel 2 cuenta con frases típicas de la vida cotidiana, mientras que el nivel 3 es un cuestionario de todo lo anteriormente mencionado para evaluar el progreso del usuario.”

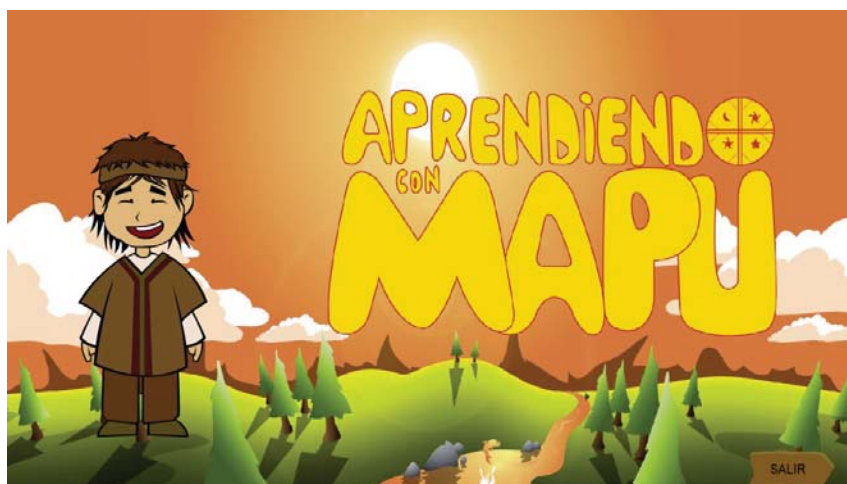


Figura 2: Aprendiendo con MAPU.

Su principal objetivo es el de devolver el interés a la población infanto juvenil en el idioma y la cultura Mapuche. Por parte del grupo siempre existió la intención de avanzar en el desarrollo de este proyecto, pero por diversos motivos este ha quedado estancado. Es por esto que ahora, y aprovechando el interés del autor en la tecnología de los sistemas de reconocimiento de voz, y la opción que se dio de poder realizar este tema como proyecto de título, se decidió enfocar este proyecto en agregar características de reconocimiento de voz, y de chatbot para Aprendiendo con MAPU, para así poder mejorar sus prestaciones. Este es parte del objetivo de este proyecto que se suma a los objetivos de Aprendiendo con MAPU con el único fin de acercar cada vez más y más a la población a la cultura y el idioma Mapuche.

## **5. Mapudungun**

### **5.1 Introducción al Mapudungun**

El Mapudungun o “el hablar de la tierra”, es el idioma de los Mapuches, pueblo amerindio que habita en Chile y Argentina. Si bien cada vez son menos los hablantes del Mapudungun, inclusive en aquellas personas de ascendencia Mapuche, en los últimos años han surgido muchos programas y planes de Gobierno, Grupos Mapuches y/o terceros con el único fin de conservar y revivir nuestra lengua y cultura materna. A través del análisis de los resultados del censo poblacional del año 2002 y la encuesta de opinión pública CEP 2006, se puede inferir que:

- ❖ El pueblo Mapuche es el de mayor presencia a nivel nacional (Chile) entre los pueblos indígenas del país.
- ❖ La población infanto juvenil es una de las de mayor presencia como rango etario entre la población Mapuche.
- ❖ Tanto las personas Mapuches como las no Mapuches concuerdan en que la característica más importante que define a una persona Mapuche es el habla del Mapudungun.
- ❖ La población Mapuche como la no Mapuche concuerdan en que el hablar la lengua Mapuche es un pilar fundamental para mantener la cultura Mapuche.

El estado actual de la lengua y la cultura Mapuche implica que si no hay esfuerzo en conjunto de los distintos participantes, prontamente el Mapudungun y los rasgos principales de la lengua Mapuche podrían desaparecer. Es por esto, y gracias a los esfuerzos de los distintos grupos involucrados en preservar la cultura y la lengua, unido por el afán de uniformar la escritura del Mapudungun, que han surgido diversos Grafemarios y alfabetos, tres de los cuales, los más aceptados por los lingüistas, académicos y comunidades Mapuches, serán presentados en los próximos subcapítulos.

### **5.2 Características básicas de los principales Grafemarios o alfabetos**

A lo largo de los años, diversos han sido los intentos por unificar la escritura de la lengua Mapuche y de esos intentos han surgido una gran variedad de Grafemarios, cada uno con sus ventajas y desventajas, cada uno con sus propias características, y cada uno más o menos aceptado por las comunidades, estudiantes, lingüísticos o académicos.

A continuación se presenta una tabla comparativa de los Grafemarios más utilizados a lo largo de Chile -Figura 3 y Figura 4. Tablas utilizadas para la elaboración de un Grafemario único.

ACADÉMICO	RAGUILEO	ANTINAO	CAÑULEF	PAINIQUEO	HUILCAMAN
a	a	a	a	a	a
e	e	e	e	e	e
i	i	i	i	i	y
o	o	o	o	0	0
u	u	u	u	u	w
ü	v	u	ú	g o h	

Figura 3: Comparación vocales en los distintos Grafemarios

Académico	Raguileo	Antinao	Cañulef	Painiqueo	Huilcaman
ch	C	Ch	C	ch	x
D	Z	D	D	dh	d
F	F	F	F	f o b	f
G	Q	G	G		g
K	K	K	K	k	k
L	L	L	L	l	l
LI	J	LI	Y		ll
M	m	M	m	m	m
N	N	N	N	n	n
Ñ	Ñ	Ñ	Ñ	ñ	ñ
P	P	P	P	p	p
R	R	R	R	rh	r
S	S	S	S	z	S
T	T	T	T	t	t
W	W	W	W	w	
Y	Y	Y	Y	y	
L	B		l		
N	H	Ng	N	Ng	
ng	G	Tr	"g	trh	
Tr	X		"r		
T			"t		
			S		

Figura 4: Comparación de Consonantes en los distintos Grafemarios.

Los Grafemarios o alfabetos que están disponibles para el estudio de la lengua Mapuche son diversos, pero solo tres de ellos son los más aceptados por las comunidades y los lingüistas, estos son, el Grafemario Raguileo, el Alfabeto Unificado y el Grafemario AZÜMCHEFI. A continuación se dará una breve introducción a cada uno de ellos, junto a sus características más importantes.

**Grafemario Raguileo** El Grafemario del escritor e investigador Anselmo Raguileo, es un alfabeto romano formado por 26 letras, en que cada sonido del idioma es representado por un solo signo; de modo que queda como sigue: A, C, Z, E, F, Q, I, K, L, B, J, M, N, H, Ñ, G, O, P, R, S, T, X, U, V, W, Y. Raguileo no reconoce la t interdental y omite el uso de letras distintas para el tono afectivo y el despectivo. Se evita el uso de dígrafos, lo que economiza espacio, y no hay ambigüedades cuando se forman grupos de consonantes en las palabras. Cuenta con el respaldo de varias organizaciones políticas Mapuches. Sus detractores le reprochan la falta de semejanza entre signos que representan fonemas relacionados, como L y B o N y H. También se afirma que este rasgo dificultaría su aprendizaje, tanto entre no hablantes de la lengua como entre los que fueron alfabetizados primero en castellano.

**Alfabeto Unificado** El Alfabeto Unificado de la Lengua Mapuche es un alfabeto romano con cinco adiciones (L, N, Ñ, T, Ü) y cuatro dígrafos (CH, LL, NG, TR); que fue elaborado por la Sociedad Chilena de Lingüística en 1986. Este alfabeto distingue como fonema separado la t interdental y reconoce hasta cierto punto los sonidos especiales del habla afectuosa, pero no de la despectiva. Por tanto, está formado por las siguientes 27 letras: a, CH, D, E, F, G, I, K, L, LL, M, N, , N, Ñ, NG, O, P, R, S, T, T, TR, U, Ü, W, Y.

**Grafemario AZÜMCHEFI** El Grafemario AZÜMCHEFI es un sistema de escritura del Mapudungun promovido por el Estado de Chile. Combina características del Alfabeto Unificado y del Grafemario Raguileo junto con adiciones originales. Fue creado por la Corporación Nacional de Desarrollo Indígena (CONADI). Es un alfabeto romano formado por 28 letras ordenadas de forma diferente a la convencional, recibe el nombre de Azümchefe (Persona que enseña) por sus siete primeras letras. Los signos en orden son los siguientes: A, Z, Ü, M, CH, E, F, I, K, T, NH, TX, O, Y, Q, G, LH, Ñ, R, S, LL, P, U, W, L, N, SH, T'.

### 5.3 Elección del Grafemario

La existencia y uso de diversas propuestas de Grafemarios para la lengua Mapuche ha impedido una adecuada comprensión de la gramática Mapuche, afectando también a la fonología de la lengua constándose algunas deformaciones, producto del uso de uno u otro instrumento o Grafemario, lo que causa dificultades e incomprensiones al usuario, sobre todo cuando es hablante. A raíz de la Ley Indígena (art 32, ley 19.253), que establece la necesidad de desarrollar un sistema de Educación Intercultural Bilingüe, se hizo necesaria la formulación de un instrumento único y oficial que posibilite la lecto-escritura de la lengua Mapuche, construido sobre la base de todos los antecedentes técnicos disponibles.

Así fue como un equipo multidisciplinario integrado por dirigentes de organizaciones sociales Mapuche, académicos, lingüistas y otros entendidos en el tema, redactó la propuesta de Grafemario denominada “Azümchefe. Grafemario Único del Idioma Mapuche”, que fue

sancionada por una asamblea de personalidades de la etnia en el Congreso de Niágara, Temuco, realizado el 22 y 23 de Junio de 1996.

Debido a su carácter oficial por parte del Gobierno de Chile, su uso en la mayoría de los programas oficiales de Gobierno relativos a la enseñanza del idioma, por sus característica función de facilitar el aprendizaje de la lecto-escritura de personas no hablantes de esta lengua y de los niños Mapuches y no Mapuches. Lo cual es el objetivo principal de Aprendiendo Con MAPU. Es por estos motivos que el Grafemario seleccionado es el AZÜMCHEFE. En el anexo *Azümchefe: Características Básicas* se detallan las características más importantes de este Grafemario obtenidas del documento oficial “AZÜMCHEFE Hacia la Escritura del Mapuzugun, POTENCIALIDAD LINGÜÍSTICA DEL GRAFEMARIO AZÜMCHEFE PALABRAS PRELIMINARES, Departamento de Cultura y Educación CONADI - Nacional, TEMUCO, Agosto de 2005.”

## 6. Sistemas de Reconocimiento de Voz

### 6.1 Introducción a los Sistemas de Reconocimiento de Voz.

El reconocimiento automático del habla es el procedimiento por el cual se convierte una señal acústica, capturada por un micrófono, en un conjunto de símbolos de un diccionario dado. Dichos símbolos están generalmente asociados con elementos semánticos de tipo palabra. (3)

Al hablar de sistemas de reconocimiento de voz (SRV), lo primero que se viene a la mente son las películas de ciencia ficción de la década de los noventa donde se ve al protagonista comunicarse de forma espontánea, natural a través del habla con su computador, el cual respondía, de forma hablada o realizando las tareas que éste le solicitaba, desde ese entonces hasta ahora no ha sido mucho el tiempo que ha pasado, pero si han sido muchos los avances en cuanto a la tecnología del reconocimiento de voz, encontrando esta tecnología desde los dispositivos móviles hasta la domótica, con funciones de dictado, o capaz de realizar las tareas solicitadas simplemente escuchando la voz junto y los comandos adecuados.

Ciertamente no es un área de la informática fácil de tratar - incluso grandes empresas de la informática como Apple tienen dificultades como se muestra en la Figura 5, en la cual se ve un juego de palabras indicando esta dificultad de reconocimiento, mostrando la similitud entre *I Helped Apple... Wreck a nice Beach* (estropear una linda playa) y *Recognize Speech* (reconocer el habla) - pues involucra muchas otras áreas de conocimiento, tales como, la fisiología, la acústica, el procesamiento de señales, la inteligencia artificial y la ciencia de la computación. El porqué de lo complejo de estos sistemas radica principalmente en el factor humano, los seres humanos muchas veces no son constantes en la expresión del habla, la forma de hablar, la velocidad, el tono, o incluso las mismas palabras variarían dependiendo de factores tales como nuestro estado anímico, estado de salud, el momento del día, etc. Muy difícilmente un SRV obtendrá siempre las mismas entradas, esto en el caso de que sea sólo para un hablante, o sea, en términos técnicos son sistemas dependientes del usuario, su complejidad aumentará aún más cuando se trate de sistemas independientes del usuario, los cuales deberán obtener entradas de diversos usuarios, muchas veces sin entrenamiento previo, de hecho es este entrenamiento previo el que permite a los SRV mejorar su porcentaje de efectividad en el reconocimiento de las palabras.

Se comenzó diciendo, que una de las dificultades que presenta el desarrollo de este tipo de sistemas, es el factor humano, también la cantidad de usuarios, pero hay otros dos factores realmente importantes a la hora de desarrollar sistemas de estas características, estos son: el primero de ellos es el tamaño de la gramática, es decir la cantidad de palabras a reconocer, ya que no es lo mismo el reconocimiento de voz para un sistema de discado telefónico, el cual a lo más tendrá una gramática de unas 20 o 30 palabras, que un sistema de dictado el cual debe ser capaz de reconocer cientos y miles de ellas. Y el segundo de ellos es la velocidad con la que se interactúa con el sistema, pudiendo ser éste capaz de reconocer solamente palabras aisladas, es decir palabras con una pausa mayor de la normal entre ellas, o palabras de habla continua, que es la forma natural como el ser humano se comunica.



Figura 5: Polera de Apple que muestra lo complejo de los Sistemas de Reconocimiento.

Si bien los desafíos que presenta el desarrollo de un sistema de reconocimiento de voz son muchos, también es cierto que las técnicas, las herramientas, y la capacidad de cómputo no sólo de los computadores sino también de los pequeños pero cada vez más inteligentes dispositivos que soportan esta tecnología va en aumento, lo que ha permitido que en las últimas décadas el reconocimiento de voz tenga un gran crecimiento, y que cada vez sea más común el poder comunicarse con estos dispositivos de manera natural, de la manera como se haría con otro ser humano.

## 6.2 Historia de los Sistemas de Reconocimiento de Voz

Mirando hacia atrás en el desarrollo de la tecnología de reconocimiento de voz es como ver a un niño crecer, progresando desde el nivel de habla de bebe, del reconocimiento de sílabas simples, a la construcción de un vocabulario de miles de palabras, para responder a las preguntas con respuestas rápidas, ingeniosas y comprendiendo la mayor parte de lo que recibe. Los sistemas vistos en capítulos anteriores hacen preguntarse hasta qué punto el reconocimiento de voz ha llegado en los últimos años. He aquí un vistazo a los acontecimientos en las últimas décadas que han hecho posible a las personas controlar sus dispositivos usando la voz.

**Años 1950 y 1960:** Los primeros sistemas de reconocimiento del habla solamente podían entender dígitos. (Dada la complejidad del lenguaje humano, tiene sentido que los inventores e ingenieros se centraran primero en los números.) Laboratorios Bell en el año 1952 diseña el “Audrey”, el cual reconoce los dígitos hablados por una sola voz. Diez años después, IBM en la Feria mundial de 1962 presenta su máquina "ShoeBox", la cual podía comprender 16 palabras pronunciadas en inglés. Laboratorios en los Estados Unidos, Japón, Inglaterra, y la Unión Soviética desarrollan hardware dedicado a reconocer sonidos hablados,

expandingo la tecnología de reconocimiento de voz soportando cuatro vocales y nueve consonantes. En estos años estos intentos pueden no parecer mucho, pero fueron un buen comienzo, especialmente si se considera lo primitivo de los computadores en esos tiempos.

**Años 70s** Uno de los grandes saltos de la tecnología de reconocimiento de voz se da en la década de los 70, debido a los intereses y el financiamiento del Departamento de Defensa de los Estados Unidos. El programa DARPA Speech Understanding Research (SUR) del departamento de defensa, activo entre los años 1971 al 1976, fue uno de los más grandes de su tipo en la historia del reconocimiento de voz, y entre otras cosas fue el responsable por el sistema de comprensión del habla “Harpy” de Carnegie Mellon. “Harpy” podía entender 1011 palabras, aproximadamente el vocabulario promedio de una persona de tres años de edad. La importancia de “Harpy” radica en que fue el que introdujo un enfoque más eficiente de búsqueda, llamada búsqueda en haz, la cual probaba la red de estado finito de las posibles sentencias, según Readings in Speech Recognition por Alex Waibel and Kai Fu Lee. Los años 70 también marcó algunos otros hitos importantes en la tecnología del reconocimiento de voz, incluyendo la fundación de la primera empresa comercial de reconocimiento de voz, la Threshold Technology, así también la introducción por parte de los laboratorios Bell de un sistema que podía interpretar las voces de distintas personas.

La historia del reconocimiento de voz está muy ligada a los avances en la metodología y técnicas de búsqueda, tal como lo demostró Google con la entrada al reconocimiento de voz, unos años atrás.

**Años 80s** Durante esta década, gracias a nuevos enfoques para comprender lo que dice la gente, los vocabularios de reconocimiento de voz saltan de unas pocas cientos palabras a varios miles de palabras, y tienen el potencial para reconocer un número ilimitado de palabras. Una razón importante fue un nuevo método estadístico conocido como el Modelo Oculto de Markov (o HMM por sus siglas en inglés). En lugar de simplemente usar plantillas para las palabras y la búsqueda de patrones de sonido, HMM considera la probabilidad de que sonidos desconocidos sean palabras. Esto sería la base del reconocimiento de voz en las próximas dos décadas. Equipado con estos vocabularios más extensos, el reconocimiento de voz comienza su camino para trabajar en aplicaciones comerciales para negocios e industrias especializadas (por ejemplo, usos médicos). Incluso entraron al hogar en la forma de una muñeca, la muñeca Worlds of Wonder's Julie (1987), la cual los niños podían entrenar para responder a sus voces.

A pesar de que el reconocimiento de voz, en esos momentos era capaz de reconocer mil palabras, así como lo logró el software de texto a voz (TTS por sus siglas en inglés) del año 1985 Kurzweil, o soportar un vocabulario de 5000 palabras, como lo logró el sistema de IBM, aún existe un importante obstáculo: estos programas funcionan a base de dictado discreto, es decir, tiene que haber una pausa significativa, mayor a lo normal, después de cada una de las palabras.

**Años 90s** En los años 90, los equipos con procesadores más rápidos finalmente llegan, y el software de reconocimiento de voz se vuelve viable para la gente común. En 1990, la empresa Dragon lanza el primer producto de reconocimiento de voz de consumo, Dragon Dictate, a un increíble precio de US\$9000. Siete años más tarde, se introduce una versión muy



mejorada Dragon Naturally Speaking. La aplicación reconocía el habla continua, así se podía hablar de forma natural, a alrededor de 100 palabras por minuto. Sin embargo, el programa tenía que ser entrenado por 45 minutos, y aún seguía siendo realmente caro US\$ 695. La llegada del primer portal de voz, VAL de BellSouth, fue en 1996; VAL fue un sistema de reconocimiento de voz interactivo, que suponía que daba información en base a lo que uno decía por teléfono. VAL allanó el camino para todas las inexactas tecnologías de menús activados por voz que aquejaron a los llamantes por los próximos 15 años.

**2000** En 2001, el reconocimiento de voz por computador había superado un 80% de precisión, y, casi al final de la década, los avances de la tecnología parecían estar estancados. Los sistemas de reconocimiento de voz, funcionaban bien cuando el universo del lenguaje era limitado, pero aun así seguían “adivinando”, con la ayuda de modelos estadísticos, entre palabras que suenan parecido, y el universo del lenguaje continuó creciendo a medida que internet creció.

El reconocimiento de voz y control por comandos fueron integrados en Windows Vista y Mac OS X, muchos usuarios no eran conscientes de la existencia de esta tecnología. El reconocimiento de voz de Windows y Voice Commands de OS X eran interesantes, pero no tan precisos y fáciles de usar como los viejos mouse y teclado.

El desarrollo de la tecnología de reconocimiento de voz comenzó a retomar su lugar en la vanguardia con un gran acontecimiento: la llegada de la aplicación de búsqueda de Google Voice para el iPhone.

El impacto de la aplicación de Google es significativo por dos razones. En primer lugar, los teléfonos celulares y otros dispositivos móviles son vehículos ideales para el reconocimiento de voz, así como el deseo de remplazar sus pequeños teclados sirve como incentivo para desarrollar mejores y alternativos métodos de entrada. En segundo lugar Google tenía la capacidad de descargar el procesamiento de su aplicación a sus centros de datos en la nube, aprovechando todo ese poder de cómputo para realizar el análisis de la gran cantidad de datos necesarios para hacer las combinaciones entre las palabras del usuario y el enorme número de muestras de habla humana que ellos reunieron.

En resumen el cuello de botella con el reconocimiento de voz, siempre ha sido la disponibilidad de datos, y la habilidad de procesarlos eficientemente. Las aplicaciones de Google añaden a su análisis, los datos de billones de consultas de búsqueda, para una mejor predicción de lo que uno probablemente está diciendo.

En 2010, Google añadió el reconocimiento personalizado a su búsqueda por voz (Voice Search) a los teléfonos Android, por lo que el software ahora podía grabar la búsqueda por voz de los usuarios y así producir un modelo de voz más preciso. En 2011 también fue añadida la búsqueda por voz a su navegador Google Chrome.

Se debe recordar cómo fueron los inicios de esta tecnología, de unas 10 a 100 palabras, y luego con el correr de las décadas, y de forma gradual a unas miles. En estos instantes la búsqueda en el idioma inglés de Voice Search incorpora 230 billones de palabras de consultas reales de usuarios.

**El Futuro** La explosión de aplicaciones de reconocimiento de voz indica que esta tecnología está para quedarse, y que su tiempo ha llegado, y se pueden esperar muchísimas más aplicaciones en el futuro. Estas aplicaciones no sólo permitirán controlar el computador u otro dispositivo compatible con la voz, o convertir voz en texto, sino que también serán compatibles con varios idiomas, ofrecerán una amplia variedad de voces para elegir, y se integraran con los dispositivos móviles.

Lo más importante, la calidad de los sistemas de reconocimiento de voz, también mejorará. Por ejemplo, algo no muy lejano es el *Sensory's Trulyhandfree Voice Control* el cual puede oír y entender incluso en entornos ruidosos.

Debido a que todas, o la gran mayoría de las personas se sienten más cómodas hablando en voz alta, no es difícil de imaginar, un futuro próximo, cuando vamos a estar al mando de las máquinas de café, hablar con la impresora, y decir que las luces se apaguen.

## 6.3 Características Básicas de un Sistema de Reconocimiento de Voz

### 6.3.1 ¿Qué es un sistema de reconocimiento de voz?

Un sistema de reconocimiento de voz es generalmente utilizado como una interfaz entre el ser humano y el computador, haciendo uso de algún software específico para este propósito y hardware especial.

Todo sistema de reconocimiento de voz debe cumplir por lo menos con las siguientes tareas:

**Procesamiento:** En esta etapa se convierte la entrada de voz a una forma que el reconocedor pueda procesar. Para poder convertir voz a texto o comandos que controlen el computador, éste debe pasar a través de varios y complejos pasos. Cuando se produce el habla, se crean vibraciones en el aire. El convertidor análogo a digital transforma esta onda análoga en datos digitales los cuales el computador es capaz de entender. Para poder realizar esto, se muestrea o digitaliza el sonido, tomando mediciones precisas de la onda en intervalos frecuentes. El sistema filtra el sonido digitalizado para eliminar el ruido no deseado y, a veces para separar en las diferentes bandas de la frecuencia (la frecuencia es la longitud de onda de las ondas de sonido, oídas por el humano como las diferencias de tonos). También normaliza el sonido, o se ajusta a un nivel de volumen constante. Además puede que tengan que ser alineadas temporalmente. La gente no habla siempre a la misma velocidad, por lo que el sonido debe ser ajustado para que coincida con la velocidad de las muestras de sonido de la plantilla que ya están almacenadas en la memoria del computador.

A continuación la señal se divide en pequeños segmentos tan cortos como unas pocas centésimas de segundo, o incluso milésimas en el caso de los sonidos de las consonantes explosivas – consonante en las cuales se detiene el flujo de aire producido por la obstrucción en el tracto vocal – como “p” o “t”. Entonces, el programa empareja esos segmentos a fonemas conocidos en el idioma apropiado. Un fonema es el elemento más pequeño de una

lengua, una representación de los sonidos que se hacen y unen para formar expresiones significativas.

**Reconocimiento:** Identifica lo que se dijo (Traducción de señal a texto). Este paso que a simple vista parece sencillo, pero que en realidad es el más difícil de lograr, es el foco de la mayoría de las investigaciones en el campo de reconocimiento de voz. El programa analiza los fonemas en el contexto de los otros fonemas alrededor de él. Se ejecuta la trama de fonemas contextuales, a través de un complejo modelo estadístico y los compara con una gran biblioteca de palabras, frases y oraciones más conocidas. El programa determina entonces lo que el usuario estaba diciendo y, o bien envía como texto o emite un comando al computador.

**Comunicación:** Envía lo reconocido al sistema (Software/Hardware) que lo requiere. En la Figura 6, se encuentra un Modelo simplificado del funcionamiento de un Sistema de Reconocimiento de Voz.

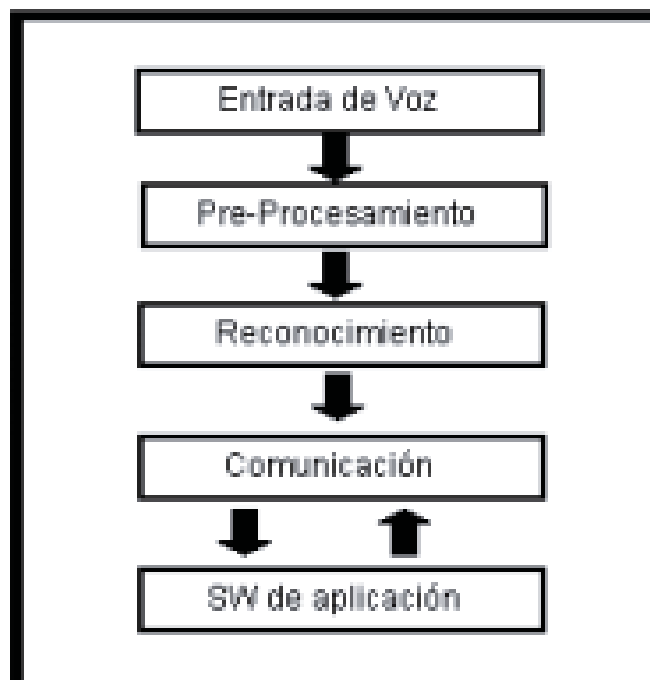


Figura 6: Modelo simplificado del funcionamiento de un SRV

Existe una comunicación bilateral en aplicaciones, en las que la interfaz de voz está íntimamente relacionada al resto de la aplicación. Estas pueden guiar al reconocedor especificando las palabras o estructuras que el sistema puede usar.

Otros sistemas sólo tienen una comunicación unilateral.

Los procesos de pre-procesamiento, reconocimiento y comunicación deberían ser invisibles al usuario de la interfaz. El usuario sólo lo nota de manera indirecta como: certeza en el reconocimiento y velocidad de respuesta. Estas características son utilizadas para evaluar una interfaz de reconocimiento de voz.

## 6.3.2 Características que definen la complejidad de un Sistema de Reconocimiento de Voz

### 6.3.2.1 Tipos de Reconocimiento

**Reconocimiento de palabras aisladas:** En este tipo de reconocimiento se asume que el locutor pronuncia de forma deliberada palabras con pausas suficientemente grandes entre ellas. Esto no quiere decir que acepte sólo palabras, sino que requiere una sola expresión a la vez tal como lo muestra la Figura 7. El hecho de que se puedan localizar de forma sencilla las fronteras entre palabras o expresiones simplifica mucho la tarea de reconocimiento. Es por esto de que esta es la forma más simple de estrategia de reconocimiento de voz.

Este tipo de reconocimiento es muy conveniente para ciertas aplicaciones, particularmente aquellas en las que se envían órdenes a una máquina a través de comandos consistentes en una palabra de un pequeño vocabulario.

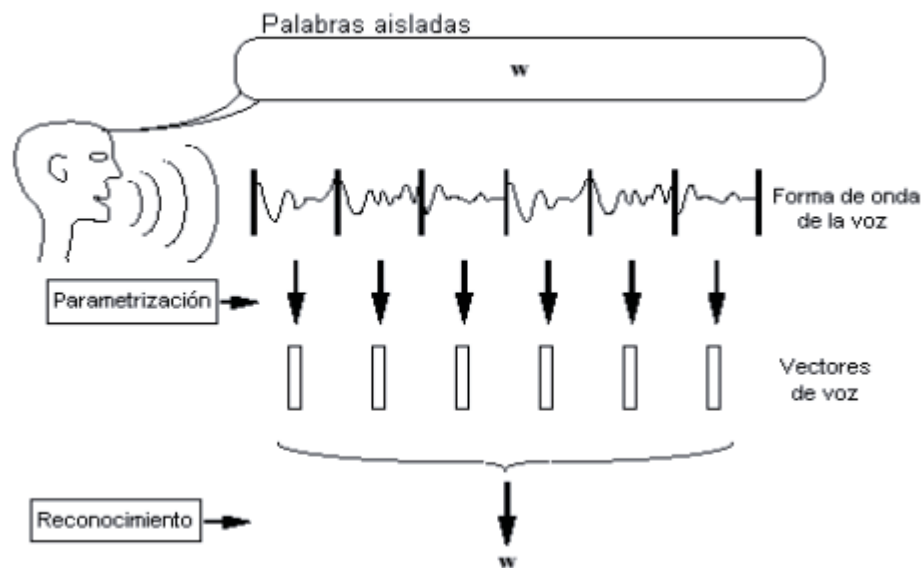


Figura 7: Reconocimiento de Palabras Aisladas

**Reconocimiento de palabras conectadas:** En aplicaciones de reconocimiento de voz continua con pequeños vocabularios, a veces se utiliza el denominado reconocimiento de voz conectada. Este término, se refiere a la estrategia de reconocimiento más que a la voz en sí misma. En general la voz se pronuncia de una forma continua.

En esta técnica, la sentencia se decodifica conectando modelos y comparando la declaración completa con esos modelos concatenados. El sistema normalmente no intenta modelar efectos alofónicos presentes en las fronteras de las palabras, ni efectos de coarticulación intra e inter palabras. Hay una hipótesis implícita que es que aunque no se pueden localizar fronteras inequívocas entre palabras, las palabras están razonablemente bien

articuladas. En general, esta suposición está violada por el locutor, pero los resultados se mejoran con la cooperación del locutor.

Cuando se conocen relaciones probabilísticas entre palabras (sintaxis) éstas se pueden explotar para mejorar las prestaciones del reconocimiento de voz de palabras conectadas.

**Reconocimiento de habla continúa:** Los sistemas de reconocimiento más complejos son los que realizan reconocimiento de voz continua, en los que el usuario pronuncia el mensaje sin ningún tipo de restricción. En primer lugar, el reconocedor debe ser capaz de tratar con fronteras temporales desconocidas en la señal acústica. En segundo lugar, el reconocedor debe ser capaz de funcionar bien en presencia de efectos co-articulatorios y pronunciaciones descuidadas (inserciones y omisiones de fonemas) que acompañan al habla fluida y natural.

Puesto que el reconocimiento de voz continua no requiere en el caso extremo ninguna cooperación del locutor, debe compensar este hecho empleando algoritmos robustos a los infinitos matices presentes en el lenguaje natural. Estos sistemas de reconocimiento de voz son los más naturales desde el punto de vista de los usuarios. Son esenciales en muchas aplicaciones en la que grandes poblaciones de usuarios interactúan con el reconocedor.

La falta de pausado es sólo uno de los comportamientos no cooperativos. Otros son la inclusión de voces extrañas o ruido, y el uso de palabras fuera del vocabulario.

### 6.3.2.2 Dependencia del locutor

Uno de los conceptos más buscados en el campo del reconocimiento de voz es la independencia del locutor. Un sistema independiente del locutor, tal como lo dice su nombre, es un sistema capaz de reconocer voz de cualquier locutor, sin previo entrenamiento (o muy poco en comparación a los sistemas dependientes del locutor). En cambio un sistema dependiente del locutor es un sistema entrenado para reconocer la voz de un locutor en particular.

**Sistemas dependientes del locutor:** Los sistemas dependientes del locutor necesitan, normalmente, una gran cantidad de datos de entrenamiento del locutor propuestos antes de ser capaz de reconocer su voz con exactitud. La mayoría de los sistemas de reconocimiento de voz implementados durante la década de los 80, y antes, fueron de este tipo y, por lo tanto, no eran capaces de reconocer voz de un nuevo locutor sin que esto implicara una considerable disminución en su efectividad.

Otro problema de los sistemas dependientes del locutor es que sólo los puede utilizar el locutor para el que se han entrenado. Por lo tanto, si se han de soportar varios locutores se necesita una gran cantidad de memoria para guardar los modelos para los diferentes locutores. Además con varios conjuntos de modelos el cambio automático entre locutores puede convertirse en una tarea compleja.

Existen todavía más problema con este tipo de sistemas, lo que ha motivado a que los investigadores se inclinen cada vez más hacia la independencia del locutor. La edad, la condición de salud, la fatiga, y el estrés son algunos de los fenómenos que pueden causar

efectos más adversos sobre los sistemas dependientes del locutor, comparado a los sistemas independientes del locutor. Además, las aplicaciones multi-locutor y multi-entorno precisan un sistema independiente del locutor.

**Sistemas independientes del locutor:** Los sistemas independientes del locutor tienen ventajas sobre los sistemas dependientes del locutor. Es deseable que cualquier nuevo usuario de un sistema que hace uso del reconocimiento de voz, no se vea obligado a hablar durante horas para entrenar el reconocedor, antes de que éste sea capaz de reconocer su voz. Además después de esta tarea, el usuario debe esperar un tiempo considerable mientras se entrena el sistema para que acepte su voz.

El principal inconveniente que poseen estos sistemas, que es lo que además logra su gran ventaja sobre los sistemas dependientes del locutor, es el tiempo requerido en etapa de desarrollo en grabaciones de diversos usuarios, y de decenas a centenas de frases lo que da horas y horas de grabación, por ejemplo para la grabación de un pequeño vocabulario de unas 50 o 100 palabras se requieren mínimamente 10 repeticiones por palabra, tiempo que muchas veces es muy difícil de obtener en pequeños proyectos.

Estos sistemas, tienen muchas ventajas en diferentes aspectos del reconocimiento de voz, por lo que se han hecho cada vez más populares y pueden alcanzar tasas de error muy pequeñas en el reconocimiento. La técnica más popular para lograr independencia del locutor es utilizar una gran base de datos que contenga un gran número de locutores, pronunciando cada uno unas pocas frases, las cuales idealmente deben estar fonéticamente equilibradas. El reconocedor entrenado con este material, si se ha entrenado de forma adecuada es capaz de reconocer la voz de cualquier nuevo locutor con un nivel aceptable de prestaciones.

### 6.3.2.3 Características del Vocabulario

**Vocabulario:** Los vocabularios o diccionarios son listas de palabras o expresiones que pueden ser reconocidas por el sistema de reconocimiento de voz. Generalmente los vocabularios más pequeños son más fáciles de reconocer para un computador, mientras más extenso sea el vocabulario más difícil esta labor será. A diferencia de los diccionarios normales, cada entrada no tiene que ser necesariamente una sola palabra, sino que ésta puede estar formada por una sentencia o dos (ejemplo: “activar voz”). Los vocabularios pequeños pueden tener una o dos expresiones compuestas reconocidas, en cambio, los más grandes pueden tener cientos o miles.

**Clasificación de los vocabularios:** El tamaño del vocabulario de un sistema de voz afecta a su complejidad, requerimientos de proceso y precisión del sistema. Algunas aplicaciones sólo requieren unas pocas palabras (por ejemplo, únicamente cifras), otros en cambio requieren diccionarios enormes (por ejemplo, máquinas de dictado). No hay definiciones establecidas, sin embargo se suele utilizar la siguiente clasificación:

- ❖ Vocabulario pequeño – decenas de palabras
- ❖ Vocabulario mediano – cientos de palabras
- ❖ Vocabulario grande – miles de palabras
- ❖ Vocabulario muy grande – decenas de miles de palabras

#### 6.3.2.4 Otros Factores

**Ruido:** Los reconocedores de voz entrenados en entornos limpios, proporcionan resultados de reconocimiento aceptables cuando procesan voz limpia. Sin embargo, en las aplicaciones reales de reconocimiento de voz los reconocedores tienen que procesar voz adquirida en entornos degradados en mayor o menor medida por la presencia de ruido.

Por ejemplo en las aplicaciones informáticas de reconocimiento de voz, como pueden ser los módulos de gestión por voz del sistema operativo o los software de dictado, la voz está usualmente contaminada por ruidos procedentes del computador. Si la aplicación opera en una oficina, la señal de voz se ve afectada además por ruidos tales como conversaciones de fondo, teléfonos, impresoras, fotocopadoras, etc.

El ruido da lugar a reducciones importantes en la precisión de los reconocedores de voz, debido principalmente a dos factores, por una parte el ruido se superpone a la señal de voz, modificando sus características y enmascarando la señal. Por otro lado en presencia de ruido el locutor modifica su forma de pronunciar, lo que se conoce como el efecto Lombard, lo que puede provocar que la utilización de SRV capaces de reconocer en entornos ruidosos, llevada a ámbitos reales sea poco efectiva. Por esto, en el diseño de una aplicación que incluye reconocimiento de voz es importante conocer las condiciones acústicas en las que se va a operar, y así aplicar las técnicas adecuadas para poder obtener una tasa de reconocimiento aceptable. En la actualidad se dedica un gran esfuerzo de investigación al reconocimiento de voz en entornos ruidosos.

### 6.3.3 Enfoques de Reconocimiento de Voz

#### 6.3.3.1 Enfoque Acústico-Fonético

Consiste en detectar sonidos elementales y asignarles determinados rótulos. La base de este enfoque es la hipótesis de que en el lenguaje hablado existe un número finito de unidades fonéticas distintas (fonemas) y que estas unidades pueden caracterizarse por un conjunto de propiedades acústicas que se manifiestan en la señal hablada en función del tiempo. Si bien las propiedades acústicas de los fonemas son altamente variables con el locutor y con los fonemas vecinos (co-articulación de sonidos), se asume que las reglas que gobiernan la variabilidad son simples y pueden ser aprendidas fácilmente por el sistema de reconocimiento.

El reconocimiento consiste básicamente en dos pasos:

Primer paso: segmentación y rotulado. La señal es dividida en regiones acústicas a las que son asignados uno o más fonemas, resultando en una caracterización de la señal de voz mediante un reticulado de fonemas.

Segundo paso: se trata de determinar una palabra (o conjunto de palabras) válida a partir de la secuencia de fonemas rotulados en el primer paso. Se introducen en esta etapa restricciones lingüísticas tales como vocabulario, sintaxis, y reglas semánticas.

### 6.3.3.2 Enfoque de Reconocimiento de patrones

Consiste básicamente en dos pasos:

**Primer paso:** entrenamiento de patrones

**Segundo paso:** comparación de patrones

La característica principal de este enfoque es que usa un marco matemático bien definido y que establece representaciones consistentes de los patrones de voz que pueden usarse para comparaciones confiables a partir de un conjunto de muestras rotuladas, usando algoritmos de entrenamiento. La representación de los patrones de voz puede ser una plantilla (template), o un modelo estadístico (HMM: Hidden Markov Model), que puede aplicarse a un sonido (más pequeño que una palabra), una palabra, o una frase.

En la etapa de comparación de patrones se realiza una comparación directa entre la señal de voz desconocida (a reconocer) y todos los posibles patrones aprendidos en la etapa de entrenamiento, de manera de determinar el mejor ajuste de acuerdo a algún criterio.

### 6.3.3.3 Enfoque de Inteligencia Artificial

En este enfoque se intenta automatizar el procedimiento de reconocimiento de acuerdo a la forma en que una persona aplica su inteligencia en la visualización, análisis y caracterización de la voz basada en un conjunto de características acústicas. Algunas técnicas que se emplean son: sistemas expertos que integran conocimientos prácticos fonéticos, sintácticos y semánticos, para la segmentación y el rotulado, y usan herramientas tales como redes neuronales artificiales para aprender las relaciones entre eventos fonéticos.

## 6.4 Principales Tecnologías de Voz

**Control por comandos:** Tecnología que permite al usuario interactuar de forma más natural con un sistema software. Esta tecnología se caracteriza por dar la opción al usuario de realizar las tareas que normalmente se realizan con los dispositivos de entrada típicos (mouse y teclado) usando su voz, los sistemas que soportan esta tecnología emplean palabras reservadas para realizar ciertas tareas específicas, como por ejemplo, presionar un botón, abrir un archivo, guardar, iniciar algún programa, etc. Esta tecnología es de ayuda para las personas con capacidades distintas, incluyéndose en la mayoría del software de accesibilidad.

**Reconocimiento del hablante:** Tecnología que es capaz a través de la voz, y de las características únicas de cada hablante, ser capaz de poder reconocerle. Empleado mayoritariamente en sistemas de seguridad.

**Reconocimiento de voz:** La tecnología de reconocimiento de voz es de las más complejas en lo que a tecnologías de voz se refiere. El reconocimiento de voz tiene por objetivo la comunicación hablada entre humano y computador. Es el proceso mediante el cual se analiza una entrada de voz u onda sonora y se trata de obtener el mensaje que contiene, para traducirlo a otro tipo de dato como texto. (4)



**Síntesis de voz:** La tecnología de síntesis de voz, también conocida como Text-To-Speech (TTS), tiene como objetivo la comunicación entre el computador y el humano, esta tecnología consiste en generar voz como salida a partir de un texto proporcionado como entrada y de un conjunto de sonidos previamente grabados que se combinan para producir el resultado deseado. (4)

**Grabación de voz:** Es la tecnología más sencilla de manejar y la más conocida y utilizada, consiste en grabar la voz en archivos de audio, mediante software y hardware especializado. Otras tecnologías más complejas hacen uso o se basan en ésta, como por ejemplo, síntesis de voz.

## 7. System.Speech .Net Framework 4.0

### 7.1 Introducción a System.Speech Namespace

Microsoft Windows incluye un sistema de reconocimiento de voz conocido como *Windows Desktop Speech* (5), el cual se encuentra disponible en Windows Vista y las versiones posteriores de Windows, además de Windows XP service pack 3. El motor de reconocimiento de voz permite digitalizar sonidos y extraer palabras o frases desde los datos procesados.

El motor de voz de *Windows Desktop Speech* puede ser usado por los desarrolladores utilizando el framework .NET a partir de la versión 3.0. Todas las clases y métodos requeridos para el desarrollo de aplicaciones de reconocimiento de voz son provistas por el System.Speech.Recognition namespace. Este namespace puede ser utilizado de muchas formas distintas, por ejemplo; permite el reconocimiento de voz de forma síncrona o asíncrona, las entradas de audio varían desde los dispositivos por defecto, es decir, los micrófonos, o inclusive archivos de audio en formato WAV, también permite el uso de su gramática incorporada, para ser usada en forma de dictado, o se puede crear una gramática específica para dar más control a la aplicación.

### 7.2 Formas de reconocimiento

#### 7.2.1 Shared

La clase *SpeechRecognizer*, se usa para crear aplicaciones que accedan al motor de reconocimiento de voz provisto por Windows, este motor ejecuta reconocimiento de voz para cualquier aplicación Windows. La forma de entrada de este tipo de reconocedor es el dispositivo por defecto establecido para el computador. La ventaja de utilizar este tipo de reconocedor es que Windows provee toda la implementación, por lo que prácticamente a nivel de desarrollador no son muchas las configuraciones que se deben realizar.

Este tipo de reconocimiento es ideal para las aplicaciones de dictado ya que toda la gramática es provista por el motor; al igual que la interfaz, la cual indica el estado del reconocedor y del reconocimiento actual.

Otra ventaja que posee, es que aplicaciones que hagan uso de este tipo de reconocimiento por primera vez automáticamente se dirigen al proceso de entrenamiento del reconocedor.

La gran desventaja que posee, es la de ser un motor compartido, todas las aplicaciones hacen uso de una misma instancia del motor, por lo que muchas veces se pierde el control de la aplicación, por ejemplo si la aplicación desarrollada tiene como palabra clave “inicio”, al ser “inicio” una palabra reservada del sistema operativo, puede abrir el menú de inicio en vez de ejecutar la acción correspondiente en la aplicación.

Otra desventaja que posee es que no todas las opciones de configuración están bajo el control de la aplicación.

### 7.2.2 In-Process

La clase *SpeechRecognitionEngine*, crea para la aplicación un motor de reconocimiento de voz conocido como In-Process Recognizer, su gran ventaja es la desventaja de los sistemas Shared, es decir, el control adicional que se obtiene en las aplicaciones, debido a que los reconocedores In-Process son solamente usados por el proceso que instancia el objeto.

Además se pueden utilizar los parámetros provistos por los constructores de la clase para seleccionar un reconocedor que se ajuste a criterios específicos, como por ejemplo, lenguaje-culture, que define el idioma del reconocedor y la variante de esté (eng-USA o sp-ESP, por nombrar algunos).

Los pasos necesarios para preparar una instancia de *SpeechRecognitionEngine* son las siguientes:

1. Indicar el tipo de entrada para recibir la señal de voz.
2. Cargar la gramática, esto puede ser de forma síncrona o asíncrona.
3. Registrar eventos de reconocimiento y crear manejadores (*handlers*) para esos eventos.
4. Iniciar el reconocedor, ya sea de forma síncrona o asíncrona.

Se ha visto que los algunos de los problemas que deben manejar lo reconocedores de voz son los relacionados al ruido de fondo y silencios que acompañan a la entrada de voz. Para mejorar la respuesta frente a estos problemas, se cuenta con las propiedades *BabbleTimeout*, *EndSilenceTimeout* y *EndSilenceTimeoutAmbiguous*.

## 7.3 Gramáticas

Las gramáticas en los sistemas de reconocimiento de voz son de lo más importantes, ya que ellas son las que manejan lo que el reconocedor puede o no puede identificar.

En *System.Speech* existen tres formas de crear estas gramáticas:

- ❖ Crear gramáticas usando *GrammarBuilder*.
- ❖ Crear gramáticas usando *SrgsGrammar*.
- ❖ Crear gramáticas usando XML.

La clase *GrammarBuilder* provee una forma liviana que permite a través de programación crear gramáticas, la cual es lo suficientemente poderosa para cumplir con los requerimientos de muchos escenarios de reconocimiento de voz.

El crear gramáticas con esta clase se adecua mejor a gramáticas que tienen sólo una regla que contiene una lista, o tal vez una lista de listas. Las gramáticas más complejas

debieran usar otra forma para ser creadas, como lo son SrgsGrammar y XML ambas muy similares entre sí.

SrgsGrammar y XML ambas hacen uso de la especificación *Speech Recognition Grammar Specification Version 1.0* (6) dada por la W3C.

La diferencia entre ambas es que una, SrgsGrannar, es creada en el programa, a través de código y métodos disponibles, y la otra, XML, es creada en un archivo externo, que debe mantener el formato establecido por la recomendación, de extensión XML.

Una lista detallada con la gramática utilizada en Mapu se encuentra en el anexo B “Gramática Mapu”.

## 8. ChatBots

### 8.1 Introducción a los chatbots

Los chatbots son programas, software, algoritmos que utilizan procesamiento de lenguaje natural (NLP: Natural Language Processing) en un sistema de preguntas y respuestas (QA systems: question-answering systems.) Estos sistemas han sido definidos también como sistemas expertos que usan razonamiento basado en casos (CBR: case base reasoning) (7).

Hay en la actualidad un gran número de sistemas de preguntas y respuestas o chatbots. Algunos de ellos han competido en un concurso llamado “The Loebner Prize in Artificial Intelligence”, el cual es una primera aproximación al Test de Turing (8). El concurso es anual y se presentan, en EE.UU., diferentes chatbots que funcionan en inglés, gana el que se aproxime más a un ser humano en su modo de responder (9).

Los chatbots que han obtenido mejores resultados e incluso han sido utilizados con algún fin concreto distinto de la investigación académica son aquellos que utilizan AIML (un lenguaje de marcas creado para el desarrollo de chatbots) o aquellos basados en ELIZA (10). Hay dos chatbots que utilizan las características de ELIZA y AIML e implementan algunas mejoras: Ultra Hal y Jabberwacky, de los cuales no se posee más que transcripciones de sus conversaciones (11).

El análisis de las conversaciones entre estos programas y personas indica entre otros problemas, la falta de evaluación de un “contexto”, las respuestas que da un chatbot son en su mayoría resultado sólo de la última entrada de datos del usuario. Por ejemplo para una entrada como “Hola”, un chatbot siempre responde con algún saludo (elegido de manera aleatoria.) y no evalúa si el saludo tiene lugar a la mitad de una conversación o al principio de ésta, o si ya fue saludado o no. Antes de seguir hay que definir “contexto” en este ámbito.

Toda conversación toma lugar en un determinado contexto. Si bien "contexto" es una palabra que encierra un contenido amplio, queda restringido su significado en una conversación vía chat. En una conversación entre personas, el contexto está dado por las características de los interlocutores, sexo, edad, altura, vestimenta, etc., por el lugar en el que se encuentran, el momento del día y del año, por lo que se dijeron antes, por lo que dijo un tercero, por una mirada, un gesto, entre otras, pero todas ellas son cosas que pueden percibir (o ya percibieron) ambos, no importa si es una percepción de tipo auditiva, visual o de otro tipo. Estas otras imágenes sensoriales complementan al diálogo de manera que el significado correcto de cada frase sólo se obtiene cuando se evalúa el conjunto y no tan solo la frase literal.

Como en una conversación vía chat, dos entidades que se comunican no comparten más que sus palabras, todo el contexto queda limitado a eso, a lo que se dijo anteriormente. A pesar de la reducción notable de lo que habría que evaluar para deducir el contexto, el problema acerca de lo que define el contexto sigue siendo válido. El chatbot deberá deducir información contextualizada de su interlocutor a partir de lo que éste le diga. Y en conversaciones prolongadas, a lo largo de días, "lo que se dijo antes" se dividirá en lo que "ya se sabe de esta persona" y "lo que acaba de contar."

Finalmente el contexto puede estar también definido por las palabras utilizadas y el orden de éstas en una sentencia, aún sin tener en cuenta sentencias previas. Por ejemplo en la sentencia “un delincuente escapó ayer de presión” es claro que “presión” es un error y que la palabra correcta es “prisión” el contexto me indica esto por lo siguiente:

Las palabras “delincuente” y “escapó” están relacionadas con “prisión”, “prisión” suena y se escribe parecido a “presión”. La sentencia entera es gramaticalmente incorrecta y no significa nada “escapar de presión”. La sentencia entera, quitando la palabra “presión” limita las posibilidades a unas pocas, si no se contara con más información, las suposiciones lógicas más probables son “prisión”, “de las autoridades”, o similares.

ELIZA y los chatbots basados en ese sistema, no realizan una evaluación del contexto. Básicamente funcionan buscando palabras claves en la frase de entrada. Estas palabras claves están asociadas a conjuntos de reglas que sirven para descomponer la frase original y armar una respuesta (10). Cuando el chatbot no tiene reglas asociadas o no encuentra palabras claves en una frase, uno de los procedimientos que puede ejecutar consiste en armar una respuesta acerca de alguna frase dicha anteriormente por el usuario. De esta manera ELIZA simula darle continuidad a una conversación como si en efecto la estuviese siguiendo. Este comportamiento es una primera aproximación al seguimiento del contexto.

AIML funciona básicamente con una lista de frases patrón y cuando una frase de entrada del usuario coincide con una de esas frases patrón se devuelve la respuesta asociada a ese patrón (7). Sin embargo para frases de entrada cortas este comportamiento no es suficiente y se hace necesaria la evaluación del contexto. Entonces entra en juego el tag `that`, que agrega una condición más a ser evaluada, la respuesta anterior del chatbot. Cuando este tag está presente en una `category`, no sólo debe coincidir la frase de entrada del usuario con la frase patrón sino que además lo que el chatbot respondió anteriormente debe coincidir con el contenido de `that`.

Este mecanismo le permite a AIML tener una evaluación del contexto más sofisticada y precisa que ELIZA. Sin embargo esta alternativa es poco flexible ya que no permite una disgresión en el diálogo y es inaplicable para todas las posibles combinaciones, por ejemplo, pueden ser la respuesta a infinidad de preguntas distintas, muchas de ellas dependerán a su vez de contextos más generales.

## 8.2 ALICE

ALICE (Artificial Linguistic Internet Computer Entity Chatterbot) es un chat robot o robot conversacional, que hace uso de la inteligencia artificial y el procesamiento del lenguaje natural, está basado en un experimento especificado por Alan M. Turing en el año 1950. ALICE utiliza AIML, un lenguaje de programación derivado de XML que fue diseñado por ALICE Artificial Intelligence Foundation Inc, para la creación de chat robots del tipo estímulo-respuesta.

ALICE y AIML son muchas veces vistos como una simple extensión de ELIZA. Si bien esta comparación es correcta en términos de la arquitectura estímulo-respuesta que ambos

comparten, ALICE bot tiene a la actualidad más de 40.000 categorías de conocimiento, mientras que la original ELIZA tenía solamente alrededor de 200. Otra innovación fue provista por la web, la cual permitió la recopilación a gran escala de muestras de datos en lenguaje natural.

El modelo de aprendizaje de ALICE se denomina aprendizaje supervisado, debido a que una persona, llamada el botmaster, juega un papel crucial. El botmaster monitorea las conversaciones del robot y crea nuevos contenidos AIML para obtener respuestas más adecuadas, precisas, creíbles, “humanas”, o lo que pretenda el botmaster.

A su vez cuenta con algoritmos, desarrollados por la fundación, para la detección de patrones en los datos de las conversaciones. Este proceso llamado “*targeting*”, ofrece al botmaster nuevas entradas de para patrones las cuales aún no tienen respuestas específicas, permitiendo un proceso casi continuo de refinamiento supervisado sobre el bot.

Se ha argumentado que Turing, cuando predijo que una máquina podría jugar su juego en “50 años más” (8), después de que su paper fuera publicado en el año 1950, previó algo más, algo así como una máquina de aprendizaje de propósito general, la cual aún no existe. El concepto es bastante simple: construir un robot que crezca como un niño. Capaz de comprender el lenguaje de la forma en que los humanos lo hacen. En estos términos, el rol del botmaster será totalmente automático. Pero incluso, un niño no puede, o al menos no debiera, ir por el mundo, sin protección, aprendiendo el lenguaje “en las calles”, sin supervisión.

La generación automática de preguntas y respuestas para el chatbot, parece ser similar al problema de confianza que deben enfrentar los niños abandonados. La gente es simplemente demasiada poco fiable, para que ellos puedan enseñar al chatbot sin ningún tipo de supervisión.

Muchos clientes intentarán deliberadamente sabotear al bot con información falsa. Por lo que aún se necesita la presencia de un editor, un supervisor, un maestro o simplemente un botmaster, para filtrar lo adecuado de lo inadecuado.

El cerebro de ALICE consiste en alrededor de 41.000 elementos llamados categorías. Cada categoría combina una respuesta y una pregunta, o un estímulo y su respuesta, llamados “*pattern*” y “*template*” respectivamente. El intérprete AIML almacena los patrones en una estructura tipo árbol, gestionada por un objeto llamado *Graphmaster*, implementando un patrón de almacenamiento y algoritmo de emparejamiento. El *Graphmaster* se encuentra comprimido en la memoria, por lo que permite un tiempo eficiente emparejamiento de patrones.

### 8.3 AIML

El AIML, o Artificial Intelligence Mark-up Language es un lenguaje de programación derivado de XML. Fue diseñado específicamente para ayudar en la creación de la primera entidad chatbot informática de lenguaje artificial online ALICE Aunque descrito muy ampliamente, el lenguaje AIML está especializado en la creación de agentes software con lenguaje natural, conocidos como Alicebots.

El lenguaje de programación AIML fue desarrollado por el Dr. Richard Wallace y la comunidad de código abierto Alicebot entre los años 1995 y 2000. Con él se crearon las bases del primer Alicebot, ALICE, el cual ha ganado el concurso Loebner Prize Contest, tres veces el premio The most human, y también ganó el Campeonato de Chatbot en 2004. Desde sus comienzos en internet, gracias a que el lenguaje AIML y los programas Alicebot son código-abierto, se han creado numerosos clones de Alicebot basados en la implementación original del programa y de la base de conocimiento AIML. El desarrollo de este lenguaje continua, y en el otoño de 2004 se ha creado una nueva versión del conjunto ALICE AIML. La implementación más activa hoy en día es pandorabots.com, un hosting de bots online y compañía de desarrollo que permite el alojamiento gratuito de Alicebots.

### 8.4 El porqué de los bots conversacionales

Los estudiantes de un lenguaje extranjero, en el ámbito de este proyecto el Mapudungun, comúnmente tiene muy pocas oportunidades de usar su idioma de estudio (idioma objetivo). Los profesores, dan lo mejor de sí para crear oportunidades durante sus clases a través de trabajos en parejas o grupales, pero una variedad de factores que van desde la falta de tiempo a la timidez de los alumnos o las oportunidades limitadas de retroalimentación de calidad en el proceso de diálogo lo impiden.

Un sistema que permita la conversación interactiva, confiable y realística entre computador y persona, usando lenguaje natural, revolucionaría la interacción persona-computador, y el rol que los computadores juegan en el diario vivir. Esas interfaces, permitirían que la información estuviera disponible para los usuarios a cualquier hora y en cualquier lugar, incorporando la flexibilidad y la eficiencia del lenguaje natural. Dejando de lado las enormes dificultades técnicas y teóricas que surgen al crear este tipo de interfaces, avances significativos han sido realizados durante las últimas décadas. La demanda comercial de interfaces conversacionales es amplia, a pesar de que el rendimiento de la mayoría de esos sistemas no es de lo mejor (en comparación a la prestación que se pide en otro tipo de sistemas).

Los chatbots pueden proveer un medio para la práctica del lenguaje para los estudiantes en cualquier ocasión y virtualmente, en cualquier lugar. Y a través de esta sección se ve el porqué del uso del bot conversacional como medio para la práctica del Mapudungun en específico, o cualquier lengua extranjera en general, y el uso del reconocimiento de voz como medio atrayente para la interacción de los niños con el sistema.



### 8.4.1 Bots conversacionales en la enseñanza de idiomas

El proceso de aprendizaje de un idioma extranjero no es fácil. Incluso bajo las mejores condiciones los estudiantes enfrentan diferencias culturales, problemas de pronunciación, falta de motivación, carencia de respuestas efectivas por parte del otro interlocutor, la necesidad de aprender lenguaje especializado, y muchos otros obstáculos durante sus estudios. Los estudiantes de idioma extranjero generalmente enfrentan esos desafíos teniendo muy pocas o en algunos casos ninguna oportunidad de usar su idioma de estudio más allá de la sala de clases. Durante las clases, los estudiantes pueden o no pueden practicar con otros estudiantes y solamente en las clases más pequeñas pueden tener la oportunidad de practicar con su profesor.

La tecnología está abriendo nuevas posibilidades para el aprendizaje de idiomas. Como lo describe Benson "...internet es un apoyo sumamente fuerte en dos situaciones básicas para el aprendizaje auto dirigido: los estudiantes pueden aprender dondequiera que deseen utilizando virtualmente un ilimitado rango de material auténtico". Una de las áreas que internet ha abierto es el uso de los bot conversacionales para la práctica y el aprendizaje del lenguaje.

A pesar de que los chatbots desde sus inicios no fueron diseñados como un 'profesor' de lenguaje o incluso como una herramienta explícita para la enseñanza de idiomas, en su estado actual son una herramienta valiosa para los estudiantes y profesores de idiomas.

Los chatbots son comunicadores por naturaleza, por lo que pueden ayudar a los estudiantes con la práctica que necesiten, con el repaso y la confidencialidad.

A continuación se da una lista de formas en las cuales los chatbots pueden ser potencialmente útiles para los profesores en el proceso de aprendizaje de una lengua extranjera.

- **Conversación libre:** En la sala de clases o en el hogar, con computadores en cada escritorio, es una buena manera de dar a los estudiantes la oportunidad de experimentar. El poder tener una conversación libre, tal vez alejada de los tópicos tratados en clase le da la oportunidad a los estudiantes de expandir su área de estudios. Dependiendo de la base de conocimiento del bot, los temas de conversación pueden ser amplios y variados, y teniendo la seguridad de que al menos por parte del bot (si éste ha sido bien programado) la respuesta siempre será la correcta en su aspecto gramatical.
- **Repaso:** Este es el uso más práctico de los chatbots. En situaciones de enseñanza de lengua extranjera es común para los estudiantes el pasar una clase revisando material que nunca tendrán la oportunidad de realmente usar. El bot le brinda esa oportunidad, le da la oportunidad, de que en cualquier situación y en cualquier lugar se pueda repasar lo aprendido en clases.
- **Auto Análisis:** Algunos chatbots ofrecen una función de llevar y revisar las conversaciones realizadas. Esto puede ser una buena forma para que los estudiantes se auto-evalúen, evalúen a sus compañeros o incluso al bot.

- **Para el profesor:** Dependiendo del desarrollo, algunos chatbots traen la opción para que el profesor pueda realizar un seguimiento de las conversaciones de los alumnos con el chat y obtener una idea de cómo ha sido el progreso de ellos, en qué parte del lenguaje necesitan ayuda, y tal vez lo más importante, qué tipo de tópicos ellos desean aprender más.

Está claro que los chatbots deben parecer lo más humano posible si se quiere usarlos para ser una herramienta realmente útil para los estudiantes de idiomas. Deben tener sus familias (virtuales), historias, gustos y disgustos. En palabras simples, deben tener su propia vida. Mientras más creíbles sean como seres humanos, mayor es la calidad de las potenciales conversaciones.

Ya en el año 1992, Reeves y Nass, consistentemente mostraron que la gente responde a este tipo de interfaces de una forma social; ellos la tratan – interfaz - en algunos aspectos, como si fuera una persona real la que se encuentra interactuando con ellos. Reeves y Nass sugieren que la reacción de los usuarios a tales sistemas, puede mejorar significativamente si se prestara mayor atención a la personalidad de las interfaces, que a los problemas de ingeniería del procesamiento del lenguaje natural complejo.

Los chatbots para auto aprendizaje a través de conversaciones casuales, fuera de clase, pueden ser útiles para estudiantes de niveles avanzados. Para los de niveles más bajos o incluso para los novatos, el diseño de un bot enfocado en tareas específicas puede ser una mejor herramienta de aprendizaje. Existe una gran cantidad de potenciales tópicos, y así también un amplio rango de niveles de estudiantes, desde principiantes hasta casi nativos, para los cuales los chatbots pueden ser diseñados para interactuar con ellos.

## 9. Primera etapa del desarrollo del proyecto: Implementación reconocedor de voz.

### 9.1 Características técnicas de Aprendiendo con MAPU

*Aprendiendo con MAPU* fue desarrollado en el lenguaje de programación Visual Basic 2010, haciendo uso del framework .NET 4.0, consiste básicamente de formularios, eventos del mouse y archivos multimedia, junto con un archivo XML que es utilizado para el control del avance y los datos de usuario.

De una forma más detallada Mapu cuenta con contenedor principal en el cual son cargados todos los formularios que hacen de *padres*, es decir, desde los cuales provienen otros formularios. Cada formulario padre, reemplaza a quien lo llamó, al momento de llamar a un formulario padre desde otro formulario este último se cierra.

Los formularios hijos, provienen de los formularios padres, estos en su mayoría tratan sobre los temas de estudio que provee Mapu, estos se abren en estado maximizado, sin cerrar al su formulario padre.

Todos los temas del Nivel I cuentan con instrucciones sobre el uso, la escritura y la pronunciación en un audio (archivo WAV), de la palabra u objeto que se está estudiando, junto a una “lección” que se trata de una animación flash, a modo de cuestionario, relacionado con el tema tratado, que necesita de al menos 7 de 10 respuestas correctas para aprobar. En la Tabla 1 se encuentra una lista con los formularios de Mapu, el tipo de estos y una breve descripción de sus características principales.

Tabla 1: Formularios Mapu

Nombre	Tipo	Breve Descripción
Principal	Padre	Es el contenedor principal de todos los formularios Padre. Está presente durante toda la ejecución de Mapu
Login	Padre	Desde aquí se accede al menú principal del sistema, además, proporciona el lugar para seleccionar el usuario con el cual acceder al sistema, la creación de nuevos usuarios y la posibilidad de acceder a la animación de introducción.
Índice	Padre	Lo principal aquí es la selección del nivel al cual acceder, el avance obtenido en cada nivel, el juego <i>Atrapa los piñones</i> , y cerrar sección del usuario actual. Además de una pequeña introducción en audio.
Nivel I	Padre	Provee el temario del sistema, desde aquí se accede a los principales temas de estudio del Mapudungun. También da la posibilidad de volver al Índice o Menú Principal
Nivel II	Padre	Simplemente contiene un diálogo a modo de imágenes y labels en Español y Mapudungun. Este nivel será reemplazado por Mapu Skin y el Alicebot.
Nivel III	Padre	Este nivel posee un simple formulario con preguntas de diversa índole sobre los temas tratados en el Nivel I. Este Nivel aún se encuentra en desarrollo y no será tomado en cuenta para éste proyecto.

Abecedario	Hijo	Aquí se introduce al usuario al alfabeto Mapudungun, en especial al definido por el Grafemario Azümcheffi.
Animales 1	Hijo	Se muestran fotos de animales con sus sonidos, pronunciación y escritura en Mapudungun.
Animales 2	Hijo	Se muestran fotos de animales, distintos a los del tema anterior, con sus sonidos, pronunciación y escritura en Mapudungun.
Astros	Hijo	Se presentan los principales astros del pueblo Mapuche, junto con una pequeña descripción de ellos.
Cosas de Casa 1	Hijo	Se presentan algunos artículos presentes en el hogar.
Cosas de Casa 2	Hijo	Se presentan más artículos que los anteriores presentes en el hogar.
Colores	Hijo	Se muestran colores, cambiando el color de fondo y una imagen representativa del color.
Comidas	Hijo	Se muestran algunas comidas típicas Mapuches.
Cuerpo Humano	Hijo	Se muestra a Mapu y el usuario haciendo click puede conocer la pronunciación y la forma de escritura de esa parte del cuerpo en Mapudungun.
Estaciones del Año	Hijo	Aquí se da información sobre las estaciones del año y los puntos cardinales.
Familia	Hijo	Se muestra un árbol familiar con los distintos parientes de Mapu.
Tu Familia	Hijo	Se le permite realizar al usuario su propio árbol familiar.
Números	Hijo	Se muestran los principales números en el Mapudungun, junto con explicar cómo es su conformación y ejemplos de su escritura.
Acerca de	Hijo	Muestra información sobre los desarrolladores y colaboradores del proyecto Mapu.
Instrucción	Hijo	Presente información sobre cómo utilizar cierta parte del sistema.
Otoño	Hijo	Aparece en el tema Estaciones del año, muestra información sobre el otoño.
Invierno	Hijo	Aparece en el tema Estaciones del año, muestra información sobre el invierno.
Primavera	Hijo	Aparece en el tema Estaciones del año, muestra información sobre la primavera.
Verano	Hijo	Aparece en el tema Estaciones del año, muestra información sobre el verano.
Skin	Hijo	El skin de Mapu, será la interfaz para interactuar con el Alicebot.

Las modificaciones realizadas a Mapu, tienen que ver principalmente con “ordenar” el código fuente, dando nombre significativos a las variables utilizadas, modularizando y optimizando ciertas funcionalidades.

Conociendo los distintos formularios a continuación, Figura 8, se presenta un diagrama de secuencia con los pasos básicos y genéricos para poder estudiar un tema. El tema elegido será astros.

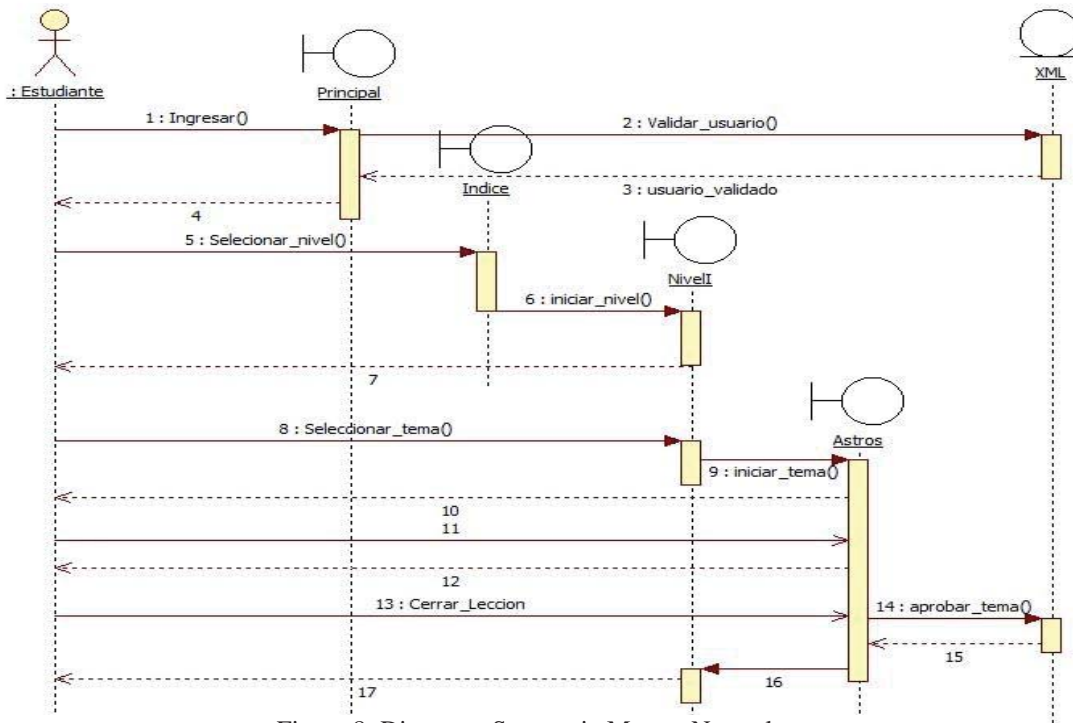


Figura 8: Diagrama Secuencia Mapu - Normal

## 9.2 Reconocimiento de voz y Control por comandos

En el apartado anterior se vio el funcionamiento normal de Mapu, en este apartado, se explicará cómo se realizó uno de los objetivos de este proyecto, el control por comandos. Se detallarán algunas de las clases y métodos de *System.Speech* utilizados para la implementación, las características de esas clases y métodos, y por qué fueron elegidas.

Primeramente, el tipo de motor de reconocimiento utilizado fue el motor *In-Process*, detallado anteriormente, su elección en desmedro del *Shared*, se debe principalmente al mayor grado de control que se logra en la aplicación, control muy necesario en sistemas de reconocimiento de voz para control por comandos.

Para hacer uso de este motor, primeramente se debe importar la librería *System.Speech.Recognition*, la cual provee el acceso a ambos reconocedores.

Cabe señalar que para hacer uso de la tecnología de reconocimiento de voz en Visual Basic 2010 se debe agregar al proyecto la referencia a *System.Speech*, en este caso en particular la versión 4.0.

Una vez agregadas las referencias, e importadas las librerías es posible comenzar a utilizar la tecnología de reconocimiento de voz.

Para el desarrollo del proyecto se hace uso de una sola instancia del reconocedor, utilizada por todos los formularios, además de una gramática contenida en un archivo XML

externo que concuerda con el estándar SRGS de la W3C sobre gramáticas para reconocimiento de voz.

Primeramente, se inicializa el motor de reconocimiento de voz, de la siguiente forma:

```
Dim recognizer As New SpeechRecognitionEngine //creado como
                                         variable Global.
Recognizer = New SpeechRecognitionEngine() //Inicialización.
```

Posteriormente, se procede a crear y cargar la gramática correspondiente:

```
Dim grammar As New Grammar ("Path", "nombre_regla", Nothing)
Recognizer.LoadGrammarAsync(grammar)
```

Luego de haber creado el reconocedor y cargado su primera gramática se procede a definir el dispositivo de entrada y la forma de reconocimiento, esta forma de reconocimiento puede ser de dos tipos, una es la cual puede reconocer sólo una palabra o frase, y la otra es la que es capaz de reconocer múltiples palabras o frases, hasta que se le indique (vía programación) cuando debe dejar de reconocer.

```
recognizer.SetInputToDefaultAudioDevice() //Se establece que el
dispositivo de entrada es el dispositivo por defecto.

recognizer.RecognizeAsync(RecognizeMode.Multiple) //Se establece el
modo de reconocimiento, en este caso, el de múltiples frases.
```

Luego se crean los distintos manejadores (*handlers*) para tratar con los diversos eventos generados por el proceso de reconocimiento o manejo de gramáticas.

```
AddHandler recognizer.SpeechDetected, AddressOf detectado
AddHandler recognizer.SpeechRecognitionRejected, AddressOf noreco
AddHandler recognizer.SpeechRecognized, AddressOf reco
AddHandler recognizer.LoadGrammarCompleted, AddressOf gm
AddHandler recognizer.RecognizerUpdateReached, AddressOf upd
```

Una vez creados estos manejadores, donde cada uno de ellos cumple una función específica en relación al reconocimiento y la gramática, se proceda a hacer uso de ellos.

Ahora se procederá a explicar la gramática de Mapu, como fue dicho esta gramática está contenida en un archivo externo de formato XML, este archivo debe respetar los estándares propuestos por la W3C, uno de ellos es el formato del *header*, este debe poseer

información sobre la versión, codificación y el tipo de gramática. A continuación se muestra el *header* utilizado en el archivo de gramáticas de Mapu:

```
<?xml version="1.0" encoding="utf-8"?>
<grammar xml:lang="es-ES"
tag-format="properties-ms/1.0" version="1.0"
xmlns="http://www.w3.org/2001/06/grammar"
xmlns:sapi="http://schemas.microsoft.com/Speech/2002/06/SGSExtensions">
```

Aquí se indica la versión, en este caso 1.0, la codificación, utf-8, y el idioma de la gramática, es-ES, esta información es muy importante ya que cualquier error hará que el intérprete XML no pueda procesar la gramática, y al no procesar la gramática el reconocimiento de voz no podrá ser llevado a cabo.

A continuación se presenta una porción de la gramática que posee Mapu:

```
<rule id="nivel_dos" scope="public">
  <one-of>
    <item>Frases</item>
    <item>Indice Principal</item>
  </one-of>
</rule>
<rule id="nivel_tres" scope="public">
  <one-of>
    <item>Cuestionario</item>
    <item>Indice Principal</item>
  </one-of>
</rule>
```

Fueron mostradas dos reglas, estas reglas tienen dos parámetros, primeramente un id, el cual define el nombre de la regla, y segundo el parámetro scope, el cual define la visibilidad de ésta. Dentro de la regla se encuentran dos tipos de *tags*:

- *One-of*: El tag one-of le indica al intérprete que puede seleccionar, o “reconocer” solamente uno de los ítems, contenidos dentro de él.
- *Ítem*: Es la unidad básica, se trata del elemento que el intérprete procesa, y es lo que devuelve al reconocedor.

Por ejemplo, en el formulario Nivel II, existen dos botones, Frases e Índice Principal, estos dos botones forman la gramática de la regla *nivel\_dos*, debido a que el usuario puede decir cualquiera de los dos para obtener un reconocimiento.

La gramática de Mapu está compuesta de los comandos necesarios para cada formulario, cada uno de ellos separado por una regla específica. Como es posible apreciar, puede existir más de una regla que posea el mismo ítem, esto generaría un problema en el reconocimiento, ya que por ejemplo, al decir Índice Principal, ambos formularios activarían la función que muestra el índice principal.

Este problema fue solucionado de la siguiente forma:

1. Se hace uso de una sola gramática, la cual contiene muchas reglas.
2. Al momento de cargar un formulario, se hace uso del procedimiento *update\_grammar* de la clase principal, con un parámetro de tipo string que contiene el nombre del formulario actual, que es equivalente al nombre de su respectiva regla en la gramática.
3. Este *update\_grammar* cambia el valor de la variable global que posee el nombre de la regla en uso "*nombre\_regla*", y llama al método *RequestRecognizerUpdate()*.
4. *RequestRecognizerUpdate()*, genera un evento de gramática, pausando el proceso de reconocimiento, y llamando al procedimiento *upd()*.
5. Este procedimiento descarga todas las gramáticas del reconocedor, posteriormente crea una nueva gramática, con la regla actual y la carga al reconocedor.

De esta forma se evita el choque de reconocimiento que sucedería si se tuvieran todos los ítems en una misma regla; al ir cambiando dinámicamente de regla dependiendo del formulario que se encuentre en uso.

Se ha visto cómo se manejan las reglas y cómo se inicializa el reconocedor, ahora bien, todo esto se realiza para lograr el reconocimiento, proceso que será explicado a continuación:

1. Cuando el usuario genera una entrada de audio, se llama al manejador *SpeechDetected*, el que fue redirigido al procedimiento *detectado*, este procedimiento simplemente indica cuando un evento de voz fue detectado, sin importar si fue reconocido o no.
2. Si una entrada de voz no fue reconocida, es decir, no se pudo encontrar su equivalente en la gramática, se genera el evento *SpeechRecognitionRejected*, redirigido a *noreco*, el cual indica que la entrada no fue reconocida.
3. Ahora bien si el reconocedor logra ubicar una entrada de voz con un ítem de la gramática se genera el evento *SpeechRecognitionRecognized*, redirigido a *reco*, el cual da toda las instancias para poder manejar el texto reconocido.
4. Una vez dentro de *reco*, se llama a la clase *cases* y su procedimiento *funciones*, con parámetros "e" que se refiere al evento *Recognized* y un String que representa el valor de la gramática actual.
5. El procedimiento *funciones*, recibe el evento y la gramática actual, y dependiendo de la clase actual se ubicará en uno de los diferentes *cases*, desde los cuales obtiene las opciones a realizar.



6. Cuando se ha ubicado el case correspondiente, se extrae la cadena de texto reconocida con `e.Result.Text.ToUpper`, convirtiéndola a mayúsculas, al obtener la cadena de texto reconocida se procede a ubicarla dentro del case en el que se encuentra, del paso 5, cuando lo encuentra, obviamente estará, ya que son los mismos valores que se encuentran en la gramática, genera el evento de mouse o teclado correspondiente.

Se ha visto de forma escrita como es el proceso de reconocimiento, los procedimientos y componentes que intervienen en el dicho proceso y cómo estos fueron implementados en Mapu.

Las ventajas de esta forma de control de la aplicación radican en la mayor interacción que logrará el usuario, al sentir un pleno control de la misma, y abrir las posibilidades a una tecnología que si bien no es nueva, sí ha sido muy poco explotada, por lo que llamará la atención de los usuario que dadas su características siempre buscan nuevas formas de interacción.

A continuación en la Figura 9: Diagrama de Secuencia Mapu - Voz, y al igual que como fue hecho en el apartado anterior se presentará un diagrama que explicará el proceso de acceder a una clase, pero esta vez haciendo uso del sistema de reconocimiento de voz.

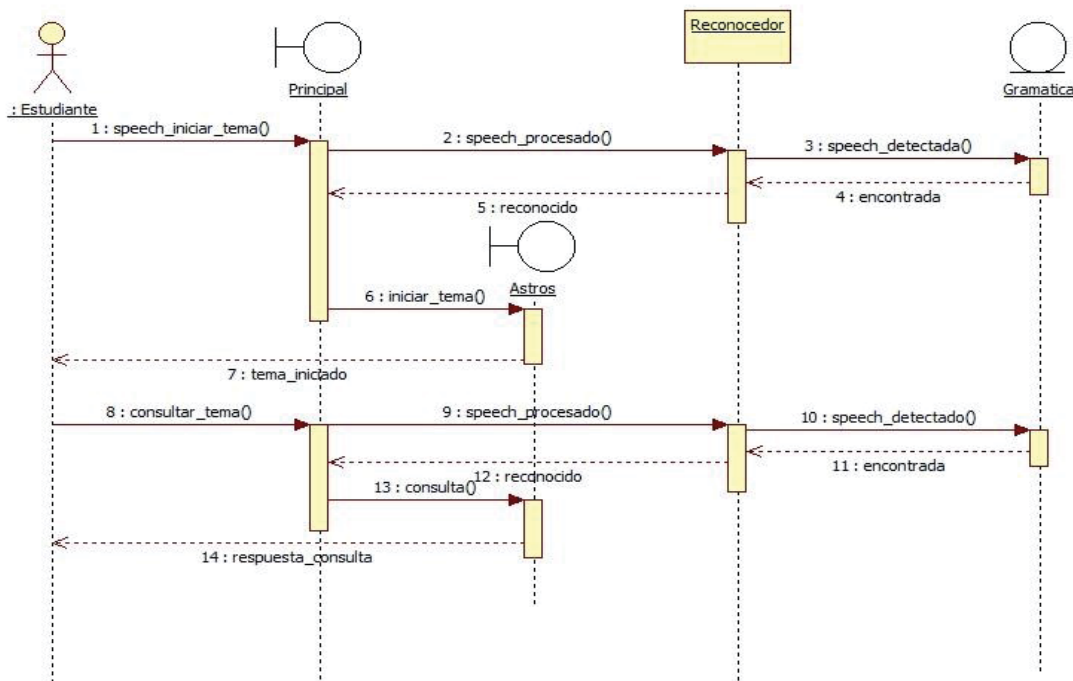


Figura 9: Diagrama de Secuencia Mapu - Voz

## 10. Segunda etapa del desarrollo del Proyecto: Desarrollo entorno chatbot

### 10.1 Integración chatbot

El objetivo de todo chatbot, categoría a la cual los Alicebot pertenecen, y tal como fue descrito en capítulos anteriores es el de mantener un diálogo con el usuario por el mayor tiempo posible, simulando un diálogo fluido, coherente y lo más humano posible.

Es por eso que se implementará en *Aprendiendo con MAPU* un Alicebot, para dar al usuario una nueva herramienta para el aprendizaje de la lengua Mapudungun, permitiéndole interactuar con el sistema no solamente a través de eventos de mouse, como lo hacía en un principio, o a través de su voz como se realizó en la primera etapa de este proyecto, sino que además se le brinda la posibilidad de entablar un diálogo, con la intención de que este diálogo sea lo más “humano” posible lo que le ayudará a un mejor aprendizaje, y mayor interés por el uso del sistema.

El funcionamiento de este chatbot será a través de voz o simplemente del teclado, eso quedará a elección de usuario. Su implementación es posible gracias a *Program# - AIMLbot.dll*, desarrollado por Nicholas H. Tollervy, el cual es una implementación de AIML para .NET; funciona tanto en entornos de desarrollo Windows así también como el entorno libre Mono. (12)

Haciendo uso de los métodos provistos por *Program#* se puede acceder y tratar las reglas contenidas en los archivos de conocimiento AIML.

AIML cuenta con diferentes tags los cuales indican las diversas funcionalidades que ofrece este lenguaje de programación.

Los principales tags en el lenguaje AIML son:

- ❖ <aiml>: el tag aiml marca el comienzo y el fin de un documento AIML.
- ❖ <category>: el tag que indica una unidad de conocimiento, en la base de conocimiento de un Alicebot.
- ❖ <pattern>: contiene un simple patrón el cual se compara a un patrón que el usuario haya dicho o escrito a un Alicebot.
- ❖ <template>: contiene la respuesta a un patrón de entrada.

Un ejemplo sencillo de base de conocimiento AIML es el siguiente:

```

<?xml version="1.0" encoding="UTF-8"?>
  <aiml version="1.0">
    <category>
      <pattern>Hola</pattern>
      <template>
        <random>
          <li>Hola</li>
          <li>Mari Mari</li>
          <li>Como estas</li>
        </random>
      </template>
    </category>
    <category>
      <pattern>QUIEN ERES</pattern>
      <template>Mi nombre es MAPU</template>
    </category>
  </aiml>

```

Cabe recordar que el botmaster es la persona que crea o es el autor de su propio chat robot, bot conversacional o simplemente bot. Él es el encargado de mantener y/o ampliar la base de conocimiento del bot conversacional.

La base de conocimiento, define el rango de conocimiento, o el cerebro del bot, en palabras simples, lo que el bot es capaz de reconocer.

Ya definidos, en palabras simples, lo que es un botmaster, y en que consiste la base de conocimiento, se puede decir que, una de las labores más importantes del botmaster es la de revisar los registros de las conversaciones realizadas con el bot. El porqué de esta importancia radica en que solamente a través del análisis de los registro de las conversaciones el botmaster logrará conocer qué tipo de conversación es la que los usuarios están solicitando al bot. Conociendo esta información el botmaster es capaz de modificar o ampliar la base de conocimiento, para así lograr una mayor satisfacción para los usuarios.

## 10.2 Interfaz

La interfaz, consiste de un *form* desarrollado en Visual Basic 2010, con cinco áreas principales:

- ❖ Área de conversación: En esta área es donde se van registrando las conversaciones realizadas en la sesión actual, consiste de un *textbox* con *scroll* automático que se posiciona en la última frase escrita. Consiste de dos interlocutores: Tú que representa al usuario, donde se muestra lo que el usuario ha escrito. Y Mapu: que representa al bot, y muestra las respuestas de éste a las consultas del usuario.
- ❖ Área de escritura: En este lugar es donde el usuario ingresa las consultas o conversaciones que luego serán respondidas por el bot.

- ❖ Área imagen referencial: En este sector se muestra una imagen relacionada al tema de conversación que se está tratando.
- ❖ Activación/Desactivación Reconocimiento de Voz: En este lugar, que se encuentra inserto en el formulario principal, se permitirá al usuario, activar y/o desactivar las funcionalidades del reconocedor de voz.
- ❖ Labels: Los labels aquí encontrados, indican, cuando el reconocimiento de voz se encuentra activo, que palabra y por ende que pronunciación corresponde a cada ítem.

## 10.2.1 Imagen

### 10.2.1.1 Descripción

Uno de los puntos importantes en el ámbito de la enseñanza de idiomas es que el niño sea capaz más que de asociar una palabra del idioma nativo con otra del idioma objetivo, dar significado a las palabras. Para poder contribuir con esto, se han asociado imágenes a ciertos temas de uso común en la lengua Mapudungun.

### 10.2.1.2 Desarrollo

Para lograr esto, debido a que el intérprete utilizado (AIMLBot) no brinda la oportunidad de manejo de variables desde AIML a Visual Basic, se ha hecho uso de una de las opciones que si trae, la cual es la clase *SubQuery*, esta clase devuelve un objeto con los principales tags utilizados en el archivo AIML - no olvidar que los archivos AIML son los que proporcionan la base de conocimiento al bot – estos tags son: FullPath, Template, ThatStar y TopicStar.

Haciendo uso de *SubQuery.Template*, el cual devuelve una cadena de texto con los tags *template* más todo lo que está contenido entre ellos, junto con el tag *think*, el cual es un tag AIML que hace que lo que esté entre ellos no sea mostrado al usuario sino que solamente queda para ser usado por el bot.

```
<template>
<think>saludo</think>
  <random>
    <li>Hola</li>
    <li>Buen día</li>
  </random>
</template>
```

Tomando como ejemplo el extracto de código anterior obtenido desde *SubQuery.Template*, el cual es una respuesta a la entrada 'hola', se ve en *template* dos tags: *think* y *random*. *Random* da la respuesta a la entrada y *think* contiene un nombre que representa el tema de conversación.

Obteniendo aquella cadena de texto como resultado, ahora se procesa esta cadena para obtener un resultado más significativo. Para ello se hace uso de la función

```
Mid(SubQuery.template, 18, 3)
```

Lo que hace esta función es obtener los tres siguientes caracteres, desde el cierre del tag de apertura del think, en este caso la cadena 'sal', con este código de tres caracteres se procede ahora a obtener la imagen desde una lista de imágenes asociadas a un código único de tres caracteres, imagen que servirá para acompañar el tema de conversación.

## 10.2.2 Generador Texto

### 10.2.2.1 Descripción

Una de las cosas en que los computadores son realmente buenos, es en la velocidad de procesamiento, la velocidad de respuesta a una conversación. Es por ello y con el objetivo de mantener al bot en una apariencia lo más humana posible, se han utilizado funciones que modifican la velocidad de respuesta y la forma en como el texto es presentado.

### 10.2.2.2 Desarrollo

```
MyResult.bot.writeToLog("Tu: " & TextBox2.Text & vbNewLine & "Mapu:"  
                        & MyResult.Output)  
  
Thread.Sleep(300)  
TextBox1.AppendText("Tu: " & TextBox2.Text & vbNewLine & "Mapu: ")  
For i As Integer = 0 To (MyResult.Output.Length - 1)  
    TextBox1.AppendText(MyResult.Output.Substring(i, 1))  
    Thread.Sleep(50)  
Next  
TextBox1.AppendText(vbNewLine)
```

Arriba se presenta un trozo de código utilizado para lograr el efecto deseado, primeramente, luego de escribir en el log, se realiza una pausa de 300 milisegundos, y comienza a escribirse letra por letra, con una pausa de 50 milisegundos por letra, la salida obtenida desde *MyResult.Output*.

## 10.3 Registros

### 10.3.1.1 Descripción

La parte fundamental de todo esta etapa es la de lograr llevar un registro de las conversaciones realizadas con el bot para su posterior análisis.

Haciendo uso de funciones de manejo de archivos de texto propias del lenguaje de programación, más el uso de funciones de fecha y hora, se permite crear y guardar los registros de las conversaciones en formato de archivo de texto, para su posterior análisis por parte del botmaster.

### 10.3.1.2 Desarrollo

```
log = TextBox1.Text
_date = "User: " & MyUser.UserID.ToString & " - Fecha: " &
Date.Now.ToString
toLog = _date & vbNewLine & log & vbNewLine
file = My.Computer.FileSystem.OpenTextFileWriter("gameData\log.txt",
    True)
file.WriteLine(toLog)
file.Close()
```

Código perteneciente al procedimiento saveLog(), el cual es el encargado de generar los reportes de todas las conversaciones realizadas con el bot.

En primer lugar se genera un identificador para el usuario, correspondiente al nombre del bot, por defecto Mapu, y la fecha actual, incluyendo segundos. A estos datos se le agrega el contenido del textbox1, que es donde se realizan los diálogos con el bot.

Todo esto se almacena en la carpeta gameData, bajo el nombre log.txt.

## 10.4 Upload Registro

### 10.4.1.1 Descripción

Para que el botmaster pueda mejorar la base de conocimiento, se requiere el constante envío de información sobre los registros para su posterior análisis. Para ello, se ha creado una función que envía a través de comunicación ftp el último log de conversación, una vez enviado, procede a mover el archivo en el disco local a una carpeta llamada logs enviados, para no volver a enviar el mismo log al servidor. Para evitar el duplicado de nombres en el servidor, al archivo enviado se le vuelve a agregar una marca de tiempo, con la fecha y hora completa del momento en que fue enviado.

### 10.4.1.2 Desarrollo

Este proceso se logra en tres etapas:

Primero, al hacer click en el Menu Item correspondiente, se genera un nuevo nombre de archivo y se obtiene la ruta del archivo de log actual y se envía a la función encargada de subir aquel archivo al servidor.

```
nombreArchivo = DateTime.Now.ToString("yyyyMMdd-HH:mm:ss.fff - ") &
    "log.txt"
ruta = My.Computer.FileSystem.CurrentDirectory & "\gameData\log.txt"
EnviarArchivoFTP(ruta, nombreArchivo)
```

Segundo, el procedimiento *EnviarArchivoFTP*, recibe la ruta y el nombre del archivo a subir, realiza la configuración del servidor, e intenta subir el archivo, si es posible, pasa al paso tres, sino, levanta una excepción.

```

Dim clsRequest As System.Net.FtpWebRequest
    'Dim conexion As Net.Sockets.TcpClient
clsRequest =
DirectCast(System.Net.WebRequest.Create("ftp://xxxxxxxx.org/" &
    NombreArchivo), System.Net.FtpWebRequest)
    clsRequest.Proxy = Nothing '        clsRequest.Credentials =
New System.Net.NetworkCredential("xxxxx@xxxxxxxx.org", "xxxxxxxxx") '
    clsRequest.Method =
System.Net.WebRequestMethods.Ftp.UploadFile
Try
    Dim bFile() As Byte =
System.IO.File.ReadAllBytes(OrigenArchivo)
    Dim clsStream As System.IO.Stream = _
    clsRequest.GetRequestStream()
    clsStream.Write(bFile, 0, bFile.Length)
    clsStream.Close()
    clsStream.Dispose()
    mover_log() 'envío archivo log a carpeta con logs a
    anteriores.
Catch ex As Exception
    MsgBox(ex.Message & ". El Archivo no pudo ser enviado,
    intente en otro momento")
End Try

```

Etapa Tercera, aquí, una vez subido el archivo al servidor, se procede a mover el archivo original a la carpeta llamada *logs\_enviados*, todo esto con el propósito de no subir logs duplicados al servidor.

```

ruta_inicial = My.Computer.FileSystem.CurrentDirectory & "\gameData\"
ruta_final = My.Computer.FileSystem.CurrentDirectory & "
    \gameData\logs-enviados\"
nombre_archivo_inicial = "log.txt"
nombre_archivo_final = DateTime.Now.ToString("yyyyMMdd-HH:mm:ss.fff-")
    & nombre_archivo_inicial
MsgBox("cmd.exe /c move " & ruta_inicial & nombre_archivo_inicial & "
    " & ruta_final & nombre_archivo_final)
Shell("cmd.exe /c move " & ruta_inicial & nombre_archivo_inicial & "
    " & ruta_final & nombre_archivo_final)

```

Como se puede apreciar, para lograrlo se hace uso del comando *move*, perteneciente a la *consola* de Windows.

## 10.5 Diálogos

### 10.5.1.1 Descripción

Los diálogos, o la base de conocimiento que tendrá el bot conversacional se dividirá en dos puntos, y en dos lenguajes distintos.

El primero de ellos, será realizado en español y contendrá información sobre la cultura Mapuche y el estilo de vida del pueblo Mapuche.

El siguiente será desarrollado en Mapudungun, y será un medio para practicar el idioma, y contendrá en su mayoría los temas tratados en aprendiendo con MAPU.

### 10.5.1.2 Desarrollo

El proceso aquí utilizado es el mismo que se utiliza en el Generador de Texto.

Lo importante a destacar aquí es cómo se obtienen los resultados de las consultas

```
Dim MyRequest As New Request (TextBox2.Text, MyUser, MyBot)
Dim MyResult = MyBot.Chat (MyRequest)
```

En este extracto de código, se establece la consulta, como un Nuevo *Request*, clase perteneciente a AIMLBot. Este nuevo request consta de texto o consulta ingresada, el nombre de usuario y el nombre del bot.

La segunda parte, el resultado, es el patrón obtenido al comparar la consulta, o request con la base de conocimiento del bot.

## 10.6 Análisis Informes

### 10.6.1.1 Descripción

Los registros subidos al servidor ftp, estarán disponibles para el botmaster en una página web, protegida por contraseña, la cual le permitirá acceder a los registros cuando y donde lo desee.

### 10.6.1.2 Desarrollo

Para lograr esto simplemente se ha hecho uso de una característica que ofrecía el hosting que alberga Mapu, es la de proteger por contraseña un directorio en específico.

## 10.7 Update Base de Conocimiento

### 10.7.1.1 Descripción

El propósito de todo lo desarrollado hasta ahora, es el de poder ampliar la base de conocimiento, para ello es que se realiza el respaldo de las conversaciones en los registros, y luego estos son subidos al servidor ftp para su posterior análisis. De esta forma, con lo visto por el botmaster en los registros, es posible ampliar la base de conocimiento, a las reales



necesidades de los usuarios, o incluso, le permite informar al maestro, en el caso de que su uso se de en un aula, que temas debe reforzar, o en cuales los alumnos están más interesados.

Una vez ampliada la base de conocimiento, le será ofrecida al usuario, para reemplazar la antigua base, pudiendo ser actualizada, si consta con una conexión a internet, de forma automática, o bien de manera manual.

### **10.7.1.2 Desarrollo**

El proceso de actualización de la base de conocimiento, es un proceso largo, en la cual intervienen tres clases y consta de los siguientes pasos:

1. Comprobar Versiones: Localmente se almacena un archivo que posee la versión actual de la base de conocimiento, si esta versión difiere con la almacenada en el servidor, se procederá a descargar los archivos correspondientes.
  - a. Para lograr esta comparación se baja una copia del archivo que se encuentra en el servidor, y se analiza, por medio de funciones de manejo de archivos su contenido.
  - b. Antes de realizar cualquier descarga, se comprueba si se encuentra conectado a la red o no, si lo está se procede a descargar, si no, levanta una excepción.
2. Descargar Archivos: Si corresponde la descarga de archivos, se procede a realizarse. Esto se logra, estableciendo una conexión con el servidor FTP, para luego descargarlos.
3. Modificación versión local: Si la descarga ha sido exitosa, se procede a modificar el número de la versión actual del programa.
4. Este proceso se realiza de forma automática, cuando se lanza la aplicación, mientras se ejecuta la pantalla de inicio, o bien de forma manual, en el menú correspondiente.
5. El usuario tiene plena libertad de decidir si aplicar o no los cambios a la base de conocimiento. Esto se realiza a través de una consulta (MsgBox) donde él escoge si *Si o No*, respecto de descargar la actualización.

## **11. Pruebas del Reconocedor de Voz**

Una de las características del presente proyecto, es la del control por comandos a través del reconocimiento de voz.

Con la intención de asegurar la calidad y el correcto funcionamiento de este se realizarán una serie de pruebas las que permitirán tras el posterior análisis conocer el estado del reconocedor y saber si es necesario el entrenamiento de éste.

### **11.1 Desarrollo**

Las pruebas se dividen en dos etapas, la primera de ellas tiene por fin el conocer si es necesario el realizar un entrenamiento previo de la aplicación, para lograr un correcto funcionamiento del reconocedor de voz.

La siguiente etapa tiene por objetivo, ya una vez identificado si es o no necesario realizar el entrenamiento previo del reconocedor, el conocer si es necesario algún tipo de procesamiento de la señal de voz.

### **11.2 Entorno**

La primera prueba será realizada la semana del 23 al 28 de Septiembre, con alumnos de 1 a 4 básico del Colegio Adventista de La Calera. La prueba será realizada en las dependencias del establecimiento.

El equipo a utilizar será un notebook, con Sistema Operativo Windows 7, y unos auriculares con micrófono integrado o mejor conocidos como headset de marca Genius. El equipo contará con dos sesiones, una con el entrenamiento del reconocedor de voz por defecto de Windows realizado, y otra sin este entrenamiento.

### **11.3 Pruebas**

La primera prueba, tiene por objetivo conocer si es necesario realizar un entrenamiento previo de la aplicación. En ella se dará una lista de tareas, que consisten en manejar vía voz, es decir el control por comandos de la aplicación, en esta lista que se muestra a continuación, se le solicita al participante ir avanzando a través de la aplicación por alguno de los temas de ésta. Estos temas han sido seleccionados, al igual que los elementos que se toman al interior de cada tema debido a que mucho de ellos son similares en su pronunciación, en especial en las partes donde la señal generada es más débil que son el inicio y el término de cada palabra. Junto a seleccionar palabras con similitud, se ha dado la posibilidad de que los participantes aleatoriamente puedan seleccionar ciertos ítems y temas, para lograr una mayor variedad a la hora de analizar los resultados.

La prueba consiste de las siguientes tareas:

1. Ver introducción
2. Iniciar con Usuario prueba
3. Ir a Nivel I
4. Seleccionar la lección Astros
  - a. Oír las instrucciones
  - b. Ver cada uno de los ítems que contiene esta lección
  - c. Cerrar Lección para volver al Nivel I
5. Seleccionar lección Animales y cada uno de los siguientes ítems
  - a. Vaca
  - b. Oveja
  - c. Cabra
  - d. Llama
  - e. Ir a Siguiente Lección
  - f. Seleccionar 2 animales de aquella lección
  - g. Cerrar Lección para volver al Nivel I
6. Seleccionar Lección Cuerpo Humano y cada uno de los siguientes ítems
  - a. Ojo
  - b. Cabeza
  - c. Ombligo
  - d. Boca
  - e. Estomago
  - f. Una parte del cuerpo que no haya sido seleccionada
  - g. Cerrar Lección para volver al Nivel I
7. Seleccionar Lección Números y cada uno de los siguientes ítems
  - a. Uno
  - b. Ocho
  - c. Cuatro
  - d. Cinco
  - e. Un número de los que no han salido
  - f. Cerrar Lección para volver al Nivel 1
8. Seleccionar una lección de las que no se hayan seleccionado anteriormente
  - a. Escoger 3 ítems de aquella lección

## 11.4 Conclusiones Prueba

A raíz de la prueba realizada con estos usuarios se puede destacar que:

- ❖ El sistema de reconocimiento de voz funciona correctamente para equipos con una cantidad de entrenamiento mínimo, es decir realizando una fracción, generalmente las primeras 4 páginas del entrenamiento del Reconocedor de voz de Microsoft.
- ❖ Las mayores dificultades se dan en la forma en la cual se pronuncian las palabras, problema originado por la falta de lectura, sobretodo en edades tempranas, lo que provoca que no se pronuncien correctamente ciertas palabras, por ejemplo, la palabra estrella (4.a)
- ❖ Derivado de lo anterior, y por motivo de las características de las ondas sonoras, se producen cierta ambigüedad en algunas palabras similares en sus extremos, en especial el último tramo de la señal, al ser señales muy similares en esos extremos, el reconocedor tiende muchas veces a equivocar su elección, por ejemplo, Gato – Ratón (5.f)
- ❖ Los usuarios no tienen por qué sabe cómo se pronuncia alguna palabra, cuando ésta tiene más de un significado.

## 12. Conclusiones

Las oportunidades que ofrecen, tanto la tecnología de reconocimiento de voz, como de bot Conversacional, aún tienen mucho por explotar, y sus aplicaciones, sus usos seguirán en aumento.

Ya en el año 2009 lo decía Zig Serafin, Corporate Vice President Online Services Division en Microsoft Corporation

*"La voz es el nuevo 'touch'. Es la evolución natural de teclados y pantallas táctiles. Hoy en día, el habla se está convirtiendo en una parte esperada de nuestra experiencia diaria a través de una variedad de dispositivos."*

*Y al igual que las interfaces touch tuvieron su etapa de rechazo frente a alternativas más económicas y amigables, ahora, cada vez son más los dispositivos móviles y los computadores que incorporan este tipo de interfaz. La voz, será dentro de no muchos años el dispositivo de interfaz por excelencia.*

Más allá de las dificultades técnicas que el reconocimiento de voz ha tenido hasta ahora, su potencial sigue siendo enorme, un ejemplo de ello es el último avance en la tecnología de reconocimiento y síntesis de voz realizado por Microsoft, específicamente por su división *Research*, se trata de un reconocedor de voz y traducción instantánea, ¿qué quiere decir esto?, usando la misma voz del interlocutor, es posible traducir instantáneamente lo que se está diciendo al idioma objetivo, haciendo uso del *Speech to Speech*.

En lo que concierne a la realización del proyecto, se puede decir que en términos generales, cumplió las expectativas planteadas al comienzo de éste. Se logró implementar un reconocedor de voz, de fácil uso y con un porcentaje de acierto que permite una interacción fluida con el sistema. Además de eso, se logró la implementación de un bot conversacional, el cual ofrece nuevas formas de interacción con el sistema, con la clara intención de ser una ayuda real en el proceso de aprendizaje.

En relación a Mapu, la inquietud de seguir ampliándolo continuará, aprovechando las sugerencias obtenidas durante el desarrollo de este proyecto. Entre ellas se encuentran el desarrollo de una interfaz más dinámica y adaptable a las necesidades de las personas encargadas de enseñar este idioma.

En definitiva, el presente proyecto ha otorgado la oportunidad al autor, de acercarse a ambas tecnologías, las cuales aún tienen mucho por ofrecer, el futuro es amplio, y las oportunidades que presentan también lo son.

## 13. Bibliografía

1. **Armstrong, Bruce**, Mgr. del Grupo Speech Technologies. The Novell Applications Group. 1994.
2. **insularesdivergentes.org**. <http://insularesdivergentes.org/node/63>. [En línea]
3. **Cole R.A et. al.** *Survey of the state of the Art in Human Language Technology*. s.l. : Cambridge University press, 1997.
4. **Nejat, Ince.** *Digital Speech Procassing: Speech Coding, Synthesis and Recognition*. s.l. : Springet-Verlag, 1992.
5. **Microsoft**. <http://msdn.microsoft.com/en-us/library/gg145021>. [En línea]
6. **W3C**. <http://www.w3.org/TR/speech-grammar/>. [En línea] <http://www.w3.org/TR/speech-grammar/>.
7. **Wallace, R.** *The Elements of AIML Style*. s.l. : ALICE A.I Foundation, 2003.
8. **Turing, A.**, *Computing Machinery and Intelligence*. 236, 1950, Mind, Vol. 59, págs. 433 - 460.
9. **CIRG**. Cybernetic Intelligence Research (Group) - Loebner Prize. [En línea] 2008.
10. **Weizenbaum, J.** *ELIZA - A computer program for study of Natural Language Communication Between Man and Machine*. 1, 1966, Communication of ACM, Vol. 9, págs. 36 - 45.
11. **Johnson, C.** EETime.com - AI researchers think 'Rascals' can pass Turing test. [En línea] 2008.
12. **H.Tollervey, Nicholas**. <http://aimlbot.sourceforge.net/>. [En línea] 2006.
13. **CONADI**. *Ficha didáctica N° 1 Azümtaiñ, Metodología y Sugerencias*.
14. **Vygotsky, Lev S.** *Pensamiento y lenguaje*. Buenos Aires : Lautaro, 1964.

## Anexos

### A. Azümcheffi: Características Básicas

El Grafemario Azümcheffi es un documento técnico elaborado por tres organizaciones Mapuches, sancionado en asamblea de líderes el 22 y 23 de junio de 1996. Este Grafemario tiene 6 vocales y 20 consonantes normales y 2 especiales; en consecuencia tiene 28 grafemas para la escritura del Mapudugun, la que se describe a continuación (13):

- ❖ VOCALES: A - E - I - O - U - Ü
- ❖ CONSONANTES: CH - Z - F - Q - K - L - LH - LL - M - N - NH - Ñ - G - P - R

En Tabla A1 y Tabla A2 se verán más detalles de este Grafemario.

Tabla A1: Sonidos o formas del Mapudugun distintas al castellano

Grafemas	Sonidos más próximos
Ü	El sonido de la sexta vocal difiere de la Ü castellana, puesto que se asemeja a una “I” o “U” más corta, al punto que se escucha casi imperceptible.
Z	Sonido explosivo fricativo dental que se asemeja a la “D” de dedo o “C” de cero en castellano.
F	Sonido explosivo labial fricativo que se asemeja a la “F” de frío o “V” de vaso en castellano.
Q	Sonido explosivo sonoro velar, semejante a la “G” y “Gü” del castellano.
Lh	Se diferencia de la “L” porque es un sonido más alargado. Se realiza adelantando la lengua y poniéndola entre los dientes.
Nh	Se diferencia de la “N” porque es un sonido más alargado. Su pronunciación se produce adelantando la lengua y poniéndola entre los dientes.
G	Sonido velar oclusivo sonoro nasal, parecido a la combinación “NG” del castellano, un poco más nasal.
R	Sonido alveolar fricativo sonoro, más fuerte que la “R” y un poco más suave que la “RR”.
TX	Sonido explosivo cercano a la combinación “TSR” o al sonido “TR” en el habla informal espontánea del castellano chileno.
T’	Sonido fricativo palatal sordo, se utiliza para manifestar con más fuerza expresiones despectivas y afectivas.
SH	Sonido continuo silbante, que se utiliza también para matices de afectividad y desprecio.

Tabla A2: Escritura de palabras en Mapudungun utilizando Azümcheñi

Grafemas	Ejemplo	Significado
A	Achawall	Pollo
Z	Zuchetum	Descabezar (trigo)
F	Furi	Espalda
I	Inal	A orillas
L	Lolin	Hoyo
LL	Llaskün	Tristeza
N	Newen	Fuerza
Ñ	Ñuke	Madre
O	Ofül	Sorbo
R	Rewe	Símbolo o punto focal de la ceremonia Guillantun
T	Metawe	Cántaro
U	Upen	Olvidar
W	Wagülhen	Estrella
SH	Pit'i shomo	Niñita (afectivo)
CH	Chofü	Flojo
E	Eltun	Cementerio, lugar de los muertos
Q	Reqle	Siete
K	Kechu	Cinco
LH	Lhaku	Abuelo paterno
M	Malal	Corral, cerco
NH	Punh	Noche
G	Gülham	Consejo
P	Püran	Subir, montar
S	Sañwe	Chanco, cerdo
TX	Txewa	Perro
Ü	Ürkün	Cansarse
Y	Yafü	Duro, firme
T'	Weza t'e	Gente mala (despectivo)



## B. Gramática Mapu

```
<?xml version="1.0" encoding="utf-8"?>
<grammar xml:lang="es-ES"
tag-format="properties-ms/1.0" version="1.0"
xmlns="http://www.w3.org/2001/06/grammar"
xmlns:sapi="http://schemas.microsoft.com/Speech/2002/06/SRGSExtensions">
  <rule id="iniciarAction" scope="public">
    <one-of>
      <item> iniciar </item>
      <item> ocultar</item>
      <item> mostrar </item>
      <item> activar </item>
    </one-of>
  </rule>
  <rule id="login" scope="public">
    <one-of>
      <item>Introduccion</item>
      <item>Iniciar</item>
      <item>Crear</item>
      <item>Aceptar</item>
    </one-of>
  </rule>
  <rule id="indice" scope="public">
    <one-of>
      <item>Bienvenido</item>
      <item>Cerrar Sesion</item>
      <item>Nivel Uno</item>
      <item>Nivel Dos</item>
      <item>Nivel Tres</item>
      <item>Juego</item>
      <item>Derecha</item>
      <item>Izquierda</item>
      <item>Click</item>
      <item>Salir</item>
    </one-of>
  </rule>
  <rule id="nivel_uno" scope="public">
    <one-of>
      <item>Abecedario</item>
      <item>Animales</item>
      <item>Astros</item>
      <item>Cosas de Casa</item>
      <item>Colores</item>
      <item>Comidas</item>
      <item>Cuerpo Humano</item>
```

```

        <item>Estaciones del Año</item>
        <item>Familia</item>
        <item>Numeros</item>
        <item>Indice Principal</item>
    </one-of>
</rule>
<rule id="nivel_dos" scope="public">
    <one-of>
        <item>Frases</item>
        <item>Indice Principal</item>
    </one-of>
</rule>
    <rule id="nivel_tres" scope="public">
        <one-of>
            <item>Cuestionario</item>
            <item>Indice Principal</item>
        </one-of>
    </rule>
    <rule id="abecedario" scope="public">
        <one-of>
            <item>A</item>
            <item>Z</item>
            <item>UU</item>
            <item>M</item>
            <item>CH</item>
            <item>E</item>
            <item>F</item>
            <item>I</item>
            <item>K</item>
            <item>T</item>
            <item>NH</item>
            <item>TX</item>
            <item>O</item>
            <item>Y</item>
            <item>Q</item>
            <item>G</item>
            <item>LH</item>
            <item>Ñ</item>
            <item>R</item>
            <item>S</item>
            <item>LL</item>
            <item>P</item>
            <item>U</item>
            <item>W</item>
            <item>L</item>
            <item>N</item>
            <item>Instruccion</item>
            <item>Cerrar Lección</item>
        </one-of>
    </rule>

```

```

    </one-of>
</rule>
<rule id="animales" scope="public">
  <one-of>
    <item>Cerdo</item>
    <item>Vaca</item>
    <item>Oveja</item>
    <item>Cabra</item>
    <item>Buey</item>
    <item>Gallina</item>
    <item>Llama</item>
    <item>Perro</item>
    <item>Instrucción</item>
    <item>Siguiete Lección</item>
  </one-of>
</rule>
<rule id="animales2" scope="public">
  <one-of>
    <item>Gato</item>
    <item>Raton</item>
    <item>Gallo</item>
    <item>Pato</item>
    <item>Zorro</item>
    <item>Puma</item>
    <item>Culebra</item>
    <item>Caballo</item>
    <item>Practicar</item>
    <item>Cerrar Lección</item>
  </one-of>
</rule>
<rule id="astros" scope="public">
  <one-of>
    <item>Sol</item>
    <item>Estrellas</item>
    <item>Luna</item>
    <item>Practicar</item>
    <item>Cerrar Lección</item>
  </one-of>
</rule>
<rule id="cosas_de_casa" scope="public">
  <one-of>
    <item>Batea</item>
    <item>Cántaro</item>
    <item>Olla</item><item>Paila</item>
    <item>Cama</item>
    <item>Cuna</item>
    <item>Canasta</item>
    <item>Plato</item>
  </one-of>
</rule>

```

```

        <item>Instrucción</item>
        <item>Siguiete Lección</item>
    </one-of>
</rule>
<rule id="cosas_de_casa2" scope="public">
    <one-of>
        <item>Cuchillo</item>
        <item>Cucharón</item>
        <item>Frazada</item>
        <item>Piedra de moler</item>
        <item>Mesa</item>
        <item>Huso</item>
        <item>Practicar</item>
        <item>Cerrar Lección</item>
    </one-of>
</rule>
<rule id="colores" scope="public">
    <one-of>
        <item>Azul</item>
        <item>Verde</item>
        <item>Café</item>
        <item>Amarillo</item>
        <item>Naranja</item>
        <item>Violeta</item>
        <item>Verde Claro</item>
        <item>Morado</item>
        <item>Plomo</item>
        <item>Blanco</item>
        <item>Negro</item>
        <item>Rojo</item>
        <item>Instrucción</item>
        <item>Practicar</item>
        <item>Cerrar Lección</item>
    </one-of>
</rule>
<rule id="comidas" scope="public">
    <one-of>
        <item>Catutos</item>
        <item>Chicha</item>
        <item>Pan</item>
        <item>Mote</item>
        <item>Ñachin</item>
        <item>Muday</item>
        <item>Curanto</item>
        <item>Milcao</item>
        <item>Charquican</item>
        <item>Merken</item>
        <item>Harina Tostada</item>
    </one-of>
</rule>

```

```

        <item>Tortilla de rescoldo</item>
        <item>Instrucción</item>
        <item>Practicar</item>
        <item>Cerrar Lección</item>
    </one-of>
</rule>
<rule id="cuerpo_humano" scope="public">
    <one-of>
        <item>Cabeza</item>
        <item>Ojo Izquierdo</item>
        <item>Ojo Derecho</item>
        <item>Oreja</item>
        <item>Cuello</item>
        <item>Panza</item>
        <item>Ombligo</item>
        <item>Rodilla</item>
        <item>Pies</item>
        <item>Pera</item>
        <item>Pierna</item>
        <item>Mano</item>
        <item>Brazo</item>
        <item>Boca</item>
        <item>Nariz</item>
        <item>Instrucción</item>
        <item>Practicar</item>
        <item>Cerrar Lección</item>
    </one-of>
</rule>
<rule id="estaciones_del_año" scope="public">
    <one-of>
        <item>Verano</item>
        <item>Otoño</item>
        <item>Invierno</item>
        <item>Primavera</item>
        <item>Norte</item>
        <item>Sur</item>
        <item>Este</item>
        <item>Oeste</item>
        <item>Instrucción</item>
        <item>Practicar</item>
        <item>Cerrar Lección</item>
    </one-of></rule>
<rule id="familia" scope="public">
    <one-of>
        <item>Abuela Materna</item>
        <item>Abuelo Materno</item>
        <item>Tío</item>
        <item>Mamá</item>

```

```

        <item>Sobrino</item>
        <item>Hermano</item>
        <item>Yo</item>
        <item>Hermana</item>
        <item>Sobrina</item>
        <item>Papá</item>
        <item>Tía</item>
        <item>Abuela Paterna</item>
        <item>Abuelo Paterno</item>
        <item>Instrucción</item>
        <item>Practicar</item>
        <item>Tu Familia</item>
        <item>Cerrar Lección</item>
    </one-of></rule>
<rule id="numeros" scope="public">
    <one-of>
        <item>Uno</item>
        <item>Dos</item>
        <item>Tres</item>
        <item>Cuatro</item>
        <item>Cinco</item>
        <item>Seis</item>
        <item>Siete</item>
        <item>Ocho</item>
        <item>Nueve</item>
        <item>Diez</item>
        <item>Cien</item>
        <item>Mil</item>
        <item>Consultar</item>
        <item>Practicar dos</item>
        <item>Instrucción</item>
        <item>Practicar</item>
        <item>Cerrar Lección</item>
    </one-of> </rule>
<rule id="juego" scope="public">
    <one-of>
        <item>Derecha</item>
        <item>Izquierda</item>
        <item>Click</item>
        <item>Salir</item>
    </one-of></rule></grammar>

```