



TESINA DE LICENCIATURA

TITULO: Adaptación de software educativo para alumnos con deficiencia motriz, mediante comandos por voz.

AUTORES: Ostermann, Stefania - Moralejo, M. Lucrecia

DIRECTOR: Sanz, Cecilia

CODIRECTOR:

CARRERA: Licenciatura en sistemas

Resumen

En el presente trabajo, se realizó un desarrollo de investigación aplicada, en el que se propuso la adaptación de un software educativo mediante comandos por voz.

Durante este proceso se analizaron y probaron algunos programas educativos y software de adaptación. Además, se revisaron conceptos generales de Tecnologías de la Información y la Comunicación (TIC) vinculados a la educación, en particular al ámbito de la educación especial. Se explicaron las características que presentan las personas con deficiencia motriz y cuáles son sus necesidades, analizando algunas ayudas técnicas posibles.

Por otro lado, se estudiaron diferentes motores de reconocimiento de voz, y los fundamentos teóricos sobre esta temática. Se eligió y se abordó el análisis del motor de RV Sphinx-4. Se explicó la arquitectura y principales características.

Antes de realizar y presentar el prototipo, se hizo un estudio de la arquitectura de diseño y desarrollo de la herramienta educativa elegida, JClic, interiorizándose en sus componentes, clases y cuestiones específicas del lenguaje JAVA.

Finalmente, se obtuvo el desarrollo de un prototipo de adaptación del programa JClic, con la integración Sphinx-4 para proveer reconocimiento de voz, en particular, para las actividades de asociación simple. Se realizó una propuesta de extensión del prototipo para el resto de las actividades y una serie de pautas para la integración de un motor de RV con una aplicación (en forma genérica).

Para concluir, se sometió el prototipo a juicio de expertos y se detallan las conclusiones obtenidas.

Líneas de Investigación

Algunas de las líneas de investigación abordadas son:

- ✓Tecnologías de la Información y la Comunicación aplicadas a la educación, en particular educación especial
- ✓Software educativos y de adaptación
 - Estudio y pruebas con diferentes software
 - Análisis comparativo
- ✓Discapacidad motora
 - Causas y clasificación
 - Necesidades particulares en el área de TIC
 - Ayudas técnicas
- ✓Reconocimiento de voz
 - Fundamentos teóricos
 - Estudio y pruebas con diferentes motores
 - Arquitectura y funcionamiento de Sphinx-4

Trabajos Realizados

Se comenzó el trabajo, analizando algunas herramientas de autor y programas educativos, abarcando aspectos funcionales y posibilidades de uso. Luego, se realizó un estudio más detallado de la herramienta elegida para la integración, JClic. Se estudiaron, además, conceptos bases sobre reconocimiento de voz para entender su lógica y funcionamiento. Se analizaron diferentes motores de reconocimiento de voz y las conveniencias para la adaptación propuesta. Posteriormente, se eligió un motor de RV, Sphinx-4, y se analizó en detalle. Luego, se propuso una estrategia de adaptación para JClic, que permitiera integrar los comandos por voz en la resolución de actividades.

Finalmente, se sometió a evaluación el prototipo propuesto y su estrategia de solución, a través de una encuesta a algunos expertos vinculados a la temática. Se presentan las opiniones de los mismos.

Conclusiones

Para el presente trabajo se decidió someter el prototipo realizado a prueba de expertos. A partir de estas evaluaciones, se considera que se ha realizado una buena elección del software educativo a adaptar, como así también, la utilización de comandos por voz. Como mencionó uno de los expertos, esta adaptación puede usarse de forma complementaria con otras herramientas y no necesariamente es mejor o peor que otra adaptación, sino que es una alternativa diferente la cual abre otras posibilidades.

Los expertos coincidieron en que la herramienta de reconocimiento de voz elegida es adecuada. El aspecto fundamental a resaltar, es su disponibilidad y sus posibilidades en cuanto a funcionalidad.

Trabajos Futuros

Como líneas de trabajo futuras se plantean:

- ✓Realizar prueba con alumnos y docentes del área.
- ✓ Modificar JClicAuthor para la elección de utilización de los comandos por voz.
- ✓Permitir la configuración de las etiquetas.
- ✓Extender la implementación a todas las actividades disponibles en JClic.

Fecha de la presentación: Febrero de 2010

AGRADECIMIENTOS

Queremos agradecer, en primer lugar, a Cecilia Sanz por permitirnos realizar nuestra tesina de grado bajo su dirección, por enseñarnos conocimientos indispensables para el desarrollo del presente trabajo y por brindarnos su apoyo y confianza.

A Antonio Sacco, por ofrecernos información desinteresadamente y proveernos de bibliografías y conocimientos en el área de tecnología aplicada a la educación especial.

A las personas que colaboraron en las encuestas realizadas, aportando información que resultó de gran interés para evaluar el trabajo realizado.

Por último, pero no menos importante, queremos agradecer a nuestras familias, por guiarnos, por apoyarnos, por hacer posible que llevemos a cabo nuestros estudios y acompañarnos en todas nuestras decisiones.

A todos ellos, GRACIAS!

RESUMEN

En el presente trabajo se realizó un desarrollo de investigación aplicada, en el que se propuso la adaptación de un software educativo mediante comandos por voz.

Durante este proceso se estudiaron, analizaron y probaron algunos programas educativos y software de adaptación. Además, se revisaron conceptos generales de Tecnologías de la Información y la Comunicación (TIC) vinculados a la educación, en particular al ámbito de la educación especial. Se explicaron las características que presentan las personas con deficiencia motriz y cuáles son sus necesidades, analizando cuáles algunas posibles ayudas tecnológicas.

Por otro lado, se estudiaron diferentes motores de reconocimiento de voz, y los fundamentos teóricos sobre esta temática. Se eligió y se abordó el análisis del motor de RV Sphinx-4. Se explicó la arquitectura y principales características.

Antes de realizar y presentar el prototipo, se hizo un estudio de la arquitectura de diseño y desarrollo de la herramienta educativa elegida, JClic, interiorizándose en sus componentes, clases y cuestiones específicas del lenguaje JAVA.

Finalmente, se obtuvo el desarrollo de un prototipo de adaptación del programa JClic, con la integración Sphinx-4 para proveer reconocimiento de voz, en particular, para las actividades de asociación simple. Se realizó una propuesta de extensión del prototipo para el resto de las actividades y una serie de pautas para la integración de un motor de RV con una aplicación (en forma genérica).

Para concluir, se realizó una primera evaluación del prototipo a través de la opinión de expertos vinculados de algún modo a la temática. Se presentaron en este informe algunos de los resultados y conclusiones obtenidas.

ÍNDICE

Capítulo 1 - Introducción

<i>Resumen</i>	13
<i>Objetivos</i>	13
<i>Objetivos generales</i>	13
<i>Objetivos específicos</i>	14
<i>Motivación</i>	14
<i>Desarrollos propuestos para esta tesina</i>	19
<i>Estructura del trabajo</i>	20
<i>Resultados esperados</i>	21

Capítulo 2 - Tecnologías de la Información y Comunicación aplicadas a la educación

<i>Introducción</i>	23
<i>Tecnología de la Información y la comunicación aplicada a la educación</i>	23
<i>TIC y educación especial</i>	26
<i>Los Programas</i>	28
<i>Programas de propósito general</i>	28
<i>Programas con fines formativos</i>	28
<i>Resumen comparativo de algunos software con fines formativos</i>	38
<i>Conclusión</i>	39

Capítulo 3 - Accesibilidad y Adaptabilidad

<i>Introducción</i>	41
<i>Concepto de Accesibilidad y Adaptabilidad</i>	42
<i>Revisión de algunas normas de Accesibilidad</i>	43
<i>Algunas Normas de Accesibilidad de la Informática en España</i>	44
<i>Requisitos generales</i>	45
<i>Entorno Operativo</i>	48
<i>Aplicaciones</i>	50
<i>Autopistas de la Información</i>	51
<i>Accesibilidad en la Web</i>	52
<i>Pautas de Accesibilidad Web</i>	52
<i>Pautas de Accesibilidad al Contenido en la Web (WCAG)</i>	54
<i>Pautas de Accesibilidad para Herramientas de Autor (ATAG)</i>	54
<i>Pautas de Accesibilidad para Agentes de Usuario (UAAG)</i>	55
<i>Software de adaptación</i>	56
<i>Programas de Autoayudas Informáticas</i>	56
<i>MPB - Mouse por barrido</i>	57
<i>Emuclic</i>	58
<i>Kanghuroo</i>	58
<i>Ratón Facial</i>	59
<i>Algunas pruebas de integración con JClic</i>	60
<i>Programas exclusivos según la discapacidad</i>	63

<i>¿Qué se entiende por tecnologías de ayuda?</i>	66
<i>Conclusión</i>	66

Capítulo 4 - La particularidad de las personas con problemas motrices y Ayudas Técnicas

<i>Introducción</i>	69
<i>Discapacidad Motriz</i>	69
<i>Clasificación según el origen</i>	71
<i>Clasificación según su etiología</i>	71
<i>Clasificación según localización topográfica</i>	72
<i>¿Cómo se detecta la discapacidad motora?</i>	72
<i>Consecuencias de la discapacidad motora en el desarrollo de la persona</i>	72
<i>Consecuencias en el desarrollo del lenguaje</i>	73
<i>Procesos motores básicos en la adquisición del habla</i>	74
<i>Consecuencias en el desarrollo cognitivo</i>	76
<i>Consecuencias en el desarrollo social</i>	77
<i>Ayudas técnicas para personas con movilidad limitada</i>	78
<i>Ayudas técnicas para acceder al teclado</i>	78
<i>Varillas</i>	79
<i>Licornio</i>	79
<i>Varilla Bucal</i>	80
<i>Carcasas</i>	80
<i>Sujeta teclas</i>	80
<i>Soportes</i>	81
<i>Conmutadores</i>	81
<i>Escritura e interacción</i>	86
<i>Adaptar el teclado</i>	86
<i>El joystick sustituye al ratón</i>	88
<i>Ratón de bola</i>	89
<i>Ratón de cabeza</i>	89
<i>Emulador de ratón Julián</i>	89
<i>Emulador de ratón preciso para boca</i>	90
<i>Emulador de ratón Junior</i>	90
<i>Emulador de ratón preciso para mano</i>	91
<i>Reconocimiento de voz</i>	91
<i>Mover la cabeza, dirigir la mirada</i>	91
<i>Conclusión</i>	92

Capítulo 5 - Reconocimiento de voz

<i>Introducción</i>	95
<i>Reseña histórica</i>	95
<i>Fundamentos del reconocimiento de voz</i>	96
<i>Dificultades presentadas en el reconocimiento de voz</i>	97
<i>Características de los sistemas de reconocimiento de voz</i>	98
<i>El proceso de reconocimiento</i>	100
<i>Estrategias a utilizar para el reconocimiento de voz</i>	101
<i>Obtención de la señal sonora</i>	102

<i>Obtención de muestras</i>	103
<i>Métodos de análisis de la señal vocal</i>	104
<i>Codificación digital de la señal</i>	104
<i>Modelo digital de voz</i>	105
<i>Análisis localizado</i>	107
<i>Análisis en el dominio del tiempo</i>	109
<i>Análisis en el dominio de la frecuencia</i>	109
<i>Análisis de predicción lineal</i>	112
<i>Procesamiento homomórfico</i>	113
<i>Reconocimiento de patrones de habla</i>	114
<i>Técnica de clasificación de patrones</i>	115
<i>Conclusión</i>	117

Capítulo 6 - Análisis de software para el reconocimiento de voz

<i>Introducción</i>	119
<i>Criterios de evaluación de productos de reconocimiento de voz</i>	119
<i>Análisis de diferentes herramientas de SR</i>	120
<i>Loquendo</i>	121
<i>XVoice</i>	123
<i>NICO Toolkit</i>	126
<i>CMU Sphinx</i>	127
<i>Dragon Naturally Speaking</i>	129
<i>Java Speech API</i>	130
<i>Objetivos de diseño para la Java Speech API</i>	131
<i>Características de la API</i>	131
<i>El reconocimiento de voz y las java APIs</i>	132
<i>Requerimientos</i>	132
<i>Conclusión</i>	133

Capítulo 7 - Sphinx - 4

<i>Introducción</i>	135
<i>Motivación</i>	135
<i>Sphinx -4</i>	135
<i>Características</i>	136
<i>Arquitectura de alto nivel</i>	137
<i>Entrenamiento y configuración</i>	153
<i>Aplicaciones que usan Sphinx</i>	154
<i>Conclusión</i>	155

Capítulo 8 - JClíc

<i>Introducción</i>	157
<i>Motivación</i>	157
<i>¿Qué es JClíc?</i>	158
<i>Versión y Objetivos</i>	158

<i>Características</i>	159
<i>Componentes</i>	159
<i>Compatibilidad y nuevas posibilidades</i>	160
<i>Arquitectura abierta</i>	160
<i>Formatos de datos</i>	161
<i>Internacionalización</i>	162
<i>Tipos de actividades</i>	162
<i>Obtener JClic</i>	164
<i>Estructura de clases</i>	165
<i>Conclusión</i>	167

Capítulo 9 - Modificación del software educativo JClic para la incorporación de comandos por voz

<i>Introducción</i>	169
<i>Propuesta de prototipo</i>	169
<i>Trabajos realizados previos al prototipo</i>	170
<i>Propuesta planteada para el prototipo – El caso de la asociación simple</i>	171
<i>Algunas decisiones previas y consideraciones de JClic</i>	172
<i>Elección de la etiquetas para identificar casilleros</i>	173
<i>Configuración y puesta a punto del reconocedor Sphinx-4</i>	174
<i>Integración del reconocedor con JClic</i>	190
<i>Ejecución de una actividad usando el prototipo</i>	192
<i>Extensión del prototipo</i>	195
<i>Consideraciones generales para integrar reconocimiento de voz a una aplicación</i>	199
<i>Evaluación del prototipo desarrollado</i>	200
<i>Juicio de expertos</i>	201
<i>Resultados obtenidos en la encuesta</i>	204
<i>Conclusión</i>	207

Capítulo 10 - Conclusiones y líneas de investigación futuras

<i>Conclusiones de la tesina</i>	209
<i>Introducción</i>	209
<i>Experiencia adquirida</i>	209
<i>Líneas futuras</i>	211
<i>Mejoras al prototipo propuesto</i>	211
<i>Prueba con casos concretos</i>	212
<i>Modificar JClicAuthor para la elección de utilizar voz</i>	212
<i>Configurar etiquetas</i>	212
<i>Extender la implementación a todas las actividades</i>	213
<i>Como cierre de este trabajo</i>	213

Anexo A -Encuestas.....	214
Anexo B - Definiciones.....	219
Bibliografías.....	223

ÍNDICE DE FIGURAS

Capítulo 3

<i>Figura 3.1 – Pantalla de control de grabación</i>	59
<i>Figura 3.2 – Marca del ratón facial para indicar la sección donde se realizará el clic</i>	62
<i>Figura 3.3 – Ejemplo de tablero de símbolos</i>	64

Capítulo 4

<i>Figura 4.1 – Licornio</i>	79
<i>Figura 4.2 – Varilla bucal</i>	80
<i>Figura 4.3 – Carcasa para teclado</i>	80
<i>Figura 4.4 – Soporte para mouse</i>	81
<i>Figura 4.5 – Soporte para teclado</i>	81
<i>Figura 4.6 – Teclado para una sola mano</i>	87
<i>Figura 4.7 – Teclado Maltron</i>	87
<i>Figura 4.8 – Teclado reducido</i>	87
<i>Figura 4.9 – Teclado ampliado</i>	88
<i>Figura 4.10 – Teclado de conceptos</i>	88
<i>Figura 4.11 – Teclado virtual</i>	88
<i>Figura 4.12 – Ratón de bola</i>	89
<i>Figura 4.13 – Ratón de cabeza</i>	89
<i>Figura 4.14 – Emulador de ratón Julián</i>	89
<i>Figura 4.15 – Emulador de ratón preciso para boca</i>	90
<i>Figura 4.16 – Emulador de ratón junior</i>	90
<i>Figura 4.17 – Emulador de ratón preciso para mano</i>	91

Capítulo 5

<i>Figura 5.1 – Establecimiento de características espectrales</i>	101
<i>Figura 5.2 – Movimiento vibratorio periódico</i>	102
<i>Figura 5.3 – Obtención de muestras de la señal de la voz</i>	103
<i>Figura 5.4 – Señal sonora de la vocal 'a'</i>	105
<i>Figura 5.5 – Señal no sonora de la consonante 's'</i>	106
<i>Figura 5.6 – Ventana de Hamming</i>	108
<i>Figura 5.7 – Mecanismo de establecimiento de ventanas</i>	108
<i>Figura 5.8 – Ejemplo de descomposición de una señal compleja en sumatorias de señales simples</i>	111
<i>Figura 5.9 – Gráfico de espectro de la señal de la voz</i>	112
<i>Figura 5.10 – Operaciones del análisis Homomórfico</i>	113
<i>Figura 5.11 – Reconocimiento del habla empleando comparación de patrones</i>	114
<i>Figura 5.12 – Dos clases de patrones solapados</i>	115
<i>Figura 5.13 – Funcionamiento general del reconocimiento de patrones</i>	117

Capítulo 7

<i>Figura 7.1 – Arquitectura de Sphinx-4</i>	138
--	-----

<i>Figura 7.2 – Cadena de DataProcessors</i>	139
<i>Figura 7.3 – Ejemplo de archivo ConfigurationManager</i>	141
<i>Figura 7.4 – Definición de microphone en el configurationManager</i>	141
<i>Figura 7.5 – Definición de speechClassifier</i>	142
<i>Figura 7.6 – Definición de speechMarker</i>	143
<i>Figura 7.7 – Ejemplo de “stream” con solo una región de voz</i>	144
<i>Figura 7.8 - Ejemplo de “stream” con solo una región de voz filtrada</i>	144
<i>Figura 7.9 – Ejemplo de un “stream” con dos regiones de voz</i>	144
<i>Figura 7.10 – Ejemplo de un “stream” con dos regiones de voz después del filtrado cuando “mergeSpeechSegments” es verdadero</i>	145
<i>Figura 7.11 – Ejemplo de un “stream” con dos regiones de voz después del filtrado cuando “mergeSpeechSegments” es falso</i>	145
<i>Figura 7.12 – Ecuación CMN</i>	147
<i>Figura 7.13: Diseño de las características retornadas/devueltas</i>	148
<i>Figura 7.14 Cálculo/Cómputo del vector Delta y Doble Delta</i>	148
<i>Figura 7.15 - Ejemplo de SearchGraph</i>	151

Capítulo 8

<i>Figura 9.1a - Actividades JClic</i>	163
<i>Figura 9.1b - Actividades JClic</i>	164
<i>Figura 9.2 Estructura interna del paquete “src”</i>	165
<i>Figura 9.3 Estructura interna del paquete “edu.xtec”</i>	166

Capítulo 9

<i>Figura 9.1 – Actividad de Asociación Simple</i>	171
<i>Figura 9.2 – Diccionario creado con el modelo WSJ</i>	177
<i>Figura 9.3 – Listado de fonemas de DIMEx30-T22</i>	179
<i>Figura 9.4 - Diccionario de palabras del prototipo</i>	179
<i>Figura 9.5 - Gramática del prototipo</i>	181
<i>Figura 9.6 - Archivo de configuración</i>	182
<i>Figura 9.7 – Estructura de carpetas generada por SphinxTrain</i>	183
<i>Figura 9.8 – Ejemplo de diccionario generado para el entrenamiento</i>	185
<i>Figura 9.9 – Ejemplo de diccionario de silencios generado para el entrenamiento</i> ...185	
<i>Figura 9.10 – Ejemplo de lista de fonemas generados para el entrenamiento</i>	186
<i>Figura 9.11 – Formato del archivo .fileids</i>	186
<i>Figura 9.12 – Formato del archivo .transcription</i>	187
<i>Figura 9.13 – Código donde se instancia el reconocedor</i>	190
<i>Figura 9.14 – Declaración de la clase VoiceRecognizer</i>	191
<i>Figura 9.15 – Pantalla inicial de JClic</i>	193
<i>Figura 9.16 – Cartel de inicio de proyecto</i>	193
<i>Figura 9.17 – Actividad de asociación simple de ejemplo</i>	194
<i>Figura 9.18 – Cartel de confirmación</i>	194
<i>Figura 9.19 – Actividad con una jugada resuelta</i>	194
<i>Figura 9.20 – Ejemplo de un juego de memoria</i>	196
<i>Figura 9.21 – Ejemplo de actividad para identificar palabras</i>	197
<i>Figura 9.22 – Ejemplo de puzzle doble</i>	197
<i>Figura 9.23 – Ejemplo de puzzle de intercambio</i>	198
<i>Figura 9.24 – Ejemplo de puzzle de agujero</i>	198

ÍNDICE DE TABLAS

Capítulo 2

Tabla 2.1 - Cuadro Comparativo de Programas con Fines formativos.....38

Capítulo 4

Tabla 5.1 – Clasificación de los fonemas consonánticos de acuerdo con el punto y modo de articulación y vocalización76

Tabla 5.2 – Clasificación de los conmutadores de acuerdo a la acción83

Tabla 5.3 – Clasificación de los conmutadores de acuerdo a la parte del cuerpo utilizada85

CAPÍTULO 1

INTRODUCCIÓN

RESUMEN

En el siguiente trabajo se hará un desarrollo de investigación aplicada, el cual se basa en proponer un prototipo de adaptación de un software educativo mediante comandos por voz. Durante el desarrollo se estudiarán y analizarán algunos programas educativos, teniendo en cuenta diferentes parámetros tales como la funcionalidad, las adaptaciones que este provee, la licencia para adquirirlos, entre otras. La idea de este análisis es obtener un panorama general de los productos de software educativo disponibles en el mercado en relación con sus adaptaciones y posibilidades en las interfaces para educación especial (en particular, para niños con dificultades motoras). Además se incluirá en el trabajo, conceptos y definiciones a tener en cuenta a la hora de realizar adaptaciones a un software o desarrollarlos considerando la diversidad de alumnos. Esto incluye conceptos tales como, adaptabilidad, ayudas técnicas, software para la utilización de estas ayudas técnicas, entre otras cosas. Estos conceptos permitirán obtener información acerca de cuáles son las adaptaciones más utilizadas y qué aspectos le facilitan al alumno con necesidades especiales.

Dentro de las adaptaciones están los llamados “comandos por voz”, estos utilizan reconocimientos de voz para dar órdenes a la computadora y poder manipular los programas.

Posteriormente se evaluarán varias herramientas que podrían ser utilizadas para implementar reconocimiento de voz.

Por último se propondrá un prototipo de adaptación de un software educativo, utilizando algún programa de reconocimiento de voz previamente estudiado.

OBJETIVOS

OBJETIVOS GENERALES

Como objetivos generales del presente trabajo, se pueden nombrar:

- Estudiar posibilidades de adaptación de las interfaces de software educativo para posibilitar su utilización por parte de alumnos con dificultades motoras. Analizar ejemplos y casos.
- Analizar y recopilar algunos softwares (libres y comerciales) de reconocimiento de voz (RV) disponibles en el mercado. Realizar un análisis comparativo de éstos desde el punto de vista de su funcionalidad.

- Realizar pruebas con algunos softwares de RV para trabajar con comandos por voz.

OBJETIVOS ESPECÍFICOS

Para llevar a cabo dichos objetivos, es necesario proponer algunos fines específicos como los que se nombran a continuación:

- Analizar un caso específico que será el del software JCLIC y su variedad de actividades para los alumnos.
- Plantear posibles modificaciones al software JCLIC para presentar una interfaz más flexible para personas con discapacidades motoras, en particular, a partir de la incorporación de comandos por voz.

MOTIVACIÓN

La utilización de Tecnologías de la Información y la Comunicación (TIC) está propiciando cambios importantes, de diversa índole y sentido, en la enseñanza escolar. Además, para muchas personas la utilización de herramientas informáticas constituyen, en este momento, el único camino posible para llevar a cabo actividades tan básicas para su crecimiento personal como: expresarse, comunicarse, trabajar o aprender. Es decir, para elaborar significados más allá de lo que sus condiciones físicas les han venido posibilitando.

Existen, hoy en día, diversos programas educativos considerados recursos informáticos diseñados con la intención de ser utilizados en el contexto del proceso de enseñanza y aprendizaje. Estos programas se caracterizan por ser altamente interactivos, a partir del empleo de recursos multimedia, como videos, sonidos, fotografías, diccionarios especializados, explicaciones, ejercicios y juegos instructivos que apoyan las funciones de evaluación y diagnóstico.

Como ejemplo de dichos software, podemos mencionar algunos de ellos:

- **JClíc** está formado por un conjunto de aplicaciones informáticas que sirven para realizar diversos tipos de actividades educativas. Es un entorno para la creación, realización y evaluación de actividades educativas multimedia desarrollado en la plataforma Java. Es una aplicación de software libre, bajo los términos de la Licencia Pública General de GNU (GPL)¹, que funciona en

¹ Ver definición en Anexo B.

diversos entornos operativos: Linux, Mac OSX, Windows y Solaris. Este proyecto fue desarrollado por el Departamento de educación de la Generalitat de Cataluña (**JClic, sitio oficial – 2010**).

- **Hot Potatoes** es un conjunto de seis herramientas de autor que permiten elaborar ejercicios interactivos basados en páginas Web de seis tipos básicos. La interactividad de los ejercicios se consigue mediante JavaScript. Las herramientas de autor permiten crear ejercicios en cualquier lengua basada en el conjunto de caracteres del alfabeto romano, incluyendo español, francés, alemán entre otras (**Hot Potatoes – 2010**).

Aunque los ejercicios se elaboran utilizando JavaScript, no se necesita saber nada sobre este lenguaje para poder utilizar estas aplicaciones. Todo lo que se necesita es introducir datos, textos, preguntas, respuestas, etc. y el programa se encarga de generar las páginas Web. Posteriormente se pueden publicar dichas páginas en un servidor Web.

El conjunto de aplicaciones de Hot Potatoes es de libre distribución para actividades educativas no comerciales.

- **TexToys** es una suite de dos programas de autor, WebRhubarb y WebSequitur. Los programas son usados para crear páginas web (documentos HTML), las cuales contienen ejercicios interactivos de aprendizaje. Los ejercicios están basados sobre los programas Rhubarb y Sequitur originalmente desarrollados por John and Muriel Higgins. Una vez que se hayan creado los ejercicios, se puede presentar a los estudiantes como una página web. También se pueden subir a un servidor.

A pesar de que contiene dos programas de autor diferente, es descargado e instalado como un simple paquete.

Es una aplicación shareware² que se puede descargar e instalar gratuitamente. Sin embargo, la versión de descarga es limitada, para obtener la versión completa que incluye el hosting que se mencionó anteriormente, se debe registrar el paquete utilizando el formulario de pedido (**TextToys – 2010**).

- Otra herramienta interesante es **Markin**. Las nuevas tecnologías requieren nuevas herramientas, y los docentes que trabajan "on-line", necesitan facilidades que no están disponibles en un navegador web estándar o en un procesador de textos. Muchas veces los estudiantes envían trabajos realizados en algún procesador de textos por correo electrónico, y los docentes deben

² Ver definición en Anexo B.

realizar marcas de corrección y devolver estos documentos con la mayor rapidez y facilidad. Este software facilita este circuito de interacción ofreciendo una herramienta que lo simplifique o sustituya.

Es un programa Windows que se ejecuta sobre la computadora del profesor. Se puede importar el texto de un estudiante seleccionando el texto y copiándolo al portapapeles, o directamente desde un RTF (Rich Text Format) o un archivo de texto.

Una vez que el texto ha sido importado, este software proporciona todas las herramientas que un docente necesita para marcar y anotar sobre el documento a corregir.

Cuando se completa el marcado, se puede exportar este texto como un archivo RTF para pasarlo a un procesador de textos, o como una página web para que los alumnos puedan ver las correcciones en un navegador web. Además, también podría ser enviado directamente al estudiante dentro del program **(Markin – 2010)**.

Como se puede apreciar, y se verá en el desarrollo de este trabajo, existe una gran cantidad de software orientado a la educación en sus distintos niveles. Aquí sólo se mencionan algunos y este tema se profundizará en los capítulos 2 y 3. Muchos de ellos han sido adaptados o ya fueron creados teniendo en cuenta la diversidad de alumnos, pero otros son sólo herramientas estándares que no brindan adaptación alguna, por lo que están destinados a un conjunto restringido de alumnos.

Para las personas con necesidades especiales, la mera utilización de las TIC puede representar la consecución de un elevado grado de autonomía en su vida personal.

La contribución de las TIC al desarrollo cognitivo en el ámbito de la educación especial no sólo estriba en sus funciones objetivas (calcular, almacenar, ejecutar) sino en la calidad de las interacciones educativas propiciadas por su utilización **(Sancho - 1998)**.

Una de las razones de la escasa implantación de las TIC en la educación especial es la diversidad y la especificidad de las necesidades. Lo que sucede, en general en el mundo del software es que cuanto más estándar sean las aplicaciones, mayor la cantidad de personas que vayan a utilizarlas o más grande el prestigio o prioridad del ámbito donde se apliquen, las inversiones serán más altas y la implantación más rápida y eficaz. La utilización de las TIC como herramientas, sobre todo en el campo de la educación especial, requiere desarrollos muy complejos y variados, algunos personalizados, que además van a ser utilizados por colectivos poco numerosos y con escasa consideración social y económica.

Las personas que están afectadas de algún tipo de disminución motriz se encuentran con numerosos obstáculos y barreras que les impiden el desarrollo de habilidades, la ejecución de actividades, la relación con las personas y el entorno, etc. La situación actual presenta grandes retos a superar para que una persona con disminución pueda estar en una posición de igualdad de condiciones respecto de las demás personas.

El comportamiento humano se caracteriza por una continua utilización de habilidades que podemos denominar básicas, y que tienen que ver con la percepción (visual, auditiva y táctil), la comunicación, el desplazamiento y la manipulación. Las personas que padecen algún tipo de disminución motriz ven afectadas una o varias de estas áreas indispensables, viendo significativamente mermadas sus posibilidades de interacción con el entorno y, en consecuencia, de relación e integración social.

Actualmente vivimos en una sociedad fundamentalmente pensada para personas que no presenten discapacidad alguna. El ser humano se distingue de los animales, entre otros aspectos, por su capacidad de modificar y alterar el entorno para adecuarlo a sus necesidades. Desde esta perspectiva y volviendo a la situación de las personas con disminución, no hay que hacer nada diferente de lo que tradicionalmente se viene haciendo desde el principio de los tiempos, que es adaptar adecuadamente el entorno y utilizar adaptaciones y ayudas técnicas que permitan eliminar el máximo de barreras que dificultan que una persona con discapacidad pueda interaccionar en él como una persona más. Y sin embargo, éste ha sido el punto principal de conflicto: las personas con disminución motriz no disponen con frecuencia de las ayudas técnicas y adaptaciones necesarias para interaccionar en un entorno hostil.

Las personas con disminución motriz elevada tienen poca autonomía para la realización de tareas, necesitando de una forma constante la llamada ayuda de terceros, para cambiar de tareas, acceder a información, desplazarse, etc. En la computadora este problema también se manifiesta, pues suele ser el profesional quien ayuda constantemente a la persona a realizar las tareas.

Un alto nivel de autonomía los haría sentir independientes, podría aumentar su autoestima y su autosuficiencia, y en consecuencia, hacerlos sentir útiles y capaces. También influirá considerablemente en su comportamiento, facilitando que puedan ser emprendedores y que puedan realizar actividades nuevas por iniciativa propia. Podrá aumentar el nivel de interacción con el entorno, lo que a su vez tendrá efectos en el mismo entorno que aumentará también las situaciones de interacción, en forma de reacción en cadena.

Los niños con severos trastornos motrices suelen tener, además de importantes dificultades de movilidad y manejo de su entorno, problemas de comunicación. Un niño que no pueda manejar sus manos, probablemente presente dificultades para acceder a la computadora y, por lo tanto, tenga dificultades para aprender muchísimas

cosas a las cuales estaría en condiciones de acceder teniendo en cuenta su capacidad intelectual. Algunos de los aspectos que se pueden abordar utilizando programas de computación son: el enfoque de la atención, el estímulo visual, la disminución de la sensación de aislamiento, el entretenimiento, el control del ambiente, entre otras **(Castellano, Sacco, Zurueta – 2009)**.

Las computadoras poseen varias características y capacidades que pueden ser muy bien aprovechadas para trabajar con personas con necesidades especiales. Entre otras cosas permiten utilizar una gran variedad de recursos como imagen y sonido, cubriendo múltiples necesidades con programas que pueden ser modificados para adaptarse a las necesidades de cada persona.

Por lo tanto, estas razones motivan el desarrollo de adaptaciones a un software muy utilizado en el ámbito educativo como es JClick, para facilitar su uso por parte de alumnos con dificultades motoras, y así incentivar su desarrollo intelectual.

Existen ciertas directrices, que vale la pena nombrar, y que se deben tener en cuenta al momento de desarrollar aplicaciones de cualquier tipo. Estas características se basan en el concepto de accesibilidad y usabilidad. La clave de la accesibilidad consiste en conseguir que la información se presente independientemente de los equipos y aplicaciones que se empleen para navegarla.

Algunos de los principios a tener en cuenta en el momento de diseñar una aplicación, también llamados "Principios del diseño para todos", son los siguientes³

1. Uso equitativo: los productos deben ser útiles y rentables para cualquier tipo de usuarios.
2. Flexibilidad en el uso: el diseño debe acomodarse a una amplia gama de preferencias y habilidades individuales. Esta flexibilidad supone que el producto puede ser usado directamente, sin ayudas técnicas o modificaciones, por personas con distintos tipos de habilidades y en las más diversas circunstancias.
3. Uso fácil e intuitivo: el producto debe ser fácil de usar y comprender independientemente del grado de experiencia, conocimiento y formación del usuario.

³ Apoyos digitales para repensar la educación especial. Navarro Sierra, Juana María Sancho Gil, John Woodward. Editorial Octaedro, S.L. (2001)

4. Información percibida: el producto debe ofrecer información pertinente y eficaz para el usuario. Un producto que no proporcione claramente información sobre la función para la que se creó, posee un mal diseño.
5. Tolerancia con el error: erradicar el error es una utopía, pero diseñar sistemas tolerantes al error y que permitan una fácil reconducción del mismo es algo posible y totalmente necesario.
6. Esfuerzo físico: la utilización del sistema requerirá del menor esfuerzo físico posible.
7. Compatibilidad: los sistemas deben ser compatibles con productos y ayudas técnicas destinadas a aquellos que no pueden acceder al uso de prestaciones de forma directa y eficiente.

El diseño universal o para todos propone no crear entornos o productos específicos para personas con discapacidad, sino que cuando se fabrique cualquier producto se tengan en cuenta las necesidades o intereses de todos los posibles usuarios. Esa iniciativa enfatiza al individuo, facilita su integración y busca la igualdad y no clasificar a los productos e individuos por deficiencias.

El diseño para todos se debería aplicar a las diversas áreas de la vida: hogar, educación, trabajo, ocio, transporte, etc. Sólo así, se podrán eliminar las barreras que impiden a los ciudadanos, independientemente de su edad, constitución, capacidad, limitaciones físicas, etc., desarrollarse como personas autónomas.

Si se diseñan los sistemas, servicios y productos teniendo en cuenta los niveles más altos de dificultad y en situaciones adversas, se están haciendo diseños aptos para cualquier persona. La usabilidad, o facilidad de uso, es un factor primordial para el éxito de un producto. Generalmente los artefactos o sistemas diseñados para personas con necesidades especiales con un buen nivel de usabilidad, son adecuados para todos los consumidores, para quienes la facilidad de uso es un elemento clave.

DESARROLLOS PROPUESTOS PARA ESTA TESINA

En principio, se comenzará con el estudio de las diferentes herramientas de software de uso educativo, analizando sus funcionalidades, alcance y otras características relacionadas con la adaptación a la diversidad de alumnos.

Luego, se realizará un análisis y estudio de algunas tecnologías de reconocimiento de voz que existen actualmente y se evaluarán las ventajas y desventajas de cada una, así

como también sus bondades y dificultades para integrarlas a alguna aplicación educativa.

Posteriormente, se propondrá un prototipo de adaptación, mediante comandos por voz, a uno de los softwares educativos analizados anteriormente. Esta tarea incluye los siguientes puntos:

En primer lugar, se hará un estudio exhaustivo de las diferentes funcionalidades presentadas por el software con el que se realizará el prototipo (JClic), la forma de generar actividades, así también como la interfaz presentada al alumno para resolverlas.

Luego, se analizará la manera en que se puede modificar la interfaz ya implementada por JClic con el objetivo de permitir la resolución de actividades mediante comandos por voz.

Como último paso, se analizará el caso concreto de integrar un software elegido de RV a la aplicación JClic.

ESTRUCTURA DEL TRABAJO

En el siguiente capítulo, se presentará una introducción de la importancia actual de la utilización de Tecnologías de la información y Comunicación en el ámbito educativo y se detallará una posible clasificación de software educativo. Otro objetivo de este capítulo es dar a conocer algunos de estos softwares, teniendo en cuenta las funcionalidades que ofrecen, y las posibilidades de adaptación que proveen, en particular, para educación especial. Por último, en este mismo capítulo, se hará un análisis comparativo de todas las herramientas presentadas.

En el capítulo 3 se definirán los conceptos de adaptabilidad y accesibilidad, se presentará el concepto de ayuda técnica, así como también diferentes softwares que hacen posible el uso de éstas.

En el capítulo 4, se verán las particularidades que presentan las personas con discapacidad motora, ya que a ellas está destinado el prototipo de adaptación. Además se hará hincapié en la importancia de la utilización de ciertas palabras en el uso de comandos por voz, basando nuestra conclusión en la teoría de adquisición de fonemas de Jakobson que se estudiará en el mismo capítulo.

En el capítulo 5, se hará una breve introducción a los fundamentos del reconocimiento de voz, de manera de ofrecer un marco conceptual para la tesis. En el capítulo 6, se incluirá un estudio de algunos programas de reconocimiento de voz, los cuales podrían

ser utilizados en el prototipo propuesto. Se evaluará su funcionalidad y facilidad de adaptación para integrarlo al programa educativo seleccionado.

En el capítulo 7, se hará un estudio detallado de la herramienta seleccionada para realizar el reconocimiento de voz, se estudiará su arquitectura y características.

En el capítulo 8, se analizarán las funcionalidades y características del software educativo seleccionado (JClic), presentando algunos elementos técnicos que sentarán una base para el próximo capítulo.

En el capítulo 9, se propondrán las modificaciones a realizar en el software educativo a nivel de interfaz para poder incorporarle comandos por voz. Esto incluye cualquier tipo de modificación de interfaz que sea necesaria para la adaptación. Además, se explicará la configuración, utilización y preparación del motor de reconocimiento de voz en función de realizar la integración.

Finalmente, en el capítulo 10, se verán algunas conclusiones y se propondrán líneas de trabajo futuras.

RESULTADOS ESPERADOS

Del corriente trabajo se esperan obtener varios resultados de los que se pueden mencionar:

En base al estudio de las diferentes herramientas educativas, se espera adquirir conocimiento sobre las mismas, enfatizando la flexibilidad y posibilidad de sus interfaces para que sean accesibles por todos.

También generar un informe acerca de algunas ayudas técnicas y adaptaciones que están disponibles hoy día.

A partir del análisis y estudio de las diferentes tecnologías de reconocimiento de voz, se espera obtener un análisis comparativo de dichas tecnologías y ponderar su posible integración para un software educativo.

Finalmente, se tendrá como resultado el diseño de un prototipo de adaptación a una herramienta de autor, mediante comandos por voz.

CAPÍTULO 2

TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN APLICADAS A LA EDUCACIÓN

INTRODUCCIÓN

En este capítulo se comenzará por introducir el concepto de tecnología. A medida que la ciencia y la tecnología avanzan, los alumnos necesitan la habilidad para adaptarse a nuevas alternativas.

Según expresa Juana Sancho, algunas corrientes educativas prestaron una gran atención a todas las técnicas, en particular a las mecánicas, incorporándolas al «saber» (la ciencia). Esta incorporación ha sido tan completa que en algunos momentos se ha llegado a considerar no sólo que la técnica es un saber sino que el saber es fundamentalmente técnico. Esta fusión indisoluble (y aparentemente indispensable) entre ciencia y técnica abre un nuevo espacio de conocimiento, el de la tecnología. Se la puede considerar como una técnica que emplea conocimientos científicos y que a su vez fundamenta a la ciencia al darle una aplicación práctica. La tecnología se configura como un cuerpo de conocimientos que, además de utilizar el método científico, crea y/o transforma procesos materiales⁴ (Sancho, 1998).

En el presente trabajo se intenta poner de manifiesto la importancia de las nuevas tecnologías, que todo profesor ha de fomentar a lo largo de su actividad docente y que, realizada de forma coherente, podrá potenciar la eficacia educativa; con ello el alumno puede adquirir mayor autonomía, al enfrentarse a un trabajo intelectual con nuevas técnicas, que lo lleva a familiarizarse con nuevas fuentes de consulta y a aprender a generar su propio plan de aprendizaje.

En este capítulo se hará mención de la importancia de la incorporación de las Tecnologías de la Información y la Comunicación en el ámbito educativo. Además se introducirá una clasificación de programas (entre las muchas que se encuentran en la bibliografía), la cual agrupa los distintos softwares en cuatro categorías que son: los de propósito general, con fines formativos, de autoayudas y exclusivos para algún tipo de discapacidad⁴. En el presente capítulo se detallarán solamente las dos primeras categorías, dejando las categorías restantes para ser discutidas en el capítulo 3.

TECNOLOGÍA DE LA INFORMACIÓN Y LA COMUNICACIÓN APLICADA A LA EDUCACIÓN

En el ámbito educativo son de amplia aplicación las tecnologías de la información y la comunicación (TIC), que pueden definirse como cualquier medio, recurso, herramienta, técnica o dispositivo que favorece y desarrolla la información, la

⁴ Ordenador y discapacidad, Sánchez Montoya Rafael, Ciencias de la educación Preescolar y especial, Madrid, 2002.

comunicación y el conocimiento, definición que conlleva un marcado carácter práctico y aplicado, dentro del ámbito y sistema educativo, por lo que deberá considerarse además, como un soporte didáctico para el aprendizaje, un elemento para el trabajo cooperativo y colaborativo, y también como elemento de gestión y administración⁵.

La Comisión sobre Tecnología Educativa de los Estados Unidos en 1970 propuso la siguiente definición de tecnología educativa:

“Es una manera sistemática de diseñar, llevar a cabo y evaluar todo el proceso de aprendizaje y enseñanza en términos de objetivos específicos, basados en la investigación del aprendizaje y la comunicación humana, empleando una combinación de recursos humanos y materiales para conseguir un aprendizaje más efectivo.”⁶

En el artículo “La tecnología educativa apropiada y crítica”, Beatriz Fainholc define a la tecnología educativa como:

“una tecnología especial dentro del marco mayor de la Tecnología considerada como disciplina que posee objeto y métodos propios. Se trataría de la tecnología especial que se ocupa de las mediaciones pedagógicas desde una perspectiva sociocultural histórica y crítica llevada a cabo en los procesos de diseño, ejecución y evaluación/investigación de programas y materiales educativos para contextos convencionales como para los nuevos entornos virtuales de aprendizaje...”.

“...la organización integrada de personas, significados, conceptualizaciones, procedimientos, artefactos simples y/o equipos complejos electrificados, pertinentemente adaptados, a ser utilizados para la elaboración, implementación y evaluación de programas, proyectos y materiales educativos que tienden a la promoción del aprendizaje contextualizado de un modo libre y creador.”⁷

Las TIC representan todo un conjunto de aplicaciones que van a aportar matices de innovación y desarrollo en sus actuaciones en la comunidad educativa. Sin duda, pueden proporcionar una dimensión de enriquecimiento y mayor desarrollo de áreas de intervención, tanto en el proceso de enseñanza y aprendizaje, como en sus agentes:

⁵ Tecnología de la Información y Comunicación aplicada al alumnado con discapacidad: un acercamiento docente. Diego Jesús Luque Parra y Gemma Rodríguez Infante Universidad de Málaga, España. Extraído de http://www.educrea.cl/documentacion/articulos/educacion_especial/10_tecnologia_informacion_aplicada_alumnado_discapacidad--OK.html

⁶ Para una tecnología educativa, Visiones y conceptos sobre la tecnología educativa. Juan de Pablos Pons. Barcelona 2001.

⁷ “La tecnología educativa apropiada y crítica, Beatriz Fainholc - <http://contexto-educativo.com.ar/2002/3/nota-05.htm>”

profesores, alumnos y familias. Así, en ese marco, debe quedar patente que el papel del profesorado con las nuevas tecnologías, se amplía en su importancia, aunque dentro de una acción tutorial, mediadora y de guía de aprendizajes, lo que conlleva su necesidad de formación y preparación, así como la reflexión y desarrollo de actitudes innovadoras, haciendo una labor más educadora y favorecedora del aprender a aprender. En efecto, las TIC van a suponer para el profesorado un elemento de formación continua y actualización docente, de contacto con otros profesores y comunicación en general, haciendo que su trabajo sea más creativo y exigente, aunque ello no le ahorre esfuerzo y dedicación (**Tecnología de la Información y Comunicación aplicada al alumnado con discapacidad – 2010**).

Una de las formas en que las tecnologías podrían afectar la capacidad del intelecto humano tiene que ver con los cambios en el rendimiento, manifestados por los estudiantes en el curso de su actividad asistida por un programa o un ordenador. En este, como en otros muchos casos, el hecho de trabajar con una máquina inteligente influye en lo que hacen los estudiantes, en la calidad de lo que hacen, y en cuando lo hacen. Se llamará “efectos con la tecnología” a estos posibles resultados. Otro significado del término “efecto” se refiere a las transformaciones relativamente duraderas que se observan en las capacidades cognitivas generales de los estudiantes como consecuencia de su interacción con una computadora. A éste tipo de efectos pertenecen los cambios posteriores, en el dominio del conocimiento, de la habilidad, o bien de la profundidad de la comprensión, después de que el estudiante se aleje del ordenador. Vamos a llamar a estos resultados “efectos de la tecnología”. Un ejemplo podría ser el de un procesador de textos inteligente. Por un lado, los estudiantes podrían escribir mejor utilizando el ordenador, por otro lado, un ordenador tan inteligente podría enseñar principios básicos acerca del arte de escribir que ellos podrían aplicar ampliamente cuando les tocara escribir con una máquina muy simple. Esto sugiere de nuevo el efecto obtenido de la máquina.

Trabajar con el ordenador adecuado sirve para mucho más que permitir al principiante hacer la misma cosa pero con más rapidez y menos esfuerzo. La existencia de tales sistemas podría significar una nueva definición y una nueva estructuración de la tarea de aprender o de operar, de la misma manera que el lápiz ha significado una reestructuración cualitativa del acto de memorizar. En suma, la colaboración intelectual con estos ordenadores puede cambiar la relación entre la adquisición de conocimientos preparados y la construcción de nuevos conocimientos, a favor de este último. La obra de una persona en colaboración con la tecnología podría ser mucho más “inteligente” que la obra de la persona a solas (**Coparticipando en el conocimiento – 2010**).

La idea misma de trabajar con un ordenador se basa en la suposición de que los usuarios exploren, diseñen, investiguen, escriban o comprueben hipótesis con

métodos que acoplan la inteligencia de la máquina con la suya, en una participación conveniente en la tarea. La consecuencia es que el mayor rendimiento en el trabajo con un ordenador solo se logra cuando los estudiantes operan con atención y a conciencia.

Este estado de atención o concentración en la colaboración con un ordenador proviene como mínimo de dos corrientes. Una de estas corrientes es la tendencia general de las personas de ser procesadores atentos de la información que reciban. Otra corriente que en nuestro estudio es de mayor importancia, es la combinación de materiales, tarea y medios que estimulan la atención.

En resumen, los ordenadores ofrecen una posibilidad de colaboración capaz de ampliar el rendimiento intelectual del usuario. Pero el grado de realización o cumplimiento de ese potencial depende en gran medida del compromiso voluntario del usuario. No depende solamente de la "pareja" de interacción estudiante-ordenador sino de cómo se emprende la colaboración⁸.

En la siguiente sección se hará una introducción del uso de la tecnología en la educación, en particular, a la destinada a chicos con necesidades especiales. Además se verán diferentes programas que pueden ser utilizados en este ámbito.

TIC Y EDUCACIÓN ESPECIAL

El desarrollo tecnológico posibilita la realización de las actividades de la vida diaria de cualquier persona, esto es aún más evidente en las personas con discapacidad, al contribuir a hacer una vida autónoma o más independiente. Pero también es cierto que deben considerarse los elementos que hagan útil esa tecnología para el mayor número de personas, con una atención y búsqueda constantes en sus técnicas y recursos, evitando que puedan convertirse en barreras de comunicación, de información o de accesibilidad.

En cualquier caso, la tecnología (herramientas, técnicas o dispositivos) ha de pensarse para hacer la vida de una persona más plena y autónoma, lo que incluye actuaciones y atención a los aspectos de:

1. Asistencia y apoyo en la vida diaria.
2. Educación y formación.
3. Integración laboral y actividades de producción.
4. Cultura, ocio y tiempo libre.

⁸ Salomon, G.; Perkins, D. y Globerson, T. (1992): Coparticipación en el conocimiento: la ampliación de la inteligencia humana con las tecnologías inteligentes. Comunicación, Lenguaje y Educación.

5. Asistencia a la salud y bienestar.

En el ámbito y contexto educativo, la discapacidad se trata centrándose en la adecuación a las necesidades y características de la persona, valorándose sus aspectos, a fin de compensar, con medidas y recursos, las limitaciones y anular o reducir así la discapacidad.

Vale la pena recordar que estas necesidades son especiales por la atención a las dificultades y a los recursos que se precisan y, por ello, dejarían de serlo si la docencia, los recursos y el ajuste curricular, tuvieran un carácter ordinario en los centros, lo cual nos remite a la importancia de la valoración de los factores de contexto (accesibilidad, uso y adaptación) **(Luque, Rodríguez y Romero, 2005)**.

En todos los procesos de enseñanza y aprendizaje, la tecnología aplicada a la educación y sus técnicas específicas de adaptación a las características individuales y necesidades del alumnado, se conforman como un elemento de suma importancia, así como de particular interés para las personas con necesidades especiales, convirtiéndose en una herramienta de compensación y adecuación a su circunstancia **(Luque Parra, Rodríguez, 2009)**.

La educación con TIC es especialmente favorable para los alumnos y personas con necesidades especiales, además favorece la mejora de sus habilidades cognitivas y de desarrollo de capacidades, así como su integración y normalización. Pero también se corre el riesgo de la aparición de barreras que dificulten el acceso de la tecnología a su vida particular y cotidiana, por lo que es conveniente hacer algunas consideraciones sobre medidas que compensen las necesidades educativas del alumnado favoreciendo la igualdad de oportunidades.

Cabe esperar que, a mayor desarrollo de la accesibilidad menor sea la necesidad de recursos complementarios en personas con discapacidad.

Las ventajas de las TIC en el alumnado con necesidades educativas especiales son muchas, dentro de la versatilidad y la posibilidad de atención a la diversidad. Así, puede destacarse el papel importante que juegan en favorecer la estimulación y atención a sus necesidades en: la facilitación del aprendizaje de la lectoescritura; la comunicación y el lenguaje; la reeducación y rehabilitación en general; el aumento de la autoestima y la motivación y de la integración social, entre otros.

En la próxima sección se mostrará una clasificación de programas destinados a la enseñanza y el auto aprendizaje que permite el desarrollo de ciertas habilidades cognitivas. Además se verán algunos ejemplos de programas utilizados en el ámbito educativo.

LOS PROGRAMAS

No hay dos personas iguales y la computadora debe satisfacer las necesidades reales de cada una de ellas; si no fuera así, sería una herramienta inútil e incluso perjudicial que sólo serviría para suscitar nuevas necesidades.

PROGRAMAS DE PROPÓSITO GENERAL

Estos programas cuentan con la ventaja de que el usuario no necesita ninguna habilidad especial para manejarlos. Suelen tener un entorno amigable y se usan con agilidad mediante los menús y los iconos.

Las Suites están integradas por varias aplicaciones: procesadores de textos, gestores de bases de datos, hojas de cálculo, etc. Su uso en el ámbito educativo y laboral es muy amplio. Ej. Microsoft Office, Open Office, etc. **(Sánchez Montoya - 2002)**.

PROGRAMAS CON FINES FORMATIVOS

Son programas con carácter marcadamente educativo. Ayudan al usuario a adquirir destrezas o habilidades para mejorar su formación. En este grupo existe una gran cantidad de programas disponibles en el mercado, de los cuales hemos seleccionado algunos de los más populares para presentar en esta sección con el fin de describir algunas de sus aplicaciones en el ámbito educativo y analizar sus posibilidades de adaptación. Para ello, de cada programa seleccionado se presenta una breve descripción funcional, el tipo de licencia, si posee adaptaciones o no y algunas de sus limitaciones **(Sánchez Montoya - 2002)**.

En cuanto a las adaptaciones, se analizarán las diferentes posibilidades que provee el programa para funcionar en conjunto con algunos de los dispositivos de adaptación disponibles, de los cuales, hablaremos en el capítulo 4. De las limitaciones interesa investigar si el producto está disponible para más de una plataforma, además se analizará qué tipo de licencia incluye cada programa, en otras palabras se quiere conocer si el producto es gratuito y si se dispone del código fuente para incorporarle mejoras o realizar cualquier modificación que el usuario desee.

Existen en el mercado, una gran cantidad de software educativo. Podríamos decir que hay herramientas para la gestión del proceso educativo, para la comunicación y para la creación de materiales educativos (herramientas de autor). De este último grupo, que es el que interesa en el presente trabajo, se puede mencionar algunos de los más utilizados en el ámbito⁹:

⁹ Información sacada de http://gluc.unicauca.edu.co/wiki/index.php/Herramientas_Autor_MECs (2009)

- **JClic:** Programa para la creación de diversos tipos de actividades educativas multimedia: puzzles, asociaciones, ejercicios de texto, crucigramas, sopas de letras, etc.
- **Sophie:** Programa para escribir y leer libros multimediales en un entorno de red.
- **Ardora:** Aplicación informática que permite crear actividades educativas de una forma muy sencilla. Se pueden crear más de 30 tipos distintos de actividades, crucigramas, sopas de letras, completar, paneles gráficos, relojes, etc. Este programa crea la página Web y el archivo (normalmente, applet de Java) que contiene la actividad.
- **Lim:** Es una aplicación para la creación de materiales educativos, formado por un editor de actividades (EdiLim), un visualizador (LIM) y un archivo en formato XML (libro) que define las propiedades del libro y las páginas que lo componen. Las actividades requieren del plugin de flash en el navegador Web.
- **Malted:** Es un sistema de autor gratuito que permite la creación de actividades y cursos multimedia para la enseñanza de lenguas extranjeras. Esta aplicación está formada por un navegador, un editor de actividades tipo applets y un depurador.
- **Markin:** Es un programa que facilita la corrección y devolución de documentos realizados por alumnos, proveyendo una herramienta que permite a los docentes marcar correcciones y devolver estos documentos con la mayor rapidez y facilidad.
- **Web questions:** Programa para elaborar cuestionarios interactivos en forma de páginas Web. Disponible para Windows de forma gratuita.
- **Hot potatoes:** Conjunto de herramientas de autor para elaborar ejercicios interactivos basados en páginas Web tales como ejercicios de elección múltiple (JBC), ejercicios de rellenar huecos (JCloze), creación de crucigramas (JCross), ejercicios de emparejamiento u ordenación (JMatch), ejercicios de reconstrucción de frases o párrafos a partir de palabras desordenadas (JMix).
- **Textoys:** El paquete Textoys está compuesto por dos componentes: Sequitur y Rhubarb. Las cuales permiten realizar dos tipos diferentes de actividades educativas que se explicarán más adelante.
- **Atenex:** Proporciona una plataforma colaborativa, en la cual se puede partir de contenidos ya existentes y construir unidades didácticas interactivas de manera fácil y para cualquier profesor.
- **Rayuela:** Es un conjunto de programas interactivos con los que se puede crear ejercicios para Internet, como por ejemplo: ahorcado, crucigrama, cuadro de mandos, hilo conductor, juego de lógica, opción múltiple, relacionar listas, rellenar huecos, rompecabezas, salto del caballo, sopa de letras, verdadero/falso. Este conjunto de aplicaciones son gratuitas.

- **FreeMind:** Programa para la creación de mapas mentales desplegados, desarrollado en Java.
- **CmapTools:** Esta herramienta facilita la construcción, búsqueda e intercambio de mapas conceptuales.
- **VUE (Visual Understanding Environment):** Es una herramienta flexible para la creación, administración e integración de recursos digitales a través de un entorno visual que permite la estructuración y presentación de información en forma de mapas.
- **Wink:** Programa para crear animaciones de extensión .swf a partir de la captura de imágenes sobre las tareas que se están realizando en un computador. Además, permite utilizar cajas de diálogo, botones y títulos para realizar explicaciones.

Se aclara que esta lista no es exhaustiva, pero intenta mostrar algunas de las aplicaciones reconocidas utilizadas como herramientas de autor.

Con el objetivo de tener un panorama variado de los programas disponibles, se analizarán con más detalles algunos de ellos, los cuales ofrecen diferentes funcionalidades y además se podrían utilizar en diversos niveles educativos.

Se optó por desarrollar una explicación de JClic, Texttoys y Hot Potatoes por ser considerados los más difundidos en utilización respecto de software similares. Se amplía la información de Markin, Lim y Wink para tomar un tipo de funcionalidad distinta.

LIM

El sistema LIM es un entorno para la creación de materiales educativos, formado por un editor de actividades (EdiLIM), un visualizador (LIM), y un archivo en formato XML (libro) que define las propiedades del libro y las páginas que lo componen (**Lim – 2010**).

Un material educativo creado con LIM se denomina libro y cada actividad se denomina página. Algunas de estas páginas son descriptivas, pues se utilizan para transmitir información a través de textos, imágenes, animaciones, mientras que otras son interactivas y se utilizan para realizar actividades (completar huecos, ordenar frases, sopa de letras, etc.).

No es necesario instalar software alguno para ver los materiales, ya que estos se presentan en formato de página web y utiliza la tecnología Flash (independiente del navegador).

Algunas de las ventajas que posee este software, desde el punto de vista educativo, son: presenta un entorno agradable y de fácil uso, cuenta con un gran número de

actividades en las que es posible evaluar los ejercicios. Además se puede utilizar en ordenadores, PDA y Pizarras Digitales Interactivas.

Para facilitar el trabajo de creación de libros existe la herramienta EdiLIM para entornos Microsoft Windows, la cual se presenta como un ejecutable de pequeño tamaño que no precisa instalación, pero se puede utilizar cualquier procesador de textos.

LICENCIA

LIM es gratuito, puede utilizarse libremente, sin limitación alguna, para cualquier tipo de uso: personal, comercial, educativo, etc. No está permitido modificar el código fuente sin autorización expresa del autor. Ya que EdiLIM/LIM son gratuitos se debe intentar que también lo sean las actividades con él creadas.

ADAPTACIONES

No posee.

LIMITACIONES

Para su funcionamiento necesita Microsoft Windows 98 o superior, ActiveX flash-player 8 o superior para Internet Explorer.

Los libros LIM pueden visualizarse en Linux, con la limitación impuesta por el plug-in flash, que para este sistema operativo es el 7.0. Esto impide que se puedan mostrar las imágenes en formatos .gif y .png. Actualmente existe un player alternativo: Gnash.

Adobe ya hizo pública una versión beta del reproductor flash 9.0 para Linux, con las mismas posibilidades que el equivalente de Windows.

EdiLIM es un ejecutable Windows, aunque está previsto que en el futuro pueda aparecer un editor para Linux. Actualmente, se puede ejecutar en Linux sobre la plataforma Wine.

Los libros-LIM son ficheros XML que se pueden crear con un editor de textos, por lo que no es imprescindible un programa de edición. Utilizando las etiquetas que definen el comportamiento de los libros-LIM, podemos elaborar actividades dinámicamente mediante lenguajes de servidor (PHP, ASP, etc.) o Javascript.

WINK

Wink es una herramienta de Software Libre de creación de presentaciones y tutoriales que permite realizar capturas de pantalla en la computadora (en formato BMP/JPG/PNG/TIFF/GIF) y le permite añadir explicaciones, botones, títulos y más. Además se permite grabar voz para agregar explicaciones a los videotutoriales creados (**Wink – 2010**).

Luego de realizado el tutorial, se puede exportar a diferentes formatos como: Macromedia Flash, Standalone EXE, PDF, PostScript, HTML o cualquier formato de imagen. Usar Flash/HTML para la publicación en web, EXE para distribuir versiones de escritorio y PDF para la creación de manuales imprimibles.

Permite crear presentaciones flash comprimidas, lo cual resulta ideal para usar en la Web.

Este programa, tiene soporte para múltiples lenguajes: Inglés, Francés, Alemán, Español y algunos más.

LICENCIA

Wink es un software con licencia freeware para uso comercial o personal, disponible para descargarse desde <http://www.debugmode.com/wink/download.php>. Sin embargo, si se desea redistribuir Wink se necesita el permiso de su autor.

ADAPTACIONES

No posee.

LIMITACIONES

No posee.

MARKIN

Muchas veces los estudiantes envían trabajos realizados en algún procesador de textos por correo electrónico, y los docentes deben realizar marcas de corrección y devolver estos documentos con la mayor rapidez y facilidad. Markin, facilita este circuito de interacción ofreciendo una herramienta que lo simplifique o sustituya **(Markin – 2010)**.

Es un programa Windows que se ejecuta sobre la computadora del profesor. Se puede importar el texto de un estudiante seleccionando el texto y copiándolo al portapapeles, o directamente desde un RTF (Rich Text Format o 'Formato de texto enriquecido') o un archivo de texto.

Una vez que el texto ha sido importado, proporciona todas las herramientas que el docente necesita para marcar y anotar sobre el documento a corregir.

Cuando se completa el marcado, el profesor puede exportar el texto como un archivo RTF para pasarlo a un procesador de textos, o como una página web para que los alumnos puedan ver lo que ha sido corregido en un navegador web.

También puede ser enviado directamente al estudiante, todo dentro del programa.

LICENCIA

Markin se puede descargar desde <http://www.cict.co.uk/software/markin/index.htm> y luego instalarlo sin tener que comprarlo. Lo que se descarga es un programa de funcionalidad completa pero que está restringido a marcar solamente textos muy cortos.

Para poder marcar textos largos debemos registrar el programa y adquirir su licencia.

Instituciones educativas como colegios y escuelas, a menudo necesitan licencias multiusuario. Todas las licencias multiusuario se basan en “seats”. Un “seat” es una licencia de usuario adicional para una copia completa del programa. Entonces, si una institución desea comprar el programa con una licencia que habilite a 25 miembros a usarlo, la institución deberá comprar una licencia completa y 24 “seats”.

ADAPTACIONES

No posee.

LIMITACIONES

No existe una versión para Unix/Linux o Mac y al parecer no hay un desarrollo en proceso, aunque se puede utilizar un emulador de Windows en máquinas con Mac o Unix/Linux, como el paquete Wine de Linux pero no existe garantía de que corra bajo otros emuladores. Los archivos generados por Markin son compatibles con navegadores Macintosh y Linux.

Aunque podemos elegir que el programa esté en español, los botones para realizar las correcciones sólo vienen en inglés, alemán e italiano, por lo que si el profesor necesita otro idioma debe crear una botonera.

HOT POTATOES

Es un programa para crear ejercicios educativos que se pueden resolver posteriormente a través de la web. Consta de seis herramientas para elaborar dichos contenidos digitales (**Hot Potatoes – 2010**).

Los ejercicios que se pueden crear son del tipo respuesta corta, selección múltiple, rellenar los huecos, crucigramas, emparejamiento y variados.

Estos ejercicios son páginas web estándar que usan XHTML 1.1 para mostrar el contenido y Javascript (ECMAScript) para la interacción. Estos estándares de la W3C son compatibles con todos los navegadores modernos como Internet Explorer 6+, Mozilla 1.2+, Phoenix, Safari, y muchos otros.

Las herramientas del programa manejan Unicode, haciendo posible crear ejercicios en prácticamente cualquier idioma, o en una mezcla de idiomas.

No se necesita tener conocimientos en XHTML o Javascript para usar el programa. Todo lo que se debe hacer es ingresar los datos (textos, preguntas, respuestas, etc.) y el programa va a crear la página web. Sin embargo, el programa está diseñado para que casi cualquier aspecto de la página web pueda ser personalizado, de manera que si se posee conocimiento sobre HTML o Javascript se puede hacer cualquier cambio que se desee, ya sea en la forma en que funcionan los ejercicios o en el formato de la página web.

LICENCIA

Hot Potatoes se puede descargar de <http://hotpot.uvic.ca/index.htm>. Pero la copia que se descarga es limitada, Hot Potatoes no es un programa gratuito. Para remover estas restricciones se debe registrar la copia.

El uso de Hot Potatoes es gratuito para instituciones educativas sin ánimo de lucro subvencionadas por el estado y para individuales que trabajan para estas instituciones a condición de que los materiales que se elaboren con este programa se distribuyan de forma gratuita y libremente a través de Internet a toda persona interesada. No obstante, la adquisición de una licencia es obligatoria en las siguientes circunstancias:

- Si el usuario no trabaja en una institución educativa sin ánimo de lucro.
- Si el usuario es trabajador por cuenta propia.
- Si el usuario cobra por el acceso a los materiales elaborados con Hot Potatoes.
- Si el usuario no distribuye gratuitamente los materiales a través de Internet.

Hay tres tipos de licencias: licencia por usuario, licencia para publicaciones grandes y licencia de actualización.

ADAPTACIONES

No posee.

LIMITACIONES

Si se desea usar este programa en Linux, solo se puede a través de una máquina virtual como Wine.

TEXTTOYS

El paquete Texttoys está compuesto por dos componentes: Sequitur y Rhubarb (**TextToys – 2010**).

Sequitur

Con este componente, al estudiante se le van presentando segmentos que pertenecen a un texto y el alumno debe ir seleccionándolos en el orden correcto. Para resolver la actividad se presentan tres partes del texto. Una de estas es el comienzo correcto. Cuando el estudiante elige el segmento correspondiente con el que comienza, este se agrega a un cuadro que se encuentra en la parte inferior de la página. Luego se muestran tres segmentos más, uno de estos es el correcto para continuar el texto. Gradualmente, el estudiante va construyendo de esta manera el contenido, eligiendo la parte adecuada en cada paso. A medida que el estudiante va resolviendo la actividad se mantiene una puntuación, que refleja la frecuencia con que hizo la elección correcta en el primer intento.

Rhubarb

En un ejercicio Rhubarb, al estudiante se le presenta un texto en el que las palabras son reemplazadas por estrellas. La actividad consiste en adivinar las palabras del texto una a una, escribiéndolas en una caja de texto. Si la palabra entera ingresada está en el texto, aparecen todas las instancias, reemplazando las estrellas en el texto mostrado. Poco a poco el estudiante puede construir el contenido, palabra a palabra, hasta completarlo.

Esta pantalla también cuenta con un botón “Hint” que se utiliza para dar ayudas. Si se presiona este botón se muestra una letra, que es con la que comienza la siguiente palabra a adivinar.

LICENCIA

El programa Textoys se puede descargar desde <http://www.cict.co.uk/software/textoys/download.htm>. Es un programa con licencia shareware que se puede descargar libremente desde internet.

Si se descarga la versión gratis de internet podrá utilizarse el programa, pero esta versión presenta dos limitaciones:

1. Sólo se pueden crear ejercicios cortos.
2. Se provee un servicio de hosting para los usuarios que compren el programa, es decir, provee un servidor para que puedan subir sus ejercicios y compartirlos a través de Internet. Por lo tanto, con la versión gratuita no se puede subir ejercicios a su servidor, ni se puede acceder al sistema de carga de resultado de hotpotatoes.net. Si se accede a esta funcionalidad, los docentes podrán cargar sus actividades a este servidor, y los alumnos podrán resolver los ejercicios a través de internet. Sus puntuaciones serán almacenadas en este servidor. Con esta característica los docentes podrán revisar y descargar los ejercicios

resueltos por los estudiantes. Para acceder a estas características, se debe registrar y adquirir la licencia.

ADAPTACIONES

No posee.

LIMITACIONES

Provee solo dos tipos de actividades, no es un ambiente educativo completo. Además, no existe una versión disponible para Linux, este deberá ejecutarse en una máquina virtual como Wine.

JCLIC

Es un entorno para la creación, realización y evaluación de actividades educativas multimedia. Es el sucesor de Clic, pero desarrollado en plataforma Java. Este programa sirve para trabajar con diferentes tipos de actividades educativas: rompecabezas, asociaciones, ejercicios de texto, de memoria, sopa de letras, crucigramas, etc. **(JClic, sitio oficial – 2010).**

Las actividades se pueden agrupar en secuencias llamadas proyectos. Un proyecto es, básicamente, una lista ordenada de actividades que se presentan al alumno. El paso de una actividad a la siguiente se puede realizar a voluntad del usuario (haciendo clic en la flechas de desplazamiento) o de manera automática (el programa pasa a la siguiente actividad una vez transcurrido cierto tiempo).

JClic está formado por tres aplicaciones:

- JClic Player: que puede presentarse como un applet o como un programa independiente. Si se presenta en forma de applet, se puede incrustar la actividad en un navegador web, sin necesidad de instalar. Se descarga automáticamente la primera vez que se visita la página con la actividad JClic incrustada. Si se presenta como un programa independiente, una vez instalado, permite realizar las actividades desde el disco duro del ordenador (o desde la red) sin que sea necesario estar conectado a Internet.
- JClic Author: es la herramienta de autor que permite crear, editar y publicar las actividades.
- JClic Reports: es el módulo encargado de recopilar los datos (tiempo empleado en cada actividad, intentos, aciertos, etc.) y presentarlos después en informes estadísticos de diversos tipos.

LICENCIA

JClic es un proyecto de software libre que el departamento de Educación de la generalitat de Cataluña pone a disposición de la comunidad bajo los términos de la Licencia Pública General de GNU (GPL).

El código fuente de JClic está disponible para su descarga.

ADAPTACIONES

Clic, el antecesor de JClic, provee opciones de accesibilidad para ser manejado mediante el sistema de barrido¹⁰. Para ello se debe activar la función cursor automático dentro de las opciones globales del menú principal. En el próximo capítulo trataremos el tema de las diferentes adaptaciones disponibles aplicables a las TIC.

LIMITACIONES

No posee.

ARDORA

Es una aplicación informática para docentes, que les permite crear sus propias actividades, en formato HTML, para sus alumnos de un modo muy sencillo. Se pueden crear más de 45 tipos distintos de actividades: crucigramas, sopas de letras, completar, paneles gráficos, relojes, etc., así como más de 10 tipos distintos de páginas multimedia: galerías, panorámicas o zooms de imágenes, reproductores .mp3 o .flv, etc. (**Ardora – 2010**).

El profesor sólo debe centrar su esfuerzo en los elementos de la actividad, no en su tratamiento informático. Una vez introducidos los elementos de la actividad, mediante formularios muy sencillos, Ardora creará la página web y los archivos necesarios, luego se puede visualizar el contenido a través de un navegador.

A partir de la versión 5.1, no necesita ser instalado, únicamente se debe descargar el archivo exe.

LICENCIA

¹⁰ Este concepto se verá en el capítulo 3.

Es un programa gratuito, siempre y cuando sea usado de forma personal, sin carácter lucrativo y con fines estrictamente educativos. No está disponible el código.

ADAPTACIONES

No posee.

LIMITACIONES

No existe una versión disponible para Linux, este deberá ejecutarse en una máquina virtual como Wine. Ardora fue probado en Ubuntu 8.04, OpenSuse 11 (32 e 64 bits) y Guadalinex.V6.

RESUMEN COMPARATIVO DE ALGUNOS SOFTWARE CON FINES FORMATIVOS

En la siguiente tabla se mostrará un esquema comparativo de las distintas características presentes en los softwares presentados en la sección anterior.

Software	Utilidad	Licencia	Adaptaciones	Limitaciones
Lim	Creación de material educativo.	Freeware	No posee	No es independiente de la plataforma
Wink	Creación de video tutoriales	Freeware	No posee	
Markin	Corrección de trabajos. Anotaciones.	Shareware	No posee	No es independiente de la plataforma.
Hot Potatoes	Creación de ejercicios educativos.	Freeware	No posee	No es independiente de la plataforma.
Textoyo	Resolución de actividades de texto.	Shareware	No posee	No es independiente de la plataforma.
Jclic	Creación y resolución de actividades educativas multimedia.	GPL	No posee. Su antecesor Clic posee barrido automático.	
Ardora	Creación de actividades educativas en formato HTML.	Freeware	No posee	No es independiente de la plataforma

Tabla 2.1 - Cuadro Comparativo de Programas con Fines formativos.

CONCLUSIÓN

Se puede concluir que la integración de las TIC en los ámbitos educativos puede favorecer la sensibilización, la aceptación y la comprensión del alumnado con discapacidad. Los docentes deben tener cierta formación para la utilización de software aplicado a la enseñanza, de manera de lograr así la promoción de actitudes hacia patrones de accesibilidad universal y de diseño para todos. De esta manera, las TIC, en su uso educativo, contribuyen a que el principio fundamental de igualdad de oportunidades se convierta en una realidad personal y social, haciendo posible la información y acceso a la formación y progreso personal y ciudadano del alumnado con discapacidad. Todavía resta mucho por hacer en este sentido ofreciendo mayor cantidad de adaptaciones y accesibilidad en los software disponibles hoy día para el ámbito educativo (**Tecnología de la Información y Comunicación aplicada al alumnado con discapacidad – 2010**).

El estudio de la discapacidad implica situarla en términos de persona-situación, evitando acentuar lo negativo de la limitación individual o deficiencia, y valorando toda ejecución de cualquier acción precisa, tanto habilidad y capacidad como disponibilidad y oportunidad. Desde este marco, toda tecnología accesible y adaptada promoverá que los alumnos con necesidades educativas especiales desarrollen sus capacidades y se desenvuelvan en la comunidad.

En este capítulo se nombró una clasificación de los programas en cuatro categorías y se presentaron sólo dos de ellas. En el siguiente capítulo se mostrarán los programas de autoayudas informáticas los cuales son un grupo de programas o utilidades que se cargan en memoria, antes que el programa principal, con el fin de ofrecer una ayuda adicional a personas con necesidades especiales. Finalmente los programas exclusivos según la discapacidad, agrupan programas diseñados para personas con una determinada necesidad.

CAPÍTULO 3

ACCESIBILIDAD Y ADAPTABILIDAD

INTRODUCCIÓN

Actualmente, son muchos los que ya consideran a las TIC como “rampas tecnológicas” que permiten usar el mismo software a todas las personas, sin importar las capacidades particulares o imposibilidades que tengan. Intentan paliar la falta de previsión de algunos fabricantes que diseñan sus productos, pensando en un usuario estándar, y olvidan de que existe una minoría que demanda pequeñas adaptaciones.

Muchas de las TIC son gratuitas y sus objetivos son variados: algunas leen la información textual de la pantalla, otras consiguen que la computadora trabaje más lentamente para que el usuario, al disponer de más tiempo, pueda responder adecuadamente a alguna consigna determinada, y otras ofrecen redundancia visual o auditiva de salida consiguiendo que las indicaciones del software puedan ser percibidas por los usuarios con deficiencia sensorial. Estos son sólo algunos ejemplos, pero existen muchos más.

Se considerará accesibilidad como la capacidad de utilizar recursos y/o servicios accesibles y, por lo tanto, susceptibles de ser adaptados a cada necesidad específica. La modularidad, adaptabilidad y accesibilidad, deberán ser tenidas en cuenta ya en el mismo diseño de software, considerándose actividades complementarias para evitar nuevas barreras de acceso a las personas que tienen dificultades, ya sea para leer el contenido de la pantalla, manejar el mouse o el teclado, escuchar las indicaciones sonoras de los altavoces, con un nivel de comprensión reducida, etc.

La accesibilidad y la adaptabilidad del software son dos conceptos de gran importancia que fueron inspiradores al momento de comenzar a realizar esta tesina. Debido a que uno de los objetivos es realizar un prototipo de adaptación de un software para que provea reconocimiento de voz, un punto importante a tener en cuenta es la accesibilidad que se alcanza a través de dicha adaptación.

En la siguiente sección se analizarán los conceptos mencionados anteriormente, para luego detallar una Norma de Accesibilidad, la cual se va a tener en cuenta al momento de analizar las características del software a adaptar y también al momento de plantear los pasos a seguir para llevar a cabo este objetivo.

En el capítulo anterior se realizó una clasificación de los programas en cuatro categorías, de las cuales sólo se explicaron dos. Los otros dos conceptos se verán en la próxima sección.

CONCEPTOS DE ACCESIBILIDAD Y ADAPTABILIDAD

Se puede definir a la accesibilidad como el grado en el que todas las personas pueden utilizar un objeto, visitar un lugar o acceder a un servicio, independientemente de sus capacidades técnicas, cognitivas o físicas (**Wikipedia Accesibilidad - 2010**).

Para promover este concepto, se hace uso de ciertas facilidades que ayudan a salvar los obstáculos o barreras de accesibilidad del entorno, consiguiendo que las personas realicen la misma acción que pudiera llevar a cabo un individuo sin ningún tipo de discapacidad. Estas facilidades son llamadas ayudas técnicas y se estudian más adelante.

En el ámbito de la informática, se incluyen ayudas como las tipografías de alto contraste o gran tamaño, magnificadores de pantalla, lectores y revisores de pantalla, programas de reconocimiento de voz, teclados adaptados, y otros dispositivos apuntadores y de entrada de información.

Cuando se aplica al contenido de Internet se denomina accesibilidad web. En la web, el W3C¹¹ ha desarrollado directrices o pautas específicas para permitir y asegurar este tipo de accesibilidad. El grupo de trabajo dentro este consorcio encargado de promoverla es el WAI (Web Accessibility Initiative), elaborando para ello Pautas de Accesibilidad al Contenido de la Web (WCAG).

El concepto de accesibilidad en Informática, comienza por los periféricos con los que se comunica una persona con el computador, el teclado y el mouse. Por ejemplo, para personas con discapacidades existen teclados especiales en los cuales las teclas son mucho más grandes y no se puede apretar más de una al mismo tiempo. Estas características son incorporadas para brindarle mayor usabilidad. En el caso del mouse, la usabilidad viene dada en otro aspecto, este cambia su forma para convertirse en un teclado de cuatro teclas, donde cada una representa una dirección. Así es que una persona al apretar la tecla representada por una flecha hacia arriba está haciendo que el mouse se mueva en esa dirección, a su vez, pueden combinarse las teclas de izquierda y arriba para hacer que el mouse se mueva en diagonal, disminuyendo la motricidad necesaria para realizar el movimiento. Ambos tipos de dispositivos, los comunes y los orientados a la usabilidad, pueden convivir y, dependiendo el usuario, pueden intercambiarse (**Access Total – 2010**).

La capacidad de adaptación de los sistemas de software es una característica altamente deseable. Un programa adaptable puede tolerar cambios en su ambiente

¹¹ Ver definición en Anexo B.

sin intervención externa (**Adaptabilidad de interfaces de usuario por reflexión – 2010**).

La etapa de diseño de un software cobra gran importancia cuando se habla de adaptabilidad ya que es el momento en el que se plantea la funcionalidad del mismo y quiénes van a utilizarlo, en otras palabras, se evaluarán las adaptaciones necesarias para los posibles destinatarios. Si se tiene en cuenta que puede resultar imposible conocer de antemano todos los posibles usuarios y sus respectivas discapacidades se debe tener como meta lograr un diseño pensado para permitir la inclusión de nuevas adaptaciones.

REVISIÓN DE ALGUNAS NORMAS DE ACCESIBILIDAD

En esta sección, se revisan algunos incisos de diferentes normas que se han consultado y que resultan de interés en el marco de este trabajo.

Las Naciones Unidas aprobaron en el año 1993 las "Normas Uniformes sobre la igualdad de oportunidades para las personas con discapacidad", cuya finalidad es "garantizar que niñas y niños, mujeres y hombres con discapacidad, en su calidad de miembros de sus respectivas sociedades, puedan tener los mismos derechos y obligaciones que los demás". El fundamento político y moral de estas normas se encuentra en la "Carta Internacional de Derechos Humanos" (**Wikipedia Accesibilidad - 2010**).

El artículo 5, "Posibilidades de acceso", de estas normas declara que "los Estados deben reconocer la importancia global de las posibilidades de acceso dentro del proceso de lograr la igualdad de oportunidades en todas las esferas de la sociedad. Para las personas con discapacidades de cualquier índole, los Estados deben (a) establecer programas de acción para que el entorno físico sea accesible y (b) adoptar medidas para garantizar el acceso a la información y la comunicación."¹²

b) Acceso a la información y la comunicación

"5. Las personas con discapacidad y, cuando proceda, sus familias y quienes abogan en su favor deben tener acceso en todas las etapas a una información completa sobre el diagnóstico, los derechos, y los servicios y programas disponibles. Esa información debe presentarse en forma que resulte accesible para las personas con discapacidad."

¹² Normas Uniformes sobre la igualdad de oportunidades para las personas con discapacidad <http://www.adeseccom.ar/pdflegislacion/NormasUniformes.pdf>

“10. Los Estados deben velar por que los nuevos sistemas de servicios y de datos informatizados que se ofrezcan al público en general, sean desde un comienzo accesibles a las personas con discapacidad, o se adapten para hacerlos accesibles a ellas.”

Actualmente Argentina no cuenta con una ley de accesibilidad web, pero sí con un proyecto de ley que tiene media sanción de la Cámara de Diputados y se encuentra en espera de ser aprobado por la Cámara de Senadores. Este proyecto no tiene en consideración los programas de escritorio.

En España ya se cuenta con un gran número de leyes, decretos y normativas, algunas de ellas son las siguientes:

- Normativa UNE 139802, llamada “Aplicaciones informáticas para personas con discapacidad, Requisitos de accesibilidad al ordenador, Software”.
- Normativa UNE 139803: llamada “Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad para contenidos en la Web”, proporciona soluciones accesibles para los desarrolladores web, creando un listado de recursos que permiten definir las características que han de cumplirse en materia de los contenidos web en Internet, Intranets y en cualquier otro tipo de redes informáticas, para que éstos puedan ser utilizados por el mayor número de personas, incluyendo las personas con discapacidad y las personas de edad avanzada.

Existe un particular interés en la Normativa UNE 139802 debido a que incluye normas para la accesibilidad al ordenador y al software, las cuales resultarán interesantes al momento de plantear el prototipo de adaptación que se presentará en el capítulo 9.

ALGUNAS NORMAS DE ACCESIBILIDAD DE LA INFORMÁTICA EN ESPAÑA

Se cuenta actualmente con normas utilizadas para el desarrollo de aplicaciones accesibles. La publicación vigente de la UNE 139802 es una versión revisada de la Norma Experimental UNE 139802 EX publicada en el año 1998 por el comité técnico AEN/CTN 139¹³. Esta norma establece las características que ha de cumplir el software de un ordenador, incluyendo su entorno operativo (sistema operativo más la interfaz de usuario asociada), las aplicaciones informáticas y la documentación asociada, para que puedan ser utilizados por la mayor cantidad de personas, incluyendo aquellas con

¹³ AENOR – CTNs - <http://www.aenor.es/desarrollo/normalizacion/ctns/buscadorctns.asp>

discapacidad y de edad avanzada, de forma autónoma o mediante las ayudas técnicas pertinentes (**Aenor - 2010**).

Esta norma impacta en cualquier tipo de aplicación informática, tenga o no interacción directa con el usuario. Se incluyen los sistemas operativos, entornos de ventanas y controladores de dispositivos. También se aplica a la documentación que se aporta con ella.

Considerando el tipo de producto que se desarrolle, se clasifican en tres secciones independientes (**Normas de Accesibilidad a la Informática – 2010**):

1. Entorno operativo: denominación que engloba al sistema operativo, a su interfaz de usuario asociado y a algunas de las aplicaciones que suelen venir con él (administrador de archivos, programas de configuración, etc.). Ejemplos: Windows, MacOS, OS/2 y Linux.
2. Aplicaciones: cualquier programa de uso general o a medida, como puede ser un procesador de textos, un programa de diseño, un juego, un compresor de archivos, entre otros.
3. Autopistas de la información: además de los contenidos de las páginas web, incluye los navegadores, los programas para correo electrónico y cualquier otra herramienta asociada a los servicios de Internet.

REQUISITOS GENERALES

Existen una serie de requisitos que afectan por igual a las tres secciones: diseños ergonómicos, configuraciones personalizables y multiplicidad de canales. A continuación, se abordan estos requisitos generales, y posteriormente se exponen los más específicos del sistema operativo, las aplicaciones e Internet.

▪ Mensajes

Una interfaz debe ser concisa, coherente y consistente si se quiere reducir el esfuerzo que el usuario necesitará para trabajar con su ordenador.

Los mensajes del tipo "Error 108:345, overflow en la pila en posición 4012. Perderá todos los datos. Pulse cualquier tecla para continuar" resultan confusos para los usuarios. Especialmente para los niños, los principiantes en la informática o aquellos usuarios con problemas de aprendizaje, que tal vez no comprendan lo que significa y se pondrán muy nerviosos pensando que es culpa suya. Por ello, se recomienda el uso de mensajes cortos y sencillos.

También resulta conveniente que el mismo mensaje siempre tenga el mismo texto, que salga en la misma zona de la pantalla y con los mismos elementos compositivos (tipo de letra, colores, botones, etc.). De esta forma, se puede identificar el mensaje por su aspecto además de por su texto y resulta fácil localizarlo en la pantalla. Lo mejor es utilizar, cuando sea posible, las convenciones del entorno operativo.

El tiempo de reacción ante los eventos es muy variable de un usuario a otro, por ello, resulta contraproducente poner mensajes en pantalla que desaparecen automáticamente transcurrido un tiempo. En la norma se exige una espera hasta que el usuario acepte el mensaje. Si no fuese posible, se pide que el tiempo de permanencia en pantalla sea configurable por el usuario. Este requisito se hace especialmente necesario en mensajes críticos del sistema operativo.

La rapidez con que se genera el mensaje es también muy importante y afecta especialmente a los mensajes de voz, ya que debe existir una coherencia temporal entre lo que se está escuchando y lo que realmente está sucediendo.

- Redundancia de canal

Resuelve muchos de los problemas de accesibilidad. Por ejemplo, los usuarios daltónicos, no pueden distinguir algunos colores, por lo tanto si una información se apoya sólo en los colores, como "Pulse en el botón rojo para terminar", estas personas no sabrán cómo actuar. Por lo tanto, las características estéticas de la interfaz sólo deben servir para acompañar o realzar, enviando la información por múltiples canales: color y texto, color y forma, color, texto y forma, etc.

Se genera una situación similar con la utilización del sonido. Habitualmente se utiliza como indicador de fin de una tarea o como alerta de algún tipo de error. Los usuarios con problemas auditivos pierden este tipo de información, por lo que debe ir acompañada de una señal visual asociada al evento.

No sólo debe existir redundancia de canal de salida, sino también en el de entrada. Por ejemplo, para las aplicaciones de Internet, debe ser posible hacer uso de ellas sin necesidad de tener un ratón.

- Textos y gráficos

Los lectores de pantalla empleados por los usuarios ciegos, no son capaces de leer textos escritos usando primitivas gráficas. Por lo tanto, aquellos que se escriban en pantalla deben utilizar los servicios de escritura de texto facilitados por el entorno operativo.

Del mismo modo, cualquier foto, vídeo, dibujo o gráfico queda fuera del alcance de los lectores de pantalla. Por ello, cuando se utilicen, deberán ir acompañados por textos explicativos, que permitan a las personas no videntes obtener información acerca del contenido de la imagen.

- Introducción de datos

Se realiza de manera similar en las interfaces de modo texto y en las de modo gráfico. Si bien en las segundas ha sido necesario crear el elemento "cuadro de edición". En cualquiera de los dos casos, el texto escrito se debe poder recorrer con el cursor para que un lector de pantalla pueda sintetizarlo a voz o convertirlo en Braille.

En los campos para introducción de datos, la etiqueta identificativa que los acompaña debe estar alineada horizontalmente con la primera línea del campo, de manera que sean fácilmente asociables tanto para los usuarios de lectores de pantalla como para los más novatos.

- Personalizar el teclado

Es un periférico esencial, por lo que todos los aspectos de accesibilidad deben ser contemplados con suma atención.

El usuario debe poder acceder a cualquier elemento de la interfaz desde el teclado, por ejemplo, activar y desactivar los menús y desplazarse por sus opciones. Además, se debe evitar el uso de acciones simultáneas (como mantener apretada una tecla mientras se pulsa otra) y si no se evitan, se debe proporcionar un método secuencial alternativo para lograr el mismo resultado.

En determinados casos, moverse por los menús con las flechas del cursor puede resultar un proceso penoso, por lo que cuando sea posible, es conveniente poner teclas de aceleración o atajos. Esto es muy útil para los usuarios con problemas de precisión o de visión, puesto que pueden necesitar varios intentos recorriendo el menú hasta acertar con la opción deseada.

Para acelerar el recorrido con teclado, los menús deben ser circulares, es decir, se salta de la última opción a la primera y viceversa. Este recorrido circular debe aplicarse también al movimiento por las opciones de un cuadro de diálogo, al cambio de ventana o área de trabajo, etc.

- Iconos

Para las personas con problemas de visión resulta incómodo, y a veces imposible, percibir los iconos y otros pequeños objetos, por lo que el propio entorno operativo debe permitir que se modifiquen sus tamaños y sus posiciones, independientemente o por grupos. Deben además, tener asociada una etiqueta, facilitando la identificación y comprensión de la función del icono.

- Ventanas

La gestión de las ventanas, tales como refrescar, mover, cambiar de tamaño, etc., habitualmente se realizan por medio del ratón, pero para los usuarios con problemas de precisión o con ceguera el uso de este resulta un inconveniente. Por

eso, la norma exige que todas estas operaciones se puedan realizar también con el teclado.

En el caso específico de las barras de herramientas, a las que no se puede acceder por teclado, se exige que todas las operaciones sean accesibles también a través de opciones de menú.

Dado que hay usuarios que necesitan una herramienta especial para el acceso a su ordenador (emuladores de teclado, predicción de palabras, etc.), y ésta debe permanecer visible en pantalla, se exige que las ventanas puedan cambiar de tamaño y de posición. También se exige que sean maximizables, minimizables y que se puedan cerrar, para evitar conflictos con dichas herramientas. Además, se le pide al entorno operativo que facilite una manera de cambiar de una ventana a otra, para que las aplicaciones especiales puedan cooperar con las generales.

- Servicios de ayuda al usuario

Los entornos operativos establecen servicios de ayuda que luego son utilizados por muchas aplicaciones. Esta ayuda suele tener formato textual, pero se debe incluir también la posibilidad de incorporar imágenes, por ejemplo, para explicar un concepto nuevo mediante la lengua de signos.

ENTORNO OPERATIVO

Es el responsable de todos los elementos que conforman la comunicación básica del usuario con la computadora. Además, el desarrollo de la interfaz hombre-máquina ha cobrado una importancia fundamental en los últimos años.

Las personas con ceguera suelen tener graves problemas de accesibilidad. Todo lo que se ve en una pantalla está pensado y diseñado para ser visto, especialmente en las interfaces gráficas, donde se encuentran un sinnúmero de aplicaciones abiertas simultáneamente en diferentes ventanas y cada ventana tiene la complejidad que antes tenía toda una pantalla.

También los usuarios con dificultades físicas, que utilizan dispositivos o programas especiales (reconocimiento de voz, emulador de teclado, etc.), pueden ver limitado el acceso, principalmente, por problemas de compatibilidad entre estas ayudas y el entorno operativo o las aplicaciones.

- Servicios del sistema

En líneas generales, la norma exige que el entorno operativo proporcione al usuario acceso para cualquier dispositivo de entrada que utilice (ratón de cabeza, pulsador, etc.) y recomienda que proporcione también un sistema de reconocimiento de voz. De igual forma, la salida de datos se debe realizar tanto por

video como por audio, para que los usuarios con ceguera tengan acceso a la misma información.

Todas las opciones deben tener carácter de activación opcional, de manera que la misma plataforma informática pueda ser utilizada indistintamente por un amplio abanico de usuarios con diferentes necesidades. Además, los servicios del entorno operativo, deben estar diseñados de manera que sean capaces de garantizar que las aplicaciones construidas por encima de él puedan ser accesibles.

Por ejemplo, proporcionar un servicio para mostrar texto que a la vez se emita con síntesis de voz.

Para realizar una descripción hablada, actualmente los lectores de pantalla utilizan "modelos de pantalla", que mantienen una representación exacta de los elementos visualizados. Para facilitar este modelo, el entorno operativo, debe habilitar la creación de los elementos de la interfaz con una etiqueta que los identifique y que permita acceder a sus propiedades (ventana abierta o cerrada, tamaño, foco, etc.), preferiblemente a través de servicios predeterminados.

Además de posibilitar el cambio de foco de una ventana a otra, los entornos operativos que incorporan el concepto de áreas de trabajo, también deben ofrecer el cambio de una a otra, ejecutándolo tanto desde el teclado como desde el ratón.

- Controlador del teclado

Es el programa que se encarga de las comunicaciones entre el ordenador y el teclado y donde es posible incorporar muchas prestaciones que faciliten la accesibilidad. Las personas que más dificultades tienen para el uso del teclado son las que tienen problemas de precisión en el uso de sus brazos, dedos o manos, seguidas de las personas con discapacidades psíquicas y visuales, por lo que se deben contemplar las diferentes problemáticas.

Para los usuarios con problemas de control fino (mantienen pulsada una tecla demasiado tiempo, se equivocan de tecla, pulsan repetitivamente la tecla debido a temblores), debe permitir configurar el tiempo de repetición de la tecla, el tiempo de pulsación antes de ser aceptada y el tiempo de rechazo de pulsaciones repetitivas de la misma tecla.

Es necesario incorporar una opción que permita bloquear las teclas de control (Mayúsculas, Alt, Ctrl, etc.), de manera que las personas que sólo puedan utilizar una mano o un puntero, eviten las maniobras de pulsación simultánea. Además, los usuarios con discapacidad visual necesitan saber el estado de dichas teclas.

Para usuarios de una sola mano, un puntero o que utilizan los nudillos se debe ofrecer la posibilidad de reconfigurar todas las teclas del teclado para permitir adaptarse a sus necesidades.

- Controlador del mouse

Más aún que el teclado, el mouse es una barrera para las personas con problemas de precisión, de movilidad o de fuerza en los miembros superiores.

Por ello, se debe permitir modificar la orientación en el movimiento del puntero para que el usuario pueda manejarlo de la forma más ergonómica a su movilidad. De igual forma, se debe permitir modificar la velocidad y aceleración del puntero, diferenciando entre la velocidad horizontal y vertical, el tiempo de aceptación del clic y el tiempo entre dos clics.

Además, para las personas con problemas de movilidad en los dedos, se debe permitir realizar el bloqueo del arrastre, disponiendo de un botón del ratón para esta función o, en su defecto, utilizando una temporización de uno de los botones o una tecla.

Para los usuarios zurdos, se deben poder intercambiar las funciones de los botones derecho e izquierdo.

APLICACIONES

Todo lo comentado anteriormente sobre el diseño de las ventanas, los iconos, los mensajes, etc. deberá tenerse en cuenta en el desarrollo de cualquier aplicación. Además, para que los elementos textuales y de identificación (nombre de la ventana, etiqueta del icono, etc.) sean susceptibles de emitirse por voz, las aplicaciones deben utilizar los servicios que facilita el entorno operativo; de esta forma, los lectores de pantalla podrán identificar la aplicación y sus contenidos.

Se debe tener en cuenta que, a pesar de los avances realizados, los usuarios con problemas de accesibilidad necesitan algunas veces utilizar dispositivos o programas especiales, por lo que la norma exige a la aplicación que coopere con estas herramientas de acceso, de manera que a veces puedan superponerse en pantalla, incluso en entornos que no sean de ventanas.

Para ello, se deben utilizar los mecanismos de coordinación proporcionados por el entorno, evitando que las aplicaciones se bloqueen las unas a las otras.

Si una aplicación utiliza ventanas, su gestión debe dejarse al entorno operativo a través de los servicios que aporta, puesto que será el encargado de facilitar la accesibilidad. Si se utilizan varias ventanas, se debe permitir el cambio de una a otra y se debe seguir siempre la misma secuencia de cambio.

Finalmente, y para evitar problemas de consistencia y de coordinación entre aplicaciones, toda aplicación debe tener una opción de finalizar.

Para conseguir que una interfaz sea completamente accesible, no basta con que existan todos los servicios y los requisitos generales estipulados hasta ahora. Hace falta que además las aplicaciones usen esos servicios, se coordinen con ellos y cumplan otras consideraciones de diseño no asignables directamente al entorno operativo.

Se debe prestar especial atención a que todas las funciones ofrecidas por la aplicación sean accesibles por teclado, requisito especialmente descuidado en programas de dibujo, juegos, aplicaciones musicales, etc. A la hora de acceder con el teclado a los menús, se deben respetar las combinaciones habituales del entorno operativo, por ejemplo, los atajos de teclado que se ponen en varias de las opciones de un menú.

Además, la aplicación se debe diseñar de manera que el número de pasos necesarios para acceder a cualquier opción sea el mínimo posible y no requiera el uso simultáneo de más de un dispositivo de entrada, haciendo especial hincapié en las opciones más frecuentemente utilizadas. De esta manera cualquier usuario conseguirá una mayor eficiencia.

AUTOPISTAS DE INFORMACIÓN

Las normas en relación a las autopistas de información, se detallan en el documento UNE 139803, el cual ha tomado como punto de partida para su realización las Directrices para la Accesibilidad de los Contenidos Web (en inglés Web Accessibility Initiative) del Consorcio de la web (en inglés, World Wide Web Consortium). Esta norma establece las características que deberán de cumplir los contenidos disponibles mediante tecnologías web en Internet, Intranets y otro tipo de redes informáticas, para que puedan ser utilizados por la mayor cantidad de personas, incluyendo personas con discapacidad y personas de edad avanzada, de forma autónoma o mediante las ayudas técnicas pertinentes (**Normas de Accesibilidad a la Informática – 2010**).

Esta norma cubre la mayoría de los tipos de discapacidad, incluyendo personas con deficiencias físicas moderadas o severas, deficiencia visual o ceguera, deficiencia auditiva o sordera y discapacidad cognitiva.

También se han tenido en cuenta las necesidades de las personas de edad avanzada, que quedan englobadas en las necesidades especiales de todas las discapacidades, considerando que al envejecer, se limitan poco a poco alguna, muchas o todas las discapacidades.

Se aplica a cualquier tipo de contenido disponible en redes informáticas, con especial énfasis en los contenidos web que son accedidos mediante programas informáticos

llamados aplicaciones de usuario (siendo los más habituales los llamados navegadores web). Esto es así, independientemente de cómo se crean esos contenidos.

Esta norma no se aplica a los programas utilizados para acceder a los contenidos (aplicaciones de usuario), a los usados para crear los contenidos (herramientas de autor), ni a las aplicaciones existentes en la red que se ejecutan en el ordenador del usuario (como los applets).

Se ampliará el tema de la accesibilidad web en la siguiente sección.

ACCESIBILIDAD WEB

La accesibilidad web significa que personas con algún tipo de discapacidad van a poder hacer uso de la web. En concreto, al hablar de accesibilidad web se está haciendo referencia a un diseño web que va a permitir que estas personas puedan percibir, entender, navegar e interactuar con la web, aportando a su vez, contenidos. La accesibilidad web también beneficia a otras personas, incluyendo personas de edad avanzada que han visto mermadas sus habilidades a consecuencia de la edad (**W3C Accesibilidad Web - 2010**).

PAUTAS DE ACCESIBILIDAD WEB

El máximo organismo dentro de la jerarquía de Internet que se encarga de promover la accesibilidad es el World Wide Web Consortium (W3C), en especial, su grupo de trabajo Web Accessibility Initiative (WAI). Tal como ya hemos mencionado, una de las funciones de la Iniciativa de Accesibilidad Web (WAI) es desarrollar pautas y técnicas que proporcionen soluciones accesibles para el software web y sus desarrolladores. Las pautas de WAI son consideradas como estándares internacionales de accesibilidad Web (**Wikipedia Accesibilidad web - 2010**).

En 1999 el WAI publicó la versión 1.0 de sus pautas de accesibilidad Web. Con el paso del tiempo, se han convertido en un referente internacionalmente aceptado. En el 2008, las WCAG 2.0 fueron aprobadas como recomendación oficial.

Es esencial que los diferentes componentes de desarrollo Web e interacción trabajen conjuntamente para que la Web sea accesible para aquellas personas con discapacidad. Estos componentes incluyen:

1. **Contenido:** información presente en una página o en una aplicación web, incluyendo:
 - **información como por ejemplo texto, imágenes y sonidos.**

- **código o etiquetado que define estructura, presentación, etc.**
2. **Navegadores Web, reproductores multimedia** y otros "agentes de usuario".
 3. **Tecnología asistiva**, en algunos casos, lectores de pantalla, teclados alternativos, intercambiadores, software de escaneo, etc.
 4. Conocimiento de los **usuarios**, experiencias y, en ocasiones, estrategias de adaptación para la utilización de la Web.
 5. **Desarrolladores**, diseñadores, codificadores, autores, etc., incluyendo desarrolladores que tienen alguna discapacidad y usuarios que proporcionan contenido.
 6. **herramientas de autor:** software para crea sitios web.
 7. **herramientas de evaluación:** herramientas para evaluar la accesibilidad web, validadores de HTML, validadores de CSS, etc.

Los desarrolladores web normalmente utilizan herramientas de autor y herramientas de evaluación para crear contenido web.

Las personas utilizan navegadores web, reproductores multimedia, tecnologías asistivas u otros "agentes de usuarios" para obtener e interactuar con el contenido.

La Iniciativa de Accesibilidad Web (WAI) del Consorcio World Wide Web (W3C) desarrolla pautas de accesibilidad web para diferentes componentes:

- Pautas de Accesibilidad para Herramientas de Autor (ATAG), que tratan las herramientas de autor.
- Pautas de Accesibilidad al Contenido en la Web (WCAG), donde se tratan temas de contenido Web. Son utilizadas por desarrolladores, herramientas de autor y herramientas de evaluación de accesibilidad.
- Pautas de Accesibilidad para Herramientas de Usuario (UAAG), donde se habla de los navegadores web y reproductores multimedia, incluyendo algunos aspectos de tecnologías asistivas.

PAUTAS DE ACCESIBILIDAD AL CONTENIDO EN LA WEB (WCAG)

Los documentos denominados Pautas de Accesibilidad al Contenido en la Web (WCAG) explican cómo hacer que el contenido Web sea accesible para personas con discapacidad. El término "contenido" web normalmente hace referencia a la información contenida en una página web o en una aplicación web, incluyendo texto, imágenes, formularios, sonido, etc. **(Introducción a las Pautas de Accesibilidad al Contenido en la Web – 2010).**

Las WCAG son parte de una serie de pautas de accesibilidad, que incluyen las Pautas de Accesibilidad para Herramientas de Autor (ATAG) y las Pautas de Accesibilidad para Agentes de Usuario (UAAG).

Las WCAG están pensadas principalmente para:

- Desarrolladores de contenido web (desarrolladores y diseñadores).
- Desarrolladores de herramientas de autor para la web.
- Desarrolladores de herramientas de evaluación de accesibilidad web.

Tanto las WCAG como otros documentos complementarios tienen como objetivo satisfacer las necesidades de diferentes usuarios, incluyendo creadores de políticas, directivos y otros.

PAUTAS DE ACCESIBILIDAD PARA HERRAMIENTAS DE AUTOR (ATAG)

Los documentos denominados Pautas de Accesibilidad para Herramientas de Autor (ATAG) muestran cómo hacer que las herramientas de autor sean accesibles para personas con discapacidad. Estas herramientas, son software que se utiliza para crear páginas y contenido web. Uno de los objetivos principales de las ATAG es definir la forma en la que las herramientas ayudan a los desarrolladores web a producir contenido web que cumpla las Pautas de Accesibilidad al Contenido en la web **(Introducción a las Pautas de Accesibilidad para Herramientas de Autor – 2010).**

Las ATAG están pensadas principalmente para sus desarrolladores. Entre estas herramientas de autor se incluyen:

- Herramientas de edición específicamente diseñadas para producir contenido web, por ejemplo, editores HTML y XML de what-you-see-is-what-you-get (WYSIWYG).

- Herramientas que ofrecen la opción de guardar contenido en formato web, por ejemplo, procesadores de texto o paquetes de publicación.
- Herramientas que transforman documentos a un formato web, por ejemplo, filtros que transforman formatos de publicación a HTML.
- Herramientas que producen multimedia, especialmente cuando se quiere utilizar en la web, por ejemplo, producción de vídeo y edición, paquetes de autor de SMIL¹⁴.
- Herramientas para la administración o publicación de sitios web, incluidos los gestores de contenido (CMS), herramientas que automáticamente generan sitios web de forma dinámica desde una base de datos, herramientas de conversión instantánea y herramientas de publicación de sitios Web.
- Herramientas de diseño, por ejemplo, herramientas de formato CSS.

Tanto las ATAG como otros documentos complementarios están pensados también para satisfacer las necesidades de diferentes usuarios, como pueden ser directivos, responsables y otros.

PAUTAS DE ACCESIBILIDAD PARA AGENTES DE USUARIO (UAAG)

Los documentos de Pautas de Accesibilidad para Agentes de Usuario (UAAG) muestran cómo hacer que sean accesibles para personas con discapacidad, en especial, cómo incrementar la accesibilidad al contenido Web. Entre los agentes de usuario se incluyen navegadores, reproductores multimedia y tecnologías asistivas, software que algunas personas con discapacidad utilizan para interactuar con los dispositivos **(Introducción a las Pautas de Accesibilidad para Agentes de Usuario – 2010)**.

Las UAAG están pensadas para los desarrolladores de navegadores Web, reproductores multimedia, tecnologías asistivas y otros agentes de usuario.

Tanto las UAAG como otros documentos complementarios tienen como objetivo satisfacer las necesidades de usuarios diversos, creadores de políticas, directivos y otros. Por ejemplo:

- Aquellos usuarios que deseen elegir agentes de usuario más accesibles pueden utilizar las UAAG para evaluar los agentes de usuario.

¹⁴ Ver definición en Anexo B.

- Aquellos que por otro lado quieran animar a que los desarrolladores de agentes de usuario existentes mejoren la accesibilidad en versiones futuras, pueden indicar a los proveedores de agentes de usuario como referencia las UAAG.

Hasta aquí se mostraron las diversas leyes y normativas que existen actualmente para fundamentar el desarrollo de Tecnologías de la Información y la Comunicación Accesibles y/o Adaptables. Posteriormente, se continuará con la clasificación vista en el capítulo anterior. Particularmente, se propuso la separación de la clasificación en diferentes capítulos ya que es aquí donde se encuadran los conceptos teóricos para formar una base para el entendimiento de los llamados software de adaptación. Los cuales, basan su desarrollo en las normas explicadas en esta sección.

SOFTWARE DE ADAPTACIÓN

En esta sección, se hablará de algunos de los softwares de adaptación que se pueden encontrar y se clasificarán de la siguiente manera.

PROGRAMAS DE AUTOAYUDAS INFORMÁTICAS

Estos se definen como un grupo de programas o utilidades que se cargan en la memoria de la computadora antes que el programa principal. Su propósito es facilitar el uso de aplicaciones informáticas de tipo general a los usuarios con discapacidad que, de otra forma, se verían obligados a usar sólo programas específicamente diseñados para ellos. Si esto fuera así, el coste de fabricación sería muy elevado, trayendo consigo una menor integración del usuario en su ambiente cotidiano (**Sánchez Montoya - 2002**).

Las autoayudas informáticas ponen todos los programas a disposición de los usuarios con discapacidad, evitando así la necesidad de acudir a los programas exclusivos. Sin embargo, en muchos casos resulta necesario diseñar programas exclusivos que se adecúen a las necesidades específicas según la discapacidad.

Estos softwares tienen muchas aplicaciones. Algunos, hacen que la computadora trabaje más lento para que el usuario disponga de más tiempo y pueda responder adecuadamente; otros ofrecen redundancia visual o auditiva de salida y, de esta forma, las indicaciones de los programas pueden ser percibidas por los usuarios con deficiencias sensoriales. Las autoayudas intentan paliar, como ya hemos mencionado, la falta de previsión de algunos fabricantes de computadoras y programas que, al diseñar sus productos, sólo piensan en el usuario estándar, olvidando que existe una minoría que necesita pequeñas adaptaciones. Parece lógico pensar que, si son

personas quienes diseñan las máquinas para ponerlas a su servicio, sean éstas las que se adapten a sus circunstancias para permitirles el acceso a ellas.

Dentro de esta categoría se podrían ubicar los siguientes softwares de adaptación.

MPB - MOUSE POR BARRIDO

Es un software de distribución gratuita, que corre bajo Windows. La versión 3.0 de este programa fue liberada por Antonio Sacco¹⁵ (**Mouse por Barrido – 2010**).

Está pensado especialmente para personas que por problemas motrices no pueden utilizar un mouse convencional con precisión, y si bien el manejo del programa requiere cierto entrenamiento, los requerimientos mentales y de capacidad de abstracción son los mismos que un mouse común.

El objetivo del programa es permitir al usuario el manejo total del puntero del mouse mediante la sola presión de un switch (o pulsador) estándar. Trabaja en modo residente, sin interferir con la ejecución de otros programas, y está destinado a funcionar en conjunto con otras aplicaciones. Una vez que se ejecuta queda funcionando en modo residente.

Aparecerá una ventana mostrando alternadamente imágenes que indican las posibles acciones a ejecutar mediante el switch. Estas acciones consisten en: mover el puntero del mouse hacia arriba, abajo, izquierda o derecha; hacer clic, doble clic, presionar y mantener el botón izquierdo del mouse, liberarlo, o presionar el botón derecho.

El proceso que realiza el programa para ir mostrando diferentes imágenes alternadamente se denomina barrido. Para manejar el mouse, el usuario debe esperar que el programa muestre la opción correspondiente y, entonces, accionar el switch. La imagen quedará fija y el puntero del mouse se moverá hasta que el pulsador haya sido liberado.

¹⁵ Antonio Sacco - Graduado en Ingeniería en Sistemas de Información, en la Universidad Tecnológica Nacional. Especialista en Informática Aplicada en la Educación. Actualmente se encuentra finalizando Master Tecnología Informática Aplicada en la Educación en la UNLP. Antonio ha realizado grandes aportes en el área de tecnología aplicada a la educación nivel internacional. Algunos de ellos han sido mencionados en capítulos anteriores, como por ejemplo, Emuclíc, MPB (Mouse Por Barrido), entre otros. En <http://www.antoniosacco.com.ar/> puede encontrarse más información.

Este software incorpora opciones para configurar la velocidad con la que se moverá, el intervalo con que se alternarán las opciones, la posición de la ventana que presenta las opciones, entre otras cosas.

EMUCLIC

Convierte la acción de un switch (conmutador) estándar, o el sonido captado por un micrófono, en una o varias combinaciones de teclas o clics del mouse (**EmuClic – 2010**).

Se puede utilizar a través de un switch con una interfaz estándar conectada a la computadora en un puerto serie, o a través de un simple micrófono conectado a la placa de sonido.

El objetivo del programa es emular diversas funciones del teclado o del mouse, de manera que cada vez que se accione el switch o se genere sonido, la computadora interprete que se ha presionado determinada tecla, realizado un clic con el mouse, etc. De esta forma, cualquier programa que espere la presión de esa tecla reaccionará a la acción del conmutador o el sonido.

Trabaja en modo residente, sin interferir con la ejecución de otros programas, y está destinado a funcionar en conjunto con otras aplicaciones.

Este software fue desarrollado a partir del programa Switch to Click¹⁶, cuya primera versión ha sido diseñada e implementada en el año 2001. La versión 1.6 fue liberada en el 2007. Ambos software fueron desarrollados por Antonio Sacco.

El módulo de reconocimiento de sonido utiliza el TotiPmSpeechRecognitionEngine, creado por Jordi Lagares Roset¹⁷.

Su distribución es totalmente gratuita, y funciona correctamente en Windows 98, Me y XP (con Windows 2000, algunas computadoras pueden no reconocer el sonido).

KANGHOORU

Es un sencillo programa que ayuda a realizar un barrido de la pantalla. Además, es de licencia gratuita, por lo que es posible descargarlo y utilizarlo sin costo ni limitaciones (**Kanghooru – 2009**).

¹⁶ Switch to Click: <http://www.antoniosacco.com.ar/s2c.htm>

¹⁷ Jordi Lagares Roset: Profesor de Matemáticas de Enseñanza Secundaria, en el centro IES Santa Eugenia, de Girona.

Tiene por objetivo hacer pasar al mouse mediante barrido automático, por los lugares de la pantalla que se desee. Está pensado para que personas con discapacidad motora, puedan controlar que un programa realice determinada tarea al ser cliqueado en diferentes zonas de la pantalla. Para ser controlado con mayor comodidad se precisa de un pulsador externo conectado al click del mouse. Cuando se activa el barrido automático, el programa se minimiza y "desaparece". Al hacer clic con el botón derecho, el barrido vuelve a hacer aparecer la ventana del programa.

Específicamente, el programa tiene una opción para grabar la secuencia de barrido, pulsando sobre el botón "rec", de manera que el programa irá registrando los diferentes clics que se hagan sobre la pantalla, la captura finaliza pulsando nuevamente el botón derecho. Al ejecutar esta grabación, se mostrará en pantalla un recuadro rojo que irá saltando a cada clic elegido anteriormente, cuando sea la opción deseada se deberá pulsar el botón izquierdo o la tecla "Enter".

Existe además una versión multiplataforma para kanghooru, llamada JavaKanghooru. Esta versión tiene licencia GPL.



Figura 3.1 – Pantalla de control de grabación.

RATÓN FACIAL

Es un programa que, unido a una webcam USB estándar, permite controlar totalmente el ordenador sin las manos, mediante ligeros y suaves movimientos de alguna zona del cuerpo previamente configurada (**Ratón Facial – 2010**).

Se puede utilizar como sustituto o complemento del ratón convencional. Está especialmente diseñado para personas con discapacidades motrices en manos o brazos. Este software le proporciona al usuario una solución efectiva y fiable para la interacción con el ordenador.

El movimiento del puntero del mouse se realiza en coordinación con la zona del cuerpo a la que esté dirigida la webcam, y que previamente ha sido seleccionada (generalmente suele ser una parte de la cara). Las funciones del clic, ya sea doble o simple, del botón izquierdo o del derecho, se pueden elegir por uno de estos dos métodos:

- Deteniendo el puntero del ratón un determinado tiempo, configurado de antemano, en la zona sobre la que se quiere hacer clic.
- Mediante la emisión de un determinado sonido.

Funciona bajo Windows y posee licencia propietaria. Desde Internet se puede descargar una versión de prueba. Fue desarrollado por la empresa CREA y puede descargarse la versión gratuita desde <http://www.crea-si.com/esp/downloadfm.php>.

Como parte de este trabajo de tesina, se han realizado pruebas con algunos de estos programas de autoayuda, de manera tal de evaluar su posible utilización conjunta con algunas herramientas de autor, para la generación de actividades educativas. En particular, se ha utilizado JClíc debido a su amplia difusión en el ámbito educativo. Se presentan en la siguiente sección algunos de los resultados de estas pruebas.

ALGUNAS PRUEBAS DE INTEGRACIÓN CON JCLIC

Para llevar a cabo estas pruebas de integración, se generó una actividad JClíc de prueba, la cual consta de una asociación simple donde se presentan dos grillas: una con sumas y otra con los resultados. El usuario deberá unir adiciones a su respectivo resultado. La forma de resolver esta actividad es la siguiente: el usuario hace clic en una celda, mueve el mouse hacia una celda de la otra grilla y hace clic, si la unión que eligió es correcta (la suma que se muestra en la celda de una de las grillas da como resultado el número que se muestra en una celda de la otra grilla), JClíc marca esas celdas para que no se puedan volver a elegir y otorga un punto de acierto al usuario. Se debe repetir esta serie de pasos hasta que todas las celdas se unan quedando resuelta la actividad.

INTEGRACIÓN DE EMUCLIC CON JCLIC

Según las pruebas realizadas, no es posible resolver la actividad creada con Emuclíc. Esta integración, no se ha podido llevar a cabo, pues Emuclíc sólo permite emular una sola función de teclado o mouse por vez, en otras palabras, si por ejemplo se acepta que cuando el programa registre un sonido se realice un clic izquierdo, esto se hará para cada sonido que el programa perciba. Esta situación, no permite configurar otras funciones de manera simultánea, de forma tal de permitir al usuario hacer clic en una celda de la grilla y mover el mouse hasta la celda par para hacer luego clic en ella y poder resolver la actividad.

INTEGRACIÓN DE MPB CON JCLIC

Fue posible realizar la actividad utilizando MBP, ya que este software provee el manejo completo de la funcionalidad de un mouse a través de un switch. Permite seleccionar las celdas necesarias, a través de las pulsaciones del switch, para resolver la actividad. Cuando el programa residente muestra la imagen de la funcionalidad que se necesita (por ejemplo, clic con el botón izquierdo) se debe pulsar el switch. Para el caso de esta actividad, se requería poder hacer clic izquierdo y mover el mouse en las 4 direcciones posibles (izquierda, derecha, arriba y abajo). Esto se logra al presionar el switch cuando aparece la imagen de la funcionalidad deseada y a continuación el mouse ejecuta la orden que representa la imagen mostrada.

En caso de que se necesite el uso del teclado para resolver alguna actividad, como ocurre con las actividades de texto, respuesta escrita y palabras cruzadas de JClíc, se podría combinar MPB con un teclado en pantalla.

La desventaja o dificultad que se experimentó mientras se resolvía la actividad de prueba fue que resultó tediosa la espera de la imagen que se quería ejecutar. Además, si por ejemplo se detenía el mouse a muy poca distancia de la celda que se quería presionar, avanzar esa mínima distancia resultaba muy difícil, y era más sencillo alejarse de ella y volver a intentar posicionar el cursor con otro movimiento.

INTEGRACIÓN CON RATÓN FACIAL

Es posible realizar actividades con JClíc a través del uso de ratón facial, ya que este software provee el manejo del mouse, a través de una cámara web que lo reemplaza, pudiéndose realizar dichas actividades sin problemas. Las pruebas se realizaron configurando el uso de clic por espera, es decir, que se debe mantener el mouse sin moverse por cierta cantidad de tiempo para que el ratón ejecute el clic. Esta cierta cantidad de tiempo es configurable desde la aplicación, en esta prueba se seleccionó un tiempo de espera de 2 segundos.

Para resolver la actividad, se debió mover el mouse a través de movimientos de la cabeza hasta ubicar el cursor sobre la primera celda que se deseaba seleccionar. Una vez ubicado allí, se debió esperar dos segundos para que el ratón facial ejecute un clic.

Antes de ejecutar el clic, el programa marca con un círculo la sección donde se va a realizar. Se debió, entonces, mantener la cabeza quieta para ejecutar el clic. Se tuvo que tener cuidado, porque si por alguna razón la cámara web detecta el movimiento de la cabeza, el círculo se hubiera borrado y no se hubiera ejecutado el clic. Cabe destacar que este círculo tarda 2 segundos para dibujarse que es el tiempo de espera configurado.

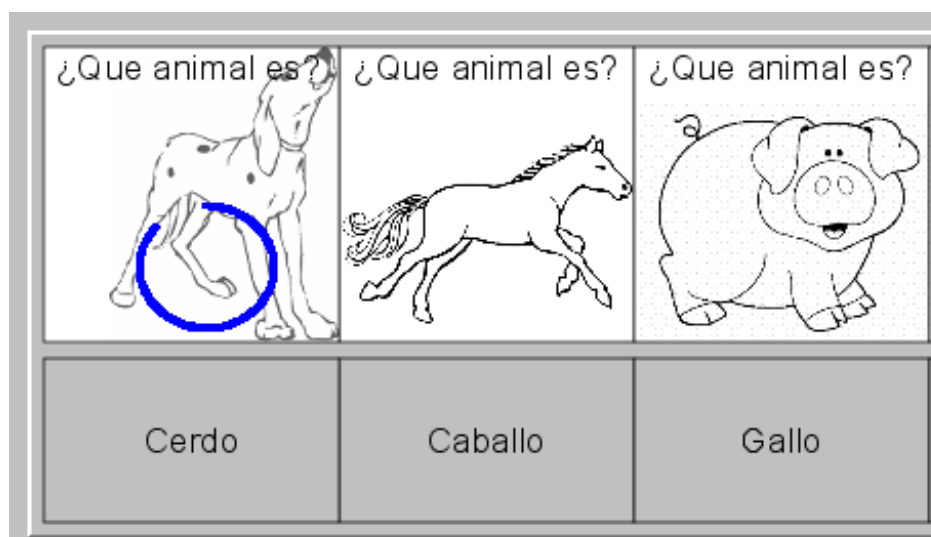


Figura 3.2 – Marca del ratón facial para indicar la sección donde se realizará el clic.

Luego se debió dirigir el mouse a la siguiente celda (con movimientos de la cabeza), con la que deseaba unir la celda anterior y se esperó nuevamente el tiempo establecido para ejecutar el clic.

De esta manera, se procedió para unir todas las celdas hasta resolver toda la actividad.

El tiempo que se necesita para resolver la actividad es bastante mayor al que llevaría resolverla sin ninguna adaptación. En principio se notó que la precisión del ratón depende de la cámara web utilizada y de la calidad del video, por lo que a veces puede dificultar su utilización.

Al igual que con MPB, en caso de actividades que necesiten del teclado, se puede combinar el uso del ratón facial con un teclado virtual.

CONCLUSIÓN

Si bien es posible utilizar algunos de estos softwares junto con JClic, se vio al reconocimiento de voz como una alternativa viable, la cual apunta al uso del habla cuando sea posible para el usuario, siempre y cuando éste se desenvuelva oralmente sin dificultades o con dificultades leves. Se considera importante destacar, que es necesario introducirse en el área de reconocimiento de voz para facilitar el uso de la computadora para las personas con problemas motrices, ya que con el uso de la voz se evita el esfuerzo físico que se genera al intentar usar ayudas técnicas como el switch.

PROGRAMAS EXCLUSIVOS SEGÚN LA DISCAPACIDAD

Se incluyen en este grupo, programas pensados y diseñados para personas con una determinada discapacidad. No se trata de adaptar un programa de tipo general a las necesidades de una persona sino de elaborar un programa para unas necesidades específicas.

Dentro de este grupo están los programas para el aprendizaje y uso de los Sistemas Aumentativos y Alternativos de Comunicación, traducir los textos en tinta a braille, acceder a la computadora mediante el uso exclusivo de conmutadores, rehabilitación del habla, etc. **(Sánchez Montoya - 2002)**.

PLAPHOONS

Es un programa de comunicación pensado para personas con discapacidad motora que no se puedan comunicar mediante el habla. Tradicionalmente las personas con estas discapacidades se comunican mediante la ayuda de otra persona y un libro de símbolos, llamado plafón de comunicación **(Plaphoons – 2010) (Utilización del programa Plaphoons – 2010)**.

Este programa es una adaptación en sí mismo, ya que fue pensado desde un principio como un medio de comunicación aumentativa y alternativa que permite a los usuarios comunicarse con el medio, a través del uso del tablero de comunicación. Permite crear plafones y a partir de estos, estructurar los símbolos para crear mensajes. Estos mensajes, pueden ser visualizados directamente en pantalla, ser impresos y/o ser escuchados mediante voz sintetizada o digitalizada.

Este software tiene más de una utilidad. Por un lado, se puede utilizar directamente como plafón de comunicación de ordenador. Por otro lado, se puede utilizar como un comunicador, utilizando las capacidades de síntesis de voz y/o voces grabadas.

En cualquiera de sus dos funciones se puede interactuar de forma directa (pantalla táctil, mouse) o con barrido automático mediante cualquier tipo de adaptación (multimouse, joystick, conmutador). El barrido automático, alterna el foco en los diferentes casilleros, para que el alumno pueda ir seleccionándolos con un sólo clic. Primero barre a nivel de filas y luego de seleccionada una fila, comienza a recorrer la fila celda por celda.

El programa actúa de dos formas distintas, una forma es en modo usuario y otra en modo edición. En modo usuario se permite crear mensajes (y todas las preclusiones del programa) a partir de plafones previamente editados. Esta funcionalidad, muestra el tablero de comunicación con las opciones de utilizarlo para crear mensajes y poder comunicarlos utilizando el sintetizador de voz, provisto por el programa, o a través de

mensajes previamente grabados por otra persona asociados a una imagen o palabra que se mostrará en el tablero.

En modo edición, se pueden crear tableros personalizados con símbolos definidos por el usuario, o bien símbolos blyss¹⁸ o símbolos SPC (tipo Boardmaker)¹⁹.

La finalidad principal de este programa es dar más autonomía a las personas con trastornos en el habla que impiden que se puedan comunicar oralmente con el entorno. Este programa, permite a dichas personas que construyan sus mensajes para expresar deseos, sugerencias u opiniones si así lo desean.

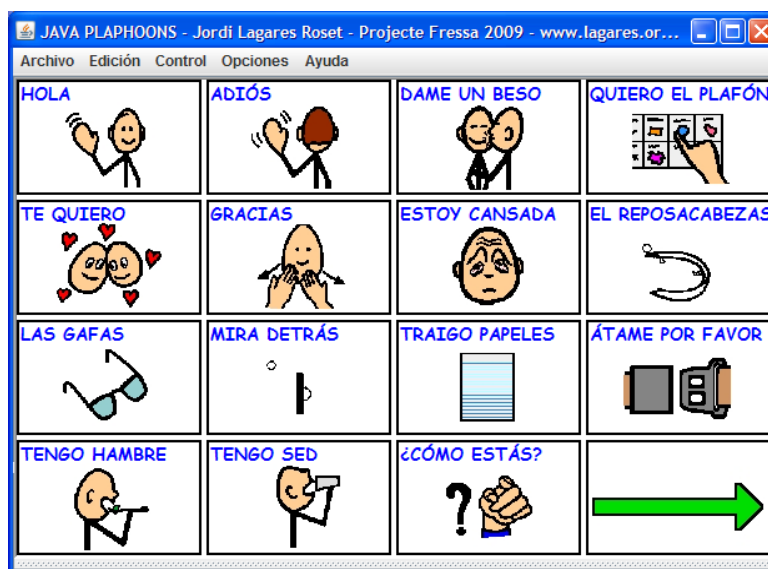


Figura 3.3 - Ejemplo de tablero de símbolos.

Existen dos versiones de este programa, una llamada Plaphoons y otra llamada JavaPlaphoons con similares características funcionales. JavaPlaphoons es una versión posterior a Plaphoons desarrollada enteramente en java.

El programa Plaphoons sólo puede ejecutarse en ambientes Windows, aunque esta limitación fue superada con su posterior versión en java, que debido al lenguaje en el cual está desarrollado, es independiente de la plataforma.

Plaphoons es un programa de distribución gratuita desarrollado por Jordi Lagares Roset que se puede descargar libremente desde www.xtec.es/~jlagares/download/plaphoons.zip. En el caso de JavaPlaphoons el programa además es de código abierto.

¹⁸ Ver definición en Anexo B.

¹⁹ Ver definición en Anexo B.

UN JUEGO DE GRAVEDAD CON RECONOCIMIENTO DE VOZ PARA NIÑOS CON PROBLEMAS DE LENGUAJE

El juego de Gravedad se desarrolló utilizando el lenguaje C y las librerías gráficas SDL²⁰ y los modelos acústicos se desarrollaron utilizando el software de reconocimiento de voz HTK²¹ (**Un juego de gravedad con reconocimiento de voz para niños con problemas de lenguaje – 2010**).

Este juego está diseñado para utilizarse en niños en edad escolar que tengan problemas del habla, como herramienta de rehabilitación.

La historia del juego se desarrolla en una casa, y el ambiente es la cocina de la misma, asimismo hay 4 personajes que el niño puede elegir. Este personaje, ayuda al niño a vencer los retos que un alux (personaje mitológico de la cultura Maya) pone durante la interacción con el juego. El cual, encanta objetos, haciéndolos volar y tratando de romperlos con su resortera con la intención de ensuciar todo a su paso. Si la mamá llega y encuentra la cocina desordenada, los culpará y no los dejará salir a jugar al patio. Entonces el alux habrá logrado su objetivo.

Para desencantar los objetos y colocarlos en su lugar, es necesario que el personaje se coloque debajo de los objetos y pronuncie su nombre correctamente, antes de que estos sean alcanzados por la resortera del alux provocando que caigan al piso rompiéndose y/o ensuciando la cocina.

A medida que va transcurriendo el juego, los objetos van cambiando de color, pasando por el verde, amarillo y rojo. Cuando el jugador pronuncia correctamente el nombre del objeto que eligió, este objeto desaparece de la pantalla y entonces el sistema le otorga una puntuación de acuerdo al color del objeto en ese momento. En caso de que el jugador no logre pronunciar el nombre del objeto, y este tiene el color rojo, el objeto es golpeado por la resortera del alux y cae rompiéndose y/o ensuciando el piso de la cocina, en este caso el sistema no le asigna puntuación al jugador.

Este software fue desarrollado por integrantes de la Universidad Autónoma de Yucatán (México), actualmente superaron las pruebas realizadas en escuelas, pero la versión final de este producto aún no fue liberada.

Existen otros softwares de este estilo, entre ellos, un Juego de Memorama con reconocimiento de voz para niños, el cual es muy similar al Juego de Gravedad con reconocimiento de voz para niños con problemas de lenguaje. Sólo se mencionan dos software a modo de ejemplo, para lograr una mejor comprensión de esta categoría.

²⁰ Ver definición en Anexo B.

²¹ Ver definición en Anexo B.

¿QUÉ SE ENTIENDE POR TECNOLOGÍAS DE AYUDA?

Las tecnologías de ayuda o ayudas técnicas son utensilios, dispositivos, o equipos de diferente grado de complejidad, fabricados, y comercializados o de manufactura artesanal, utilizables por o para personas discapacitadas, con el fin de suplir o complementar su limitación o sus carencias funcionales; en otras palabras, serían todo aparato o utensilio que a una persona discapacitada le sirva para conseguir un grado de independencia, una mejora de la calidad de vida cotidiana o profesional, que le proporcione una mayor autonomía y confort **(Ayudas Técnicas, habilitación y diversidad – 2010)**.

Hay determinados alumnos que presentan necesidades educativas especiales asociadas a déficit sensorial o motor y que necesitan de ayudas técnicas específicas para acceder al ordenador. Se entiende, pues, a estas tecnologías como recursos para ayudarles a superar las necesidades específicas que puedan presentar y mejorar su calidad de vida.

Aunque están relacionadas preferentemente con la discapacidad y las barreras de accesibilidad, el uso de las ayudas técnicas no solo está limitado a estas situaciones. Así pues, aquella persona que intente hablar por un teléfono móvil en un entorno ruidoso se encontrará con una situación similar a la que experimenta una persona con dificultades auditivas. Otro ejemplo puede ser el caso de la persona que intenta hacer uso de un ordenador portátil en un sitio donde la visibilidad es casi nula, causada, por ejemplo, por el humo. En esta última situación, mediante la ayuda técnica de la síntesis de voz sería posible usar correctamente el ordenador **(Wikipedia Ayuda Técnica – 2010)**.

Para la creación de ayudas técnicas, se usan conceptos y conocimientos extraídos de la ergonomía y otras disciplinas, con el objetivo de que el producto resultante sea adecuado a las características del usuario y le permita compensar su limitación en el acceso a entornos o productos.

En el capítulo 4, se verán algunos ejemplos disponibles para facilitar el acceso a personas con dificultad motriz.

CONCLUSIÓN

En estos últimos años se puede observar que se han producido importantes avances en la integración social de las personas con discapacidad. En la actualidad, se encuentra a disposición una gama de adaptaciones mucho más amplia, las cuales comienzan a abarcar las distintas discapacidades que existen, pero se debe tener en

cuenta que estas discapacidades no se suceden con la misma frecuencia, por lo que se encuentran más ayudas, información y adaptaciones para algunas que para otras. Esto se desprende del hecho de que se necesita un entorno que se especialice en determinada discapacidad o imposibilidad para que se logren herramientas de adaptación y accesibilidad con un nivel de madurez que permita que sean útiles al usuario final.

Las TIC son herramientas que pueden utilizarse de forma creativa para mejorar el desarrollo de capacidades y destrezas, de las personas con necesidades especiales bajo una concepción interaccionista, que desplaza su enfoque desde las características individuales de los alumnos, a un modelo de apoyo curricular que actualmente se encuentra en proceso de expansión (**Sánchez Montoya - 2002**).

Los principales obstáculos que se deben vencer al momento de pensar en una solución informática para personas con necesidades especiales, son las dificultades que estos encuentran al momento de comunicarse con el entorno que los rodea, sin depender de otras personas.

Es importante destacar que a través de estas adaptaciones, no sólo se obtiene como beneficio incrementar las posibilidades de las personas con discapacidad para comunicarse, sino que además se logra el acceso a la educación, control de su entorno, integración laboral, ocio; y que cuando se suman todos estos factores se aproxima a la independencia que toda persona necesita. En el próximo capítulo, se analizarán conceptos y características de la discapacidad motora, en la cual se centra el desarrollo del presente trabajo. Es interesante conocer cuáles son las principales patologías que producen esta discapacidad, y cuáles son sus dificultades de desarrollo a nivel de lenguaje, a nivel cognitivo y a nivel social.

Estas características se estudiarán durante el desarrollo del siguiente capítulo.

CAPÍTULO 4

LA PARTICULARIDAD DE LAS
PERSONAS CON PROBLEMAS
MOTRICES Y AYUDAS TÉCNICAS

INTRODUCCIÓN

Como en todos los casos en los que se habla de personas con limitaciones físicas, psíquicas o sensoriales, es posible adoptar diferentes ópticas y formas de tratamientos. En el ámbito educativo, se ha pasado desde una visión basada en el déficit o deficiencias orgánicas a otras de naturaleza psicosocial y educativa, que toma como referencia esencial las necesidades educativas que tiene esa persona en un medio escolar y social concreto, aprovechando al máximo sus posibilidades.

Desde este punto de vista, el problema que se plantea es la comprensión de los factores que intervienen en la discapacidad motora y cómo esta influye en el proceso de desarrollo y aprendizaje. A partir de aquí, habrá que decidir cómo debe ser la educación de los niños afectados con el fin de ayudarles a llegar al máximo de sus potencialidades.

Discapacidad es una palabra amplia, que intenta definir una situación de desventaja para un individuo, define las limitaciones que tiene la persona. Esta limitación, estará caracterizada no sólo por el tipo de deficiencia que posea, sino por la comunidad a la que pertenezca (**Discapacidad - 2010**).

En el presente trabajo, interesa conocer brevemente las diferentes patologías en las que se puede presentar discapacidad motora, haciendo referencia a distintos factores que pueden obstaculizar su desarrollo psicosocial y de aprendizaje. De relación con ello, se propone el uso de la tecnología como medio para disminuir la brecha entre la persona afectada y la sociedad.

Cabe aclarar, que para el desarrollo del presente capítulo se ha consultado a profesionales abocados al trabajo con personas con características de dificultades motoras. Particularmente se ha consultado con la Licenciada Rosalía Moralejo²².

DISCAPACIDAD MOTRIZ

La discapacidad motora es la alteración en la funcionalidad de las extremidades superiores y/o inferiores. Dicha alteración, se produce por una lesión en el sistema nervioso central (SNC) a nivel de médula o del encéfalo. Si la lesión acontece en esta parte del sistema nervioso, su consecuencia es la parálisis cerebral. Las causas pueden

²² Licenciada en Fonoaudiología, graduada en la Universidad Nacional de San Luis. Actualmente se encuentra realizando el Doctorado en Fonoaudiología. Ha trabajado con niños con discapacidad motora especialmente niños con Parálisis Cerebral.

ser perinatales (durante el embarazo), paranatales o postnatales²³. Este conjunto de discapacidades recoge la limitación del normal desplazamiento físico, o de la movilidad general. En lo concerniente a la restricción en la capacidad de desplazamiento, se puede señalar a grandes rasgos, que las personas que tienen este tipo de discapacidad pueden ser semiambulatorias o no ambulatorias. En el caso del primer tipo, se desplazan ayudadas por elementos complementarios, bastones, andadores, etc. Las no ambulatorias sólo pueden desplazarse con silla de ruedas.

Según la causa de la discapacidad, y el grado de ésta, pueden existir además restricciones para el desarrollo del lenguaje, o de la manipulación de objetos, siendo oportuno señalar que este tipo de restricciones no tiene por qué afectar a los procesos cognitivos o de aprendizaje, aunque en ocasiones, puede presentarse un entecimiento en algunas actividades y habilidades, como la comunicación, la lectoescritura, o el desarrollo de actividades de autocuidado²⁴.

La discapacidad motriz podría definirse como la pérdida o restricción de la capacidad de movimiento, desplazamiento y equilibrio de todo o de una parte del cuerpo. Se entiende como la alteración o deficiencia orgánica del aparato motor o de su funcionamiento, que afecta al sistema óseo, articulario, nervioso y muscular. Las personas afectadas, presentan una clara desventaja en su aparato locomotor, limitaciones posturales, de desplazamiento, coordinación y manipulación (**Deporte Adaptado - 2010**).

A nivel educativo, se plantea el problema en cuanto a la intervención centrada no en el déficit que presenta el alumno, sino en sus posibilidades de aprendizaje y sus potencialidades educativas.

Se mostrarán diferentes formas de clasificar los trastornos que pueden resultar en algún tipo de discapacidad motora. En esta clasificación se tiene en cuenta la lesión que originó la limitación motora, la etiología y la parte del cuerpo que se ve afectada.

Además, se hará una breve descripción de cuáles son síntomas que se pueden presentar en una persona con déficit motor y cuáles son las consecuencias/limitaciones con las que se encuentra durante el desarrollo de su vida cotidiana.

²³ Discapacidad Motora - <http://disfonema.wordpress.com/category/discapacidad-motora/>

²⁴ Guía de Apoyo a Personas con Discapacidad -

http://www.urjc.es/comunidad_universitaria/universidad_saludable/programa_discapacidad.html

CLASIFICACIÓN SEGÚN EL ORIGEN

En este caso, se tiene en cuenta, cuál fue la lesión que originó la deficiencia.

- Origen cerebral
 - Parálisis Cerebral.
 - Traumatismos craneoencefálicos.
 - Hidrocefalia.
- Origen espinal
 - Poliomiелitis aguda.
 - Espina Bífida.
 - Lesiones Medulares Degenerativas.
 - Traumatismos Medulares.
- Origen Muscular
 - Miopatías.
- Origen osteo-articular
 - Reumatismos de la infancia.
 - Lesiones osteoarticulares por desviaciones de la columna.

CLASIFICACIÓN SEGÚN SU ETIOLOGÍA.

- Trasmisión genética: madre portadora; uno de los progenitores tienen el mismo hándicap; los dos padres son portadores recesivos de una misma enfermedad genética.
- Infecciones microbianas: Tuberculosis ósea, Poliomiелitis anterior aguda.
- Accidentes: en el embarazo o parto; a lo largo de la vida.
- De origen desconocido: Espina bífida, Escoliosis ideopática.

CLASIFICACIÓN SEGÚN LOCALIZACIÓN TOPOGRÁFICA.

- Parálisis: Monoplejía, Hemiplejía, Paraplejía, Diplejía, Tetraplejía.
- Paresia: Monoparesia, Hemiparesia, Paraparesia, Tetraparesia.

¿CÓMO SE DETECTA LA DISCAPACIDAD MOTORA?

El diagnóstico del niño con discapacidad motora es muy difícil en lactantes menores de 4 meses e inclusive en los de 6 meses si el problema es leve. Los signos consisten principalmente en retardo del desarrollo motor y persistencia de reacciones primitivas. En los casos más leves, a veces el diagnóstico se hace a los 18 meses cuando el niño debería caminar.

A medida que este se torna más activo, van instalándose posturas y movimientos anormales, que cambian según las adapte a sus actividades funcionales. Estos cambios se operan de acuerdo con líneas previsible, pero difieren entre los diversos tipos de parálisis cerebral: cuadriplejías y diplejías espásticas o hemiplejías y atetosis **(Discapacidad Motriz - 2010)**.

CONSECUENCIAS DE LA DISCAPACIDAD MOTORA EN EL DESARROLLO DE LA PERSONA.

Los niños que presentan discapacidad motora manifiestan alteraciones psicomotrices en su desarrollo, derivadas directa o indirectamente de su trastorno neuromotor. La mayor parte de las habilidades que adquiere un niño a lo largo de su desarrollo, tienen un componente motor. Por lo que resulta obvio, que dicha discapacidad alterará estas habilidades. Dependiendo de la gravedad de la lesión las adquirirá o bien más tarde, de forma anómala o defectuosa o incluso, es posible que no las adquiera **(Propuestas educativas ante el alumno con discapacidad motora - 2010)**.

Por otra parte, las disfunciones motoras afectan a aspectos de la vida del individuo; limitan sus experiencias; sus posibilidades de aprender y su forma de relacionarse con otras personas. Por lo que su discapacidad afectará indirectamente el curso de su desarrollo.

CONSECUENCIAS EN EL DESARROLLO DEL LENGUAJE

La comunicación oral, para que sea eficiente, requiere la secuencia ordenada y la coordinación de tres procesos fundamentales²⁵:

1. La organización de los conceptos, su formulación simbólica y su expresión.
2. La exteriorización del pensamiento por el habla, con la intervención de las funciones motrices concurrentes: respiración, fonación, resonancia, articulación y prosodia.
3. La programación de estas destrezas motrices en la producción voluntaria de los sonidos individuales del habla y su combinación en el orden necesario para formar las palabras.

Por lo tanto, las alteraciones motrices pueden afectar directa o indirectamente en el desarrollo del lenguaje provocando trastornos de la fonación, de la articulación de las palabras y en la estructura del lenguaje, y pueden evidenciarse desde el nivel prelingüístico afectando las funciones neurovegetativas como son succión, deglución, respiración, y más tarde masticación y juego vocal o balbuceo. Es decir que las alteraciones motoras pueden afectar principalmente el aspecto motor-expresivo del lenguaje, así como también, el control de los órganos fonos articulatorios en sus funciones no lingüísticas, como son la alimentación, la comunicación gestual y el movimiento voluntario.

Las consecuencias en el lenguaje son variables, desde un trastorno articulatorio leve hasta la ausencia completa del lenguaje, donde es necesario desarrollar métodos de comunicación aumentativa y alternativa para permitir la comunicación de la persona. Teniendo en cuenta que el interés de este trabajo, es pensar en una adaptación de comandos por voz, parece interesante analizar en este punto el tipo de comandos a utilizar. Como se explico anteriormente, las personas con discapacidad motora pueden presentar también problemas en la articulación de las palabras. Por lo tanto, es conveniente referirnos brevemente a las características articulatorias de los sonidos del habla, con el fin de tenerlo en cuenta a la hora de elegir los sonidos o comandos propuestos en este trabajo.

²⁵ Alteraciones motrices del habla, Darley, F.; Aronson, A.; Brown, J., 1978.

PROCESOS MOTORES BÁSICOS EN LA ADQUISICIÓN DEL HABLA

En la ejecución del habla intervienen diferentes partes del organismo conectadas entre sí. El aparato del habla comprende 9 componentes funcionales. Estos componentes son:

1. Músculos abdominales y diafragma.
2. Músculos, huesos y cartílagos de la pared del tórax.
3. Músculos y cartílagos laringeos laríngeas.
4. Músculos de la mitad inferior de la faringe y parte posterior de la lengua que dan forma a la cavidad faríngea.
5. Músculos palatofaríngeos.
6. Músculos que regulan los movimientos de la parte media y anterior de la lengua.
7. Maxilar inferior y músculos correspondientes.
8. Músculos faciales utilizados para abrir y cerrar la boca y efectuar los movimientos de retracción y apertura de los labios.

Cada uno, actúa como una válvula de escape para la corriente de la respiración empleada para el habla. Una columna de aire en movimiento, impulsada por los músculos torácicos, forma el habla por la acción de una serie de válvulas. La producción del habla puede considerarse como una imposición, sobre la corriente de la respiración, de la influencia de una serie de válvulas músculo-esqueléticas:

- Válvula glótica: formada por las cuerdas vocales, su aducción permite la producción de un tono vocalizado y su abducción permite el paso del aire ininterrumpido o mudo. La laringe hace vibrar la columna de aire en movimiento, produciendo diversos aspectos del tono, intensidad y calidad de la fonación.
- Válvula palatofaríngea: constituida por los músculos del velo del paladar y de la faringe. Conecta o desconecta la cavidad nasal con la bucal, originando modificaciones en las cualidades vocales y resonancia.
- Válvula linguovelar: formada por la parte posterior de la lengua y el velo. Junto con las válvulas siguientes, fragmenta la columna de aire en movimiento en el lenguaje articulado.
- Válvula linguopalatina: el borde de la lengua se conecta con el paladar duro.

- Válvula linguoalveolar: la punta de la lengua toca el borde alveolar.
- Válvula linguodental: la punta de la lengua hace protrusión entre los dientes, tocándolos.
- Válvula labial: los incisivos superiores y el labio inferior.

Las diversas válvulas interrumpen, traban y comprimen la corriente de aire de innumerables maneras, para producir el repertorio completo de la voz y los fonemas del habla.

La respiración, proporciona la materia prima para el habla. El ciclo inspiración-espирación de la respiración vegetativa se altera durante el habla, el cual es una prolongación interrumpida de la parte espiratoria.

Durante la fonación, el aire de la respiración provoca la vibración de las cuerdas vocales de la laringe en aducción. La corriente respiratoria con componentes periódicos y aperiódicos, debe ser modelada y modificada por medio de la resonancia y la articulación. La resonancia, es la amplificación selectiva del tono vocal, la faringe, la cavidad bucal y la cavidad nasal sirven como resonadores que refuerzan ciertos componentes del tono. Si la cavidad nasal se junta con otras cavidades, el resultado es un tono nasal definido. Los músculos constrictores de la parte superior de la faringe y los elevadores del paladar blando inician juntos la conexión y desconexión que inducen tales cambios en la resonancia. Las modificaciones de las características de la resonancia de la cavidad bucal, se obtienen en la posición diferencial de la lengua y el maxilar inferior y con las alteraciones en el orificio por adaptaciones mandibulares y cambios en la apertura de los labios.

Por último, la apertura respiratoria es conformada en fonemas (articulación) por medio de impedancias producidas por los diversos articuladores: lengua, dientes y labios.

Durante la función articuladora normal, la impedancia o interrupción de la corriente respiratoria es, a veces, completa, como en el caso de las consonantes *plosivas* (*p, b, t, d, k, y g*), que se producen con una detención completa de la corriente de aire, una acumulación de presión, y una súbita liberación de presión y explosión de la respiración. Para los fonemas *fricativos* (*f, v, s, z*), la detención de la corriente respiratoria es menos completa, pero hay una constricción suficiente de la misma como para crear un sonido de fricción. En la formación de los *africados* (*g y j*), la lengua adopta una posición para una plosiva, pero el aire es expulsado a través de la constricción características de los fricativos. Para producir los sonidos de transición (como en inglés *wh, w, y*), se requiere una impedancia del aire bastante escasa; el atributo distintivo de estos fonemas es que varían en los aspectos acústicos y fisiológicos en el transcurso de su duración. Las *semivocales* (*r y l*), requieren asimismo

escasa impedancia de la corriente respiratoria y se asemejan a las *vocales*, que son sonidos vocalizados abiertos, relativamente ilimitados en su egreso por las válvulas bucales. La producción de las consonantes *nasales* (*m, n, y ñ*), supone la integración de la cavidad nasal como resonador; la emisión de cada sonido es nasal, mientras que en todos los demás fonemas es oral.

Algunos de los fonemas consonánticos son sordos, producidos sin activación de las cuerdas vocales, en tanto que otros se caracterizan por un tono vocalizado, producido por la vibración de las cuerdas vocales en aducción. Las consonantes también varían con respecto al volumen de presión de aire intraoral requerida para su correcta producción. Las que requieren presión máxima son las africadas, fricativas y plosivas, por eso se las denomina consonantes a presión. Las consonantes sordas precisan más presión que las vocalizadas.

PUNTO DE ARTIC.	MODO DE ARTICULACION											
	Plosivas		Fricativa		Africada		Nasal		Transición		Semivocal	
1: Muda 2: Vocalizada	1	2	1	2	1	2	1	2	1	2	1	2
Bilabial	<i>p</i>	<i>B</i>						<i>m</i>	<i>Wh</i>	<i>w</i>		
Labiodental			<i>f</i>	<i>v</i>								
Linguodental			<i>th</i>	<i>th</i>								
Linguoalveolar	<i>t</i>	<i>D</i>	<i>s</i> <i>sh</i>	<i>z zh</i>	<i>ch</i>	<i>j</i>		<i>n</i>				<i>l</i>
Linguopalatal										<i>y</i>		<i>r</i>
Linguovelar	<i>k</i>	<i>G</i>						<i>ng</i>				
Glotal			<i>h</i>									

Tabla 5.1 – Clasificación de los fonemas consonánticos de acuerdo con el punto y modo de articulación y vocalización.

Es importante aclarar, que los problemas de lenguajes no se dan en todos los casos, sólo es una dificultad que se puede relacionar a los problemas motores. Cabe destacar, que el desarrollo propuesto en el presente trabajo, está orientado a personas sin dificultades en el desarrollo del lenguaje o con dificultades leves.

CONSECUENCIAS EN EL DESARROLLO COGNITIVO

Las alteraciones cognitivas van asociadas, en la mayoría de los casos, a la discapacidad motriz. Algunas facetas que se ven afectadas son algunas funciones cognitivas complejas, como la atención, la concentración y la coordinación (**Propuestas educativas ante el alumno con discapacidad motora – 2010**).

Estas dificultades, en el ámbito cognitivo, se ven influenciadas significativamente por las situaciones de cansancio, estrés y fatiga que afectan en mayor medida a los alumnos con discapacidad motriz.

Además, este déficit motor puede alterar las posibles experiencias de la persona en relación al mundo físico como social, incluso puede afectar su sentido de autoeficacia, motivación y disposición para el aprendizaje.

Es evidente, que estos niños encuentren dificultades para manipular, controlar y explorar libremente el entorno físico. Esto va a impedir un desarrollo de la inteligencia sensoriomotriz, razonamiento operatorio y formal.

Por otra parte, las alteraciones motoras interfieren en la adquisición de otras habilidades que suponen pilares básicos para el intercambio cultural, intervención educativa y para las prácticas instrumentales.

A nivel educativo, los sistemas y ayudas técnicas para la comunicación aumentativa y alternativa constituyen un recurso fundamental, que unidos a los recursos metodológicos, pueden garantizar un correcto desarrollo cognitivo.

CONSECUENCIAS EN EL DESARROLLO SOCIAL

El déficit motor conlleva una interacción anómala con el mundo social. La motricidad reducida o poco controlada provoca una interacción alterada, porque el niño no puede producir mucho de los gestos, a los cuales la sociedad le asigna valor comunicativo. El niño encuentra dificultad en ganar y mantener la atención de otras personas, transmitir e intercambiar informaciones y afectos, falta de control sobre los objetos, etc. Todo ello, lleva al sujeto a un aprendizaje activo de falta de sincronía entre sus respuestas y las consecuencias sobre el ambiente, a esto lo denominan algunos autores como el fenómeno de la indefensión aprendida o también, faltas de expectativas de auto eficacia (**Propuestas educativas ante el alumno con discapacidad motora – 2010**).

La experiencia repetida de fracaso provoca frustración. Si al fracaso se une la sobreprotección, el fallo motivacional puede agravarse. Por otra parte, si el niño se enfrenta con frecuencia a situaciones que no puede resolver, puede quedar afectada su disposición para el aprendizaje. Existen autores que se basan en la idea de que la inteligencia no podrá manifestarse mientras el niño piense o crea que sus acciones no tendrán efecto, por lo tanto, el rendimiento escolar y el desarrollo intelectual puede disminuir.

Es aquí donde las TIC juegan un papel fundamental, posibilitando a las personas con algún tipo de discapacidad obtener cierto grado de independencia, a través del uso de programas adaptados o tecnologías de ayuda. Esta independencia, no sólo se logra a nivel lúdico, sino también mejora el rendimiento a nivel social y/o laboral.

En las siguientes secciones, se incluirá un catálogo de ayudas técnicas desarrolladas especialmente para personas con algún tipo de discapacidad motora, cualquiera sea el origen.

AYUDAS TÉCNICAS PARA PERSONAS CON MOVILIDAD LIMITADA

En secciones anteriores se vieron los tipos de lesiones que pueden tener las personas que padecen algún tipo de limitación motora (cerebral, medular, neuromuscular, etc.). Aquí, se mostrará cuales son sus consecuencias y qué recursos informáticos les pueden ayudar a tener una vida autónoma.

Desde el punto de vista de la discapacidad física, el ordenador, a modo de prótesis, abre posibilidades hasta ahora poco imaginables. Personas con un pequeño resto de control motriz pueden llegar a comunicarse con los demás, ocupar su tiempo de ocio de forma gratificante, realizar cursos de formación e integrarse laboralmente. El ordenador, ha ayudado a cambiar la idea de que muchas personas con discapacidades físicas graves deberían estar recluidas (**Sánchez Montoya - 2002**).

Hoy en día, existen muchos fundamentos teóricos sobre la capacidad de las ayudas técnicas e informáticas para adaptarse a los formatos de la actividad escolar e incidir positivamente en el proceso de enseñanza y aprendizaje de los niños y jóvenes con necesidades educativas especiales, sin embargo, al descender a la práctica, se ve que no es sencillo encontrar las TIC adecuadas a cada necesidad (**Tecnologías Adaptadas – 2010**).

En la siguiente sección, se mostrarán algunas ayudas técnicas genéricas disponibles actualmente. Estas ayudas, facilitan el acceso al ordenador, mediante adaptaciones que se usan como complemento para utilizar el teclado o el mouse, además se mostrarán dispositivos que se pueden utilizar para sustituir a algunos de los comúnmente usados.

AYUDAS TÉCNICAS PARA ACCEDER AL TECLADO

El teclado, es uno de los dispositivos más usuales para la entrada de información al ordenador y uno de los periféricos que más adaptaciones puede precisar.

La actuación directa sobre el teclado puede presentar diferentes grados de dificultad. Algunas personas usan las dos manos, una mano, uno o dos dedos y también el pie. Es necesario tener una cierta precisión visual y motórica para localizar una tecla entre un conjunto, situarse sobre ella y activarla realizando una presión que oscila entre los 20 y los 80 gramos. El recorrido suele ser de 3,5 mm, aunque en algunas portátiles se reduce a la mitad debido a sus dimensiones. Existen sencillas ayudas técnicas que pueden resultar de gran utilidad (**Tecnologías para el acceso al ordenador – 2010**) (**Sánchez Montoya – 2002**).

VARILLAS

Con este instrumento, el usuario aumenta su capacidad manipulativa sobre el teclado u otro sistema de entrada como la pantalla táctil o el tablero de conceptos. Se puede adaptar a sus necesidades graduando su longitud e inclinación. Existen una gran variedad de ellas en el mercado: desde las que pueden sujetarse con la boca directamente, o mediante una embocadura de goma que se muerde, hasta las varillas de cabezal, denominadas también licornio, que se usan con la ayuda de unas bandas elásticas por todo el perímetro de la cabeza para que presenten mayor sujeción.

LICORNIO



Es un adaptador forrado para la cabeza con un puntero o señalizador.

El puntero permite el acoplamiento de lápices, pinceles, rotuladores, etc. para facilitar el acceso a las actividades.

Figura 4.1 – Licornio

VARILLA BUCAL



Esta varilla permite al usuario poder pulsar sobre las teclas de un teclado sujetándola con la boca.

Figura 4.2 – Varilla bucal.

CARCASAS

Son unas superficies rectangulares perforadas, habitualmente de metacrilato, que se colocan encima del teclado. Los orificios están situados sobre cada una de las teclas, aunque es posible tapar algunas perforaciones durante el período de entrenamiento e incluso definitivamente, si no se van a utilizar. Este dispositivo, permite que los usuarios con reducido control motórico localicen las teclas con los dedos o con una varilla de cabezal y evita que pulsen accidentalmente otra no deseada. Permiten también que las manos descansen sobre el teclado.



Figura 4.3 – Carcasa para teclado.

SUJETA TECLAS

Este sencillo dispositivo, permite asegurar una tecla mientras se presiona otra. Esta acción, es muy útil cuando se desea pulsar dos teclas a la vez. Al presionar el dispositivo, se asegura la tecla y al volver a presionar se desbloquea. Por ejemplo, se puede asegurar la tecla de mayúscula del teclado para facilitar el uso de determinados programas.

Un ejemplo de esto es el programa Stickykeys que permite asegurar una tecla mientras se presiona otra. Al presionar el dispositivo se asegura la tecla y al volver a presionar se desbloquea.

SOPORTES

Existen en el mercado tableros sobre los que se puede situar el teclado. Su función es semejante a la de un atril. Si el usuario utiliza una silla de ruedas, se puede apoyar sobre sus reposabrazos.

Los brazos articulados facilitan el uso de los conmutadores. Se pueden fijar a la silla de ruedas, a la mesa, o la cama.



Figura 4.4 – Soporte para mouse.



Figura 4.5 – Soporte para teclado.

CONMUTADORES

Cuando se nombró “señalizadores y pulsadores” en este capítulo, se mencionó a los conmutadores. Es importante señalar la importancia de estos dispositivos, ya que existen muchas aplicaciones desarrolladas que facilitan el acceso a cualquier aplicación de la computadora mediante la mera utilización de un conmutador o switch.

Debido a que las personas con problemas motrices muchas veces no pueden utilizar los medios de acceso estándar de las computadoras (teclado, mouse, etc.), es muy común usar en estos casos un dispositivo de acceso llamado “conmutador” (en inglés “switch”, y en cierta bibliografía “pulsador”).

En el área de la tecnología aplicada a las discapacidades, se llama switch (conmutador) a un dispositivo que se conecta a una computadora y que puede encontrarse en uno de los dos siguientes estados: activado o desactivado, permitiendo “activar” o “desactivar” determinada función, y en general son utilizados por personas con problemas motrices severos, para operar diversos equipos (**Construcción de Switches e interfaces - 2010**).

Es habitual utilizar los conmutadores, cuando el usuario presenta una discapacidad motórica y/o mental severa. Siempre deben estar adaptados a sus habilidades específicas para que

pueda operar con ellos sin fatigarse innecesariamente, comprometer su buen tono muscular o causarle dolor por vulnerar sus articulaciones. Las posibilidades de elección son bastante amplias e incluyen los accionados por soplo, succión, sonido, presión, desplazamiento del dispositivo, la acción de cerrar un ojo, etc.

La elección de los interruptores adecuados, es un acto muy importante. Es necesario observar al individuo e ir probando diferentes tipos hasta encontrar el que puede controlar mejor. Es importante, no descartar prematuramente sistemas más normalizados de comunicación, o caer en la tentación de que el sujeto use sólo el miembro que se encuentre en mejores condiciones, cuando los otros pueden necesitar ejercicio y se puede aprovechar la interacción con el ordenador para su rehabilitación.

Algunas características deseables en un switch son las siguientes:

- Forma de activación práctica y cómoda para el usuario (por presión, por tacto, por soplido, etc.)
- Resistencia (sobre todo los que se activan por presión deberán ser resistentes a los golpes).
- Sensibilidad (el grado justo de sensibilidad recomendada dependerá de cada usuario).
- Atractivo (en muchos casos, sobre todo cuando se trabaja con niños, es recomendable que el switch sea “lindo” y colorido).

Si inicialmente se decide que lo conveniente es que accione un solo interruptor y se ve que se consigue un éxito importante, se deben seguir estudiando otras alternativas, como ampliarle la gama de dispositivos de entrada y programas a los que puede tener acceso.

Ahora bien, el interruptor no es una ayuda técnica en sí misma, sino un sistema de acceso a la ayuda técnica y a los recursos tecnológicos (**Pulsadores, conmutadores e interruptores. Sistemas de Acceso para el discapacitado motórico – 2010**).

Cuatro son las líneas de aplicación de los conmutadores, pulsadores o interruptores:

1. Lúdica: reemplaza y/o sustituye a los interruptores que activan el movimiento o acción de un juguete.
2. Educativa: incorporación de estos materiales a la EAO (Enseñanza Asistida por Ordenador). No se pretende reemplazar el uso del teclado o del ratón, sino reemplazar una tecla o la pulsación de la misma.
3. Personal: es posible la adaptación de materiales de uso cotidiano, con ello se pretende dotar al discapacitado motórico de autonomía en alguna de las actividades del hogar, desde encender la radio o usar el mando del televisor.

4. Laboral: con la adaptación de puestos de trabajo, se podría conseguir una inclusión real de la persona con discapacidad en la comunidad.

CLASIFICACIÓN DE CONMUTADORES

Existe una clasificación de los conmutadores, de acuerdo a la acción que hay que realizar para accionar el interruptor. Estos pueden ser:

Acción	Descripción	Modelo	Imagen
Por Presión	El usuario deberá presionar sobre el interruptor, con independencia de la parte del cuerpo que utilice.	Almohadilla Big Red Switch De 2 Funciones Con placa Vibradora.	
Posición	El pulsador deberá de ser cambiado, movido o girado de posición para que se active.	Interruptor de Mercurio.	
De Contacto o Sensibilidad o Deflexión	No necesitan presión sobre ellos, sino que el más leve contacto o roce lo activará.	Cylindrical Touch De Varilla Multicontactos.	
De Palanca	El niño deberá accionar una palanca para activar el conmutador.	De palanca Flexo Flex.	
Tipo Joystick	Se acciona en todas las direcciones. Pueden activarse cada uno independientemente.	Pulsador Joystick.	
De Pedal	Es un conmutador de presión, activado por el pie, de ahí su aspecto robusto.	Interruptor de Pedal de Silla.	

De Soplo – Succión	Se activa con la presión aérea realizada en forma de soplo o aspiración. Personas con graves problemas motóricos y control respiratorio correcto.	Pneumatic Trazador Bucal Soplo-Succión.	
Neumático	Se activa al presionar un recipiente o contenedor lleno de aire o líquido.	Interruptor de perilla Interruptor de collarín.	
Por Sonido o de Vibración	Micrófono que se activa por un sonido, ruido o la propia voz. Más complejos. Sensibilidad regulable.	Interruptor de Sonido Conmutador por Vibración.	
Por Luz	El conmutador emite un haz luminoso o rayo infrarrojo que al tapar el haz se activa.	Pulsador de haz Luminoso.	
Acción Compleja	Ha de realizarse una actividad o acción. Ej.: colocar unas figuras en su posición o completar un puzzle.	Interruptor de formas geométricas Interruptor de cuerda Interruptor de formas geométricas Interruptor de cuerda.	

Tabla 5.2 – Clasificación de los conmutadores de acuerdo a la acción.

También se podrían clasificar los interruptores según el elemento o parte del cuerpo que lo acciona, así se tendrían los siguientes:

Parte del cuerpo	Descripción	Tipo	Modelo	Imagen
Cabeza	Se activa por el movimiento o presión de la cabeza sobre el conmutador.	Presión Neumático.	Interruptor Collarín.	
Mano - Puño	Golpeando, tocando o presionando el interruptor.	Presión Sensibilidad Palanca Joystick Neumático Luz Complejo.	Interruptor de Mercurio Basculante Almohadilla De Golpeo Buddy Button Big Red Switch Cylindrical Touch De Varilla Multicontactos De Flex Joystick De pelota de goma De formas geométricas.	  
Con el Dedo	El más leve movimiento o roce del dedo lo activaría.	Contacto, Sensor Muscular Por luz Neumático.	Sensor Muscular De Varilla Multicontactos Interruptor de Dedo Interruptor de Pera.	
Con el Codo	Presiona con el codo.	Presión.	De Codo.	
Con el Pie	Se activa al pulsar el interruptor por medio del pie.	Presión Interruptor de Pedal o Silla.	Interruptor de Pie.	
Con los Ojos (cejas o pestañas)	El movimiento de las pestañas interfiere un haz luminoso.	Por Luz.	Interruptor de Haz Luminoso.	



Con la Barbilla	El interruptor se activa al pulsar con la barbilla sobre él.	Presión.	De Barbilla.	
Con la Lengua	Sensibilidad a la humedad de la lengua.	Contacto o Sensibilidad.	Interruptor Sensible a la Humedad.	
Con la Boca	Por presión aérea realizada en forma de soplo o aspiración.	De soplo o Succión.	De sonido Pneumatic Trazador Bucal Soplo-Succión Interruptor de Sonido.	

Tabla 5.3 – Clasificación de los conmutadores de acuerdo a la parte del cuerpo utilizada.

ESCRITURA E INTERACCIÓN

ADAPTAR EL TECLADO

Hay algunos teclados que son ergonómicos y están diseñados para reducir la tensión y la carga en los dedos, manos y muñecas, ya que permiten distintas posibilidades de inclinación hasta que se consigue la postura más idónea para teclear. Las teclas se activan con ligeras pulsaciones y es posible presionar la barra espaciadora sin cambiar la posición de la muñeca. Los mini teclados, al ser más reducidos que el estándar, hacen más fácil su uso a las personas que tienen un control limitado. Algunas estructuras rígidas o semirígidas, como las férulas que inmovilizan la mano o el brazo, también sirven para facilitar el acceso al teclado con mayor precisión (**Tecnologías para el acceso al ordenador – 2010**).

TECLADOS PARA UNA SOLA MANO



Teclados para una sola mano que tienen una distribución especial de las teclas y requieren menor amplitud de movimientos para abarcarlas.

Figura 4.6 – Teclado para una sola mano.

TECLADO MALTRON



Teclado ergonómico diseñado especialmente para adaptarse a la forma de las manos y los dedos. Compatible con la mayoría de los ordenadores.

Figura 4.7 – Teclado Maltron.

TECLADOS REDUCIDOS



Sus dimensiones son más reducidas que las de un teclado convencional. Requieren menor amplitud de movimiento.

Figura 4.8 – Teclado reducido.

TECLADOS AMPLIADOS

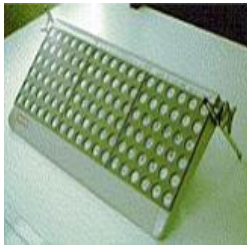


Figura 4.9 – Teclado ampliado.

Teclados ampliados con dimensiones mayores que las de un teclado convencional.

Requieren menos precisión de movimientos para su manipulación.

TECLADOS DE CONCEPTOS



Figura 4.10 – Teclado de conceptos.

Son teclados donde se pueden programar cadenas de caracteres o funciones asociadas a una o varias teclas.

TECLADOS VIRTUALES



Figura 4.11 – Teclado virtual.

En la pantalla del ordenador aparece un teclado.

Consiste en un programa que trabaja simultáneamente con las aplicaciones estándar.

El dispositivo con el que se va a manejar el teclado virtual puede ser un ratón o un pulsador.

EL JOYSTICK SUSTITUYE AL RATÓN

Es posible utilizar un joystick emulando las funciones del ratón. El objetivo del cambio es acceder a todos los programas que usen el ratón con un dispositivo más fácil de manejar. El

concepto de emulador, es muy importante dentro de la tecnología de apoyo a la discapacidad (**Tecnologías para el acceso al ordenador – 2010**).

RATÓN DE BOLA



Es un ratón estático. Tiene dos o tres botones del ratón convencional y una bola integrada con la que se puede controlar los desplazamientos del cursor. Algunos modelos permiten, mediante una sencilla adaptación y un soporte adecuado, su utilización con la barbilla.

Figura 4.12 – Ratón de bola.

RATÓN DE CABEZA



Permite el desplazamiento del cursor del ratón en la pantalla mediante movimientos precisos de cabeza. El usuario se coloca una diadema u otro dispositivo en la cabeza de manera de controlar los desplazamientos.

Figura 4.13 – Ratón de cabeza.

EMULADOR DE RATÓN JULIÁN



Figura 4.14 – Emulador de ratón Julián.

Joystick para trabajar sobre una mesa o una superficie plana, se mueve el puntero del ratón en las cuatro direcciones (arriba, abajo, izquierda, derecha) y en las diagonales. Los cuatro botones hacen las funciones de clic y doble clic sobre botón izquierdo, arrastrar y soltar sobre botón izquierdo y clic sobre botón derecho.

EMULADOR DE RATÓN PRECISO PARA BOCA



Figura 4.15 – Emulador de ratón preciso para boca.

Joystick para trabajar con la boca, se mueve el puntero del ratón de forma lineal, es decir a más desplazamiento del joystick más se desplaza el puntero, pudiendo moverse en cualquier dirección. Las funciones de clic sobre botón izquierdo y botón derecho, se realizan mediante pulsadores de soplo y succión respectivamente.

EMULADOR DE RATÓN JUNIOR



Figura 4.16 – Emulador de ratón junior.

Emulador por barrido, es decir que se trabaja seleccionando la opción deseada mediante un conmutador externo. Las opciones se van resaltando secuencialmente mediante una luz, o bien a ritmo prefijado (modo automático) o al ritmo que marque el usuario a través de otro conmutador (modo manual). Las opciones se agrupan en direcciones de puntero (arriba, derecha, abajo, izquierda) y funciones (clic y doble clic sobre botón izquierdo, arrastrar y soltar sobre botón izquierdo y clic sobre botón derecho).

EMULADOR DE RATÓN PRECISO PARA MANO



Figura 4.17 – Emulador de ratón preciso para mano.

Joystick para trabajar sobre una mesa, se mueve el puntero del ratón de forma lineal, es decir a más desplazamiento del joystick más se desplaza el puntero, pudiendo moverse en cualquier dirección. Las funciones de clic sobre botón izquierdo y botón derecho se realizan mediante conmutadores externos conectados a las dos entradas tipo jack correspondientes.

RECONOCIMIENTO DE VOZ

Uno de los avances más prometedores en el campo de la informática se está produciendo con los analizadores biométricos de matices y modulaciones, dispositivos capaces de estudiar los parámetros de la voz. Aunque actualmente, su uso no da un 100% de fiabilidad, abre nuevas perspectivas sobre todo a las personas con discapacidades motóricas severas. Con el reconocimiento de voz, se podría utilizar cualquier programa, dándole las órdenes a la computadora verbalmente mediante letras, palabras, o frases. La interacción resulta más eficaz que si utiliza, por ejemplo, el lento proceso de comunicarse a través de un conmutador siguiendo la técnica de barrido. Sobre este tema se hablará más detalladamente en el Capítulo 5 (**Sánchez Montoya - 2002**).

MOVER LA CABEZA, DIRIGIR LA MIRADA

El tipo de entrada que se emplea para controlar la computadora va en función de las habilidades del usuario y éstas dependen de factores tan importantes como la existencia de limitaciones físicas (posturales, de coordinación y manipulación), alteraciones de carácter sensorial o perceptivo y también de su capacidad intelectual. Afortunadamente, existe una gran variedad de dispositivos en el mercado, y no es demasiado difícil encontrar el más adecuado (**Sánchez Montoya - 2002**).

Dentro de los simuladores de teclado, los controlados con la cabeza o la mirada, son los más sofisticados. Consisten en un pequeño dispositivo que se sitúa sobre el monitor y un sensor,

muy ligero, que se fija en la cabeza o en la parte del cuerpo que mejor controle el usuario. En la pantalla del ordenador, aparece un teclado y los movimientos del sensor permiten mover el puntero por la pantalla. Para la selección de la opción adecuada, se suele acompañar de un conmutador que se activa con un soplo.

CONCLUSIÓN

Muchas veces la discapacidad motora conlleva a una gradual pérdida de autonomía personal. Da lugar a una serie de impedimentos a nivel social que van desde la concreta discapacidad para trabajar, hasta la limitación en el desplazamiento y desenvolvimiento personal.

Se deben ofrecer los medios necesarios en estos alumnos para que tengan una autoimagen positiva y así se producirá una mejor aceptación de la discapacidad, y como consecuencia surgirán interacciones más gratificantes para todo el entorno.

En este capítulo, se presentó una breve descripción de las diferentes formas en las que se puede manifestar la discapacidad motora y cómo afecta en las diferentes áreas (lenguaje, desarrollo cognitivo y desarrollo social). Estos conceptos sirven al presente trabajo para considerar las distintas dificultades, que se les presentan a las personas con motricidad limitada. Si bien se ha hecho hincapié en la relación entre la discapacidad motora y la dificultad en el desarrollo del lenguaje, esto no se da en todos los casos. Este trabajo está destinado a las personas con problemas motores, pero sin consecuencias o con consecuencias leves en el desarrollo del lenguaje.

Se pensó en este subconjunto de personas, ya que existe gran variedad de ayudas técnicas para personas con discapacidad motriz mediante la utilización de diferentes partes del cuerpo y se considera que sería una buena alternativa, el uso de la voz, si la persona afectada se expresa oralmente sin dificultades. Además, requeriría un menor esfuerzo si la persona pudiera usar la voz para manipular el ordenador y se evitarían las lesiones producidas por “esfuerzo repetitivo”.

Existen muchas personas, de diferentes edades, con movilidad limitada que necesitan contar con ayudas técnicas específicas para desarrollar los procesos de enseñanza y aprendizaje, en todas las áreas dentro de las cuales una persona puede desenvolverse.

Sin embargo, resulta usual, que no pueden acceder a las ayudas técnicas específicas debido a diversas razones. En la mayoría de los casos, esta razón podría ser económica, pero también tienen un alto grado de influencia el hecho de que muchas de las ayudas técnicas, disponibles en el mercado, no se encuentran adaptadas a las diferentes edades cronológicas y necesidades.

Es por esto, que se encuentra como una necesidad imperativa la construcción de adaptaciones de hardware con el software correspondiente, y la elaboración de las aplicaciones didácticas específicas que se deben implementar en dichos casos.

Se han numerado las diferentes adaptaciones que pueden ser utilizadas para facilitar a las personas con movilidad limitada, el acceso a la computadora, permitiendo al usuario lograr una mayor autonomía de su entorno.

En el capítulo siguiente, se analizará de manera general el reconocimiento de voz, sus características, ventajas y desventajas, además de las diferentes tecnologías de reconocimiento de voz, haciendo hincapié en las distintas herramientas que se encuentran en el mercado actual y las tecnologías que utilizan para llevar a cabo su objetivo.

CAPÍTULO 5

RECONOCIMIENTO DE VOZ

INTRODUCCIÓN

El avance tecnológico ha aportado al ser humano nuevas y mayores posibilidades de desarrollar un modo de vida más completo, pero al mismo tiempo exige continuamente nuevos y específicos conocimientos y habilidades en el individuo para poder hacer uso de las posibilidades que le ofrecen. En las personas con algún tipo de discapacidad, la progresiva complejidad del medio social puede tener, sin embargo el efecto contrario al buscado por el progreso social **(Nuevas Tecnologías y educación de personas con dificultades – 2010)**.

Así se encuentra en el reconocimiento de voz una alternativa para la comunicación con la computadora, permitiendo que las personas con discapacidades motoras que no pueden acceder al teclado estándar y al mouse puedan, con el habla, realizar acciones que sin esta tecnología no le serían posibles, en otras palabras, el objetivo es convertir el habla humana en acciones interpretables por la computadora.

Es importante decir que el reconocimiento de voz es aún un problema sin resolver. No existe aplicación que haya demostrado poder realizar el reconocimiento de la voz con la misma habilidad que los humanos han desarrollado. Tal es así, que aunque se hayan presentado muchas maneras de resolver este problema, no se ha alcanzado el auge en esta tecnología. Tal vez a causa de esto, sea que el reconocimiento de voz automatizado se ha transformado en un tema de estudio atractivo para muchos investigadores, desarrolladores y estudiantes. Esto se traduce en que estos sistemas no cuentan con una fiabilidad del 100%, por lo que es un área en la que se necesita una profunda investigación, en la cual se puede incurrir para mejorar la autonomía y calidad de vida de las personas, entre muchas otras aplicaciones posibles.

RESEÑA HISTÓRICA

La historia del reconocimiento de voz, se remonta en el tiempo, hasta 1950 con múltiples paradigmas de trabajo y resultados, inclusive muchas de las técnicas utilizadas con éxito debieron esperar más de 10 años para pasar de la teoría a la práctica, también en laboratorios **(Martínez, Portale, Klein, Olmos, Reconocimiento de voz – 2009)**.

Algunos de sus principales hitos fueron en 1952, cuando se desarrollaron algunos avances como reconocimiento aislado de dígitos, medición de resonancia espectral en vocales, con rangos de 50 al 100%. En 1959 se presentaron avances en cuanto al reconocimiento de vocales y algunas consonantes, con analizador de espectro y comparadores de patrones con resultados del 93%, ambos dispositivos de hardware y exclusivamente en laboratorio.

En los '60 se comenzó a experimentar con normalización temporal según la detección de los puntos de comienzo y fin de las palabras, utilizando en general hardware específico y se empezó usar computadoras.

En los '70 hubo avances significativos en reconocimientos de palabras aisladas, y comienzos de experimentación en reconocimiento independiente del locutor ("speaker independent").

Se advierte que las fuentes de información semántica, sintáctica y contextual, ayudan a mejorar la calidad de los sistemas.

El reconocimiento de una sentencia completa de gramática acotada requería de 50 computadoras (HARPY system del Carnegie Mellon University).

En los '80 se aplicaron los conceptos de "dynamic time warping" (DTW o Deformación Temporal Dinámica). Se produce un importante cambio de paradigma de comparación de plantillas hacia el modelado estadístico/probabilístico como un gran avance de aproximación al reconocimiento de voz.

A mitad de los '80 se hizo masiva una técnica que revolucionó el campo del reconocimiento, se trató de los modelos ocultos de Markov o HMM. A partir de ellos se obtuvieron excelentes resultados en el modelado de señales de voz. Además se reintroduce el uso de redes neuronales (ANN) que habían vencido algunos obstáculos de tipo conceptual y de recursos necesarios para su implementación. También se comenzó a experimentar con reconocimiento continuo de vocabularios largos, independientes del locutor.

En los '90 se comenzó a hacer énfasis en interfaces de lenguaje natural, y recuperación de la información en grandes documentos de voz, continuó la investigación de reconocimiento continuo en vocabularios grandes y a usarse masivamente a través de redes telefónicas, también en el estudio de sistemas en condiciones de ruido.

Antes y durante la mitad de los '90 se dio la investigación de sistemas híbridos HMM-ANN, que también han dado muy buenos resultados.

FUNDAMENTOS DEL RECONOCIMIENTO DE VOZ

El Reconocimiento Automático del Habla (RAH) o Reconocimiento Automático de voz es una parte de la Inteligencia Artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras electrónicas, es decir, es el proceso de conversión de un mensaje hablado a texto, que permite al usuario una comunicación con la computadora. El problema que se plantea en un sistema de RAH es el de hacer cooperar un conjunto de informaciones que provienen de diversas fuentes de conocimiento (acústica, fonética, fonológica, léxica, sintáctica, semántica y pragmática), en presencia de ambigüedades,

incertidumbres y errores inevitables para llegar a obtener una interpretación aceptable del mensaje acústico recibido **(Wikipedia Reconocimiento del habla – 2010)**.

Un sistema de reconocimiento de voz es una herramienta computacional capaz de procesar la señal de voz emitida por el ser humano y reconocer la información contenida en ésta, convirtiéndola en texto u emitiendo órdenes que actúan sobre un proceso. En su desarrollo intervienen diversas disciplinas, tales como: la fisiología, la acústica, el procesamiento de señales, la inteligencia artificial y la ciencia de la computación.

Las principales características que diferencian a los sistemas basados en Reconocimiento de Voz frente a otras alternativas son: la naturalidad que supone utilizar el habla en las operaciones de comando y control, y la precisión y robustez en la comunicación para diferentes usuarios y diferentes entornos. El estado actual de la investigación en Reconocimiento de Voz muestra excelentes resultados de sistemas trabajando en entornos controlados de laboratorio. Sin embargo, una aplicación real de esta tecnología exige un funcionamiento en el mundo real donde el grado de dificultad de los problemas es un orden de magnitud mayor **(Reconocimiento de voz y Educación – 2010)**.

El reconocimiento de la voz es un problema difícil, debido a muchas fuentes de variabilidad asociadas con la señal acústica, cambios en el ambiente, cambios en el estado físico o emocional del locutor, o el tamaño del tracto bucal.

DIFICULTADES PRESENTADAS EN EL RECONOCIMIENTO POR VOZ

Entre algunas de las desventajas que podrían considerarse en el reconocimiento de voz se encuentran los problemas de entendimiento en ambiente donde existe ruido, la pérdida de frecuencias del sonido en el ancho de banda del canal de audio, la fluidez en el habla de las personas ya que esto puede afectar en el entendimiento de las palabras y específicamente en el caso de reconocedores de voz de propósito general el vocabulario puede ser inmensamente extenso, sobre todo en el español, donde existen muchos modismos, lo cual puede crear confusión de palabras **(Capítulo 3, Reconocedores de voz – 2010)**.

Uno de las dificultades más importantes con las que lidia el reconocimiento de voz es la diversidad de discursos que existen **(Kirriemuir, Speech Recognition Technologies – 2010)**:

- En todo el mundo hay varios miles de idiomas activos.
- Dentro de esos idiomas, existen grandes variaciones en la sintaxis y el vocabulario utilizado.
- En diferentes regiones hablan el mismo idioma pero con diferentes acentos.
- Las personas hablan a distintas velocidades y tonos.

CARACTERÍSTICAS DE LOS SISTEMAS DE RECONOCIMIENTO DE VOZ

Según se vio anteriormente, reconocimiento de voz es el proceso por el cual una computadora (u otro tipo de máquina) identifica las palabras del hablante. Básicamente, esto significa que una persona le habla a la computadora y esta reconoce correctamente lo que el locutor está diciendo.

Las siguientes definiciones son básicas para entender la tecnología de reconocimiento de voz (**Speech Recognition HOWTO – 2010**).

SONIDO

Un sonido es una vocalización (hablante) de una palabra o palabras que representan un significado único para la computadora. Estos sonidos pueden ser una sola palabra, un pequeño conjunto de palabras, una sentencia o incluso múltiples sentencias.

DEPENDENCIA DEL HABLANTE O LOCUTOR

Los sistemas dependientes del hablante son diseñados para locutores específicos. Generalmente son más precisos para el reconocimiento de ese hablante, pero mucho menos precisos para otros. Ellos asumen que se hablará en una voz y tiempo coherentes. Los sistemas independientes del hablante son diseñados para aceptar una variedad de hablantes. Los sistemas adaptativos, generalmente, comienzan como sistemas independientes del hablante y utilizan técnicas de entrenamiento para adaptarse al hablante y aumentar la precisión del reconocimiento.

VOCABULARIOS

Los vocabularios (o diccionarios) son listas de palabras o sonidos que pueden ser reconocidas por el sistema de reconocimiento de voz. Generalmente, cuánto más grande es el vocabulario, o conjunto de palabras a reconocer, más difícil resulta su reconocimiento. A diferencia de un diccionario normal, cada entrada no tiene por qué ser una única palabra. Estas pueden ser tan largas como una oración o dos. Los vocabularios pequeños pueden tener una o dos sonidos a reconocer, mientras que los vocabularios muy grandes pueden tener cientos de miles o más.

MODELO DE LENGUAJE

El lenguaje puede ser modelado a través de modelos estadísticos (Statistical Model Language - SLM) o utilizando gramáticas de estado finito (Finite Grammar State - FGS) (**Language Model – 2010**).

Un modelo estadístico captura la probabilidad de las palabras y de las secuencias de palabras. Es utilizado en el decodificador para limitar la búsqueda y generalmente hace una contribución significativa a la exactitud del reconocimiento. Un buen modelo es aquel que modela con precisión la entrada esperada.

Se caracteriza por su orden, en términos de “n-gram”, donde “n” indica el tamaño de la ventana sobre la cual se computan las estadísticas. En general cuanto más grande sea “n” más preciso será el modelo. También, a mayor “n” se necesitan más datos para asegurar que las estadísticas se estiman sólidamente.

Por ejemplo un modelo de lenguaje trigram consiste en:

- **Unigrams:** el conjunto completo de palabras y sus probabilidades de ocurrencia individuales en el lenguaje. Este tipo de modelo debe incluir una sentencia de comienzo y fin especial: <s> y </s> respectivamente.
- **Bigrams:** matemáticamente es $P(\text{palabra2} \mid \text{palabra1})$. Lo que significa, la probabilidad condicional de que palabra2 aparece inmediatamente después de word1 en el lenguaje. Típicamente, solo se incluye un subconjunto de posibles pares de palabras, es decir, no es necesario cubrir todos los pares.
- **Trigrams:** de manera similar a un bigram, es $P(\text{palabra 3} \mid \text{palabra1, palabra2})$, o la probabilidad condicional de que palabra 3 aparezca inmediatamente después palabra 1 palabra 2. No es necesario cubrir todas las combinaciones posibles.

Una gramática de estados finitos define las posibles palabras, así como también el posible orden de dichas palabras.

Cuando se necesita un lenguaje finito, donde se espera que los usuarios sólo utilicen palabras que se encuentran dentro del lenguaje, las gramáticas de estado finito responden mejor. Por el contrario, si se permite que los usuarios puedan hablar con menos restricciones se deberá utilizar un modelo de lenguaje estadístico. FGS sólo reconocerá secuencias de palabras especificadas en la gramática. SML podrá decodificar nuevas secuencias de palabras, pero no aquellas que resultan impredecibles del vocabulario utilizado para el entrenamiento del modelo.

MODELO ACÚSTICO

Un modelo acústico se crea a partir de grabaciones, sus respectivas transcripciones, y el uso de software para crear representaciones estadísticas de los sonidos que componen cada palabra (**Wikipedia Acoustic model – 2010**).

La performance del reconocimiento producida por el modelo acústico puede mejorarse aún más mediante un modelo de lenguaje, el cual contribuye a desambiguar entre varias palabras similares producidas por el modelo acústico.

EXACTITUD

La capacidad de un reconocedor puede ser examinada midiendo su exactitud o lo bien que reconoce expresiones. Esto incluye no sólo la correcta identificación de un sonido sino también la identificación de los sonidos del hablante que no pertenecen al vocabulario. Un buen sistema de reconocimiento de voz tiene una exactitud de un 98% o más. La exactitud aceptable del sistema depende de su aplicación.

ENTRENAMIENTO

Algunos reconocedores de voz, tienen la capacidad de adaptarse al hablante. Cuando los sistemas tienen esta capacidad, deben permitir el entrenamiento. Un Sistema de Reconocimiento de Voz, se entrena por la repetición por parte del hablante de frases comunes o estándares y ajustando sus algoritmos de comparación para que coincida con cada hablante particular. El entrenamiento de un reconocedor, por lo general, mejora su precisión.

Este método también puede ser usado para un hablante con dificultades en el habla, o en la pronunciación de ciertas palabras. Mientras el hablante pueda repetir consistentemente un sonido, un Sistema de Reconocimiento de Voz con entrenamiento deberá ser capaz de adaptarse.

EL PROCESO DE RECONOCIMIENTO

El Reconocimiento Automático del habla consta principalmente de dos etapas, por un lado está la etapa de reconocimiento y por otro lado la de entendimiento (**Capítulo 3, Reconocedores de voz – 2010**).

Algunos de los pasos para llevar a cabo estas fases son:

1. Obtener los archivos de voz (la señal de voz) y digitalizarlos (para esto se debe llevar a cabo la obtención de muestras).
2. Extraer un conjunto de características esenciales de la señal (este conjunto de características será la entrada al clasificador).
3. Introducir el conjunto de características a un clasificador para obtener probabilidades.
4. Búsqueda para encontrar la secuencia permitida más probable. Al tener las probabilidades y con la ayuda de una estructura que tenga las pronunciaciones posibles, se aplica el algoritmo de búsqueda que dará como resultado el reconocimiento de la palabra.

Estos pasos describen de manera general como funciona un reconocedor de voz independientemente de la tecnología que utilice. En la próxima sección se mostrará en detalle que incluye cada uno de estos pasos y cuáles son los métodos utilizados para llevar a cabo cada funcionalidad.

ESTRATEGIAS A UTILIZAR PARA EL RECONOCIMIENTO DE VOZ

Antes de comenzar a contar sobre las estrategias del reconocimiento de voz, es importante hacer algunas aclaraciones.

En la primera etapa, los reconocedores de voz deben implementar lo que se conoce como el establecimiento de las características espectrales del sonido **(Bernal Bermúdez, Bobadilla Sancho, Gómez Vilda, 2000) (Martínez Bernaldo de Quirós, Fundamentos básicos del reconocimiento de voz – 2010)**.

La etapa de la caracterización espectral es necesaria debido a que la información principal de la voz se codifica en el dominio de la frecuencia, mientras que las muestras de sonido que se obtienen con dispositivos electrónicos se encuentran codificadas en el dominio del tiempo (amplitudes de la señal a lo largo del tiempo).

El dominio de la frecuencia es un término usado para describir el análisis de funciones matemáticas o señales respecto a su frecuencia (Dominio de la frecuencia – 2010).

Un gráfico del dominio temporal muestra la evolución de una señal en el tiempo, mientras que un gráfico frecuencial muestra las componentes de la señal según la frecuencia en la que oscilan dentro de un rango determinado. Una representación frecuencial incluye también la información sobre el desplazamiento de fase que debe ser aplicado a cada frecuencia para poder recombinar las componentes frecuenciales y poder recuperar de nuevo la señal original.

Al pasar del dominio del tiempo al dominio de la frecuencia se obtienen parámetros que indican la importancia de cada frecuencia en el intervalo considerado.

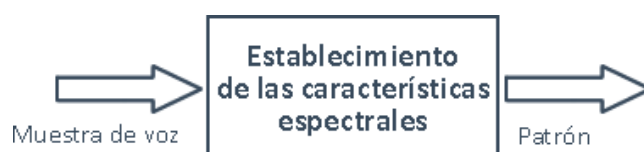


Figura 5.1 – Establecimiento de características espectrales.

Por otro lado, existen tres estrategias fundamentales para abordar el reconocimiento del habla; la primera se basa en la utilización de técnicas de comparación de patrones, la segunda en la obtención de características fonéticas y la tercera en realizar una mezcla de

las dos anteriores. A su vez, la comparación de patrones se puede realizar mediante métodos clásicos o modelos automáticos.

Se deben aplicar mecanismos que traspasen a frecuencias la información existente en las muestras de voz que se recogen en el dominio del tiempo. Estos se explicarán en secciones posteriores.

OBTENCIÓN DE LA SEÑAL SONORA

El primer paso del proceso de reconocimiento de voz es obtener la señal de la voz y digitalizarla. El Diccionario de la Real Academia define al sonido como: “sensación producida en el órgano del oído por el movimiento vibratorio de los cuerpos, transmitido en un medio elástico como el aire”, por lo tanto el origen del sonido es la vibración de los cuerpos. Una vibración es periódica si se repite a intervalos determinados. A continuación se presenta una vibración periódica con forma sinusoidal:

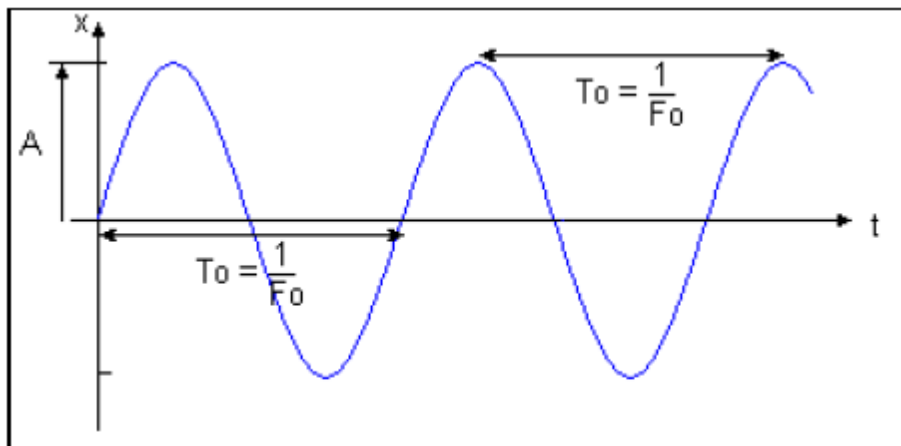


Figura 5.2 - Movimiento vibratorio periódico.

Todo sonido se puede formar a partir de la suma de movimientos armónicos simples. En la figura anterior se puede distinguir los parámetros básicos de una señal periódica. El parámetro A es la amplitud de la onda, que es el valor máximo que puede alcanzar, mientras que T_0 es el periodo de la señal y representa el tiempo que se demora ésta en repetir su comportamiento. El parámetro F_0 se conoce como frecuencia de la señal y es igual al inverso del periodo.

Para facilitar el análisis de un sonido genérico se debe descomponerlo en conjuntos de los armónicos simples que lo forman, ya que sus intensidades y frecuencias informan las características del sonido original.

Una buena parte de los sonidos producidos en el habla son cuasi periódicos, especialmente las vocales. Por ello, para reconocerlas se descomponen en sus ondas simples que lo componen, y según sean las frecuencias de las mismas se pueden distinguir entre diferentes vocales.

El sonido aperiódico o ruido es una onda que no presenta ninguna periodicidad. Al descomponerla en sus armónicos básicos resulta que la energía está muy dispersa en el conjunto de armónicos posibles y es poco estable a lo largo del tiempo.

OBTENCIÓN DE MUESTRAS

En principio se verá de manera abstracta en qué consiste el proceso de obtención de muestras de voz. Se partirá de una señal analógica producida por el hablante y el proceso de muestreo realizado por un convertidor analógico/digital:



Figura 5.3 – Obtención de muestras de la señal de la voz.

Mientras se realiza el proceso de muestreo, los valores numéricos de cada muestra (que indican amplitudes de la señal) se almacenan en la memoria de la computadora o de un hardware específico, para que el tratamiento de señal pueda ser procesado. En esta fase se suelen utilizar dispositivos específicos de entrada/salida para realizar el traspaso del convertidor analógico/digital a la memoria.

Las muestras en memoria pueden ser procesadas directamente por el computador para implementar aplicaciones en tiempo real, o bien pueden ser almacenadas en un archivo para su uso posterior. En caso de que sean almacenadas en un archivo resulta conveniente que se utilice algún estándar de almacenamiento de archivos de voz (por ejemplo WAV), con el fin de unificar formatos y universalizar su utilización desde diferentes programas.

Uno de los puntos a resolver es el tamaño de la frecuencia de muestreo a obtener. Sería razonable pensar que a mayor frecuencia de muestreo se obtendría mejor calidad de la onda capturada, concepto que resulta falso. Si se aumenta la frecuencia de muestreo se conseguirá capturar frecuencias más altas de la onda original, pero las frecuencias bajas se siguen capturando con la misma calidad; este aumento de frecuencia conlleva un coste: convertidores analógicos/digitales más costosos, mayor necesidad de memoria, mayor tiempo de cómputo, etc. Por lo tanto, se puede deducir que no es conveniente capturar muestras a más frecuencia de lo estrictamente necesario.

El criterio de Nyquist expone: “El muestreo de una onda se debe realizar al menos al doble de su frecuencia máxima”²⁶. Si se muestrea por debajo del doble de la frecuencia máxima, se pierde información de las frecuencias más altas; en cambio, al muestrear por encima del doble se incrementa el costo (en procesamiento y capacidad de memoria) sin obtener mayor beneficio.

La frecuencia de muestreo está relacionada directamente con la velocidad de procesamiento de datos y la cantidad de memoria utilizada.

MÉTODOS DE ANÁLISIS DE LA SEÑAL VOCAL

En las siguientes secciones se mostrarán las diferentes alternativas para llevar a cabo los pasos 2, 3 y 4 del proceso de reconocimiento, mencionado anteriormente.

CODIFICACIÓN DIGITAL DE LA SEÑAL

El principal problema de los sistemas analógicos es la adición de ruido y la atenuación, los cuales son más evidentes al utilizar muestras grabadas en cintas magnetofónicas; la digitalización de una señal presenta ciertas ventajas como la disminución de cantidad de información que conlleva una señal como la de voz, hecho notablemente apreciable en ambientes “ruidosos”.

A partir de la señal analógica obtenida mediante un transductor (micrófono) se hace necesario convertir la señal a formato digital para procesarla en un equipo computacional. Este proceso se conoce como Modulación por Código de Pulsos (PCM), y se realiza mediante el muestreo y cuantificación de la señal. La frecuencia de muestreo dependerá del tipo de aplicación. Empleando el criterio de Nyquist será necesario elegir una frecuencia de muestreo que debe ser al menos el doble de la frecuencia más alta.

²⁶ Steven W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, Second Edition.

Otra consideración que se debe tener en cuenta es la cuantificación de la señal, lo cual involucra la conversión de la amplitud de los valores muestreados a forma digital usando un número determinado de bits. El número de bits usados afectará la calidad de la voz muestreada y determinará la cantidad de información a almacenar. Para cada instante de muestreo, el convertidor analógico digital compara la señal muestreada con una serie de niveles de cuantificación predefinidos que dependen de la resolución del equipo. El número de niveles a usar determina la precisión del análisis y por tanto el número de bits necesarios.

Cada bit adicional que se agrega contribuye en mejorar la relación señal a ruido en aproximadamente 6 dB.

Técnicas como PCM (Pulse Code Modulation) se basan en el teorema de muestreo de Shannon:

“cualquier señal de banda limitada puede ser reconstruida con exactitud mediante muestras tomadas periódicamente con una frecuencia de muestreo mayor o igual al doble de la frecuencia más alta de la señal”.

MODELO DIGITAL DE VOZ

Es conocido también con el nombre de modelo de pulso glotal. El análisis de señal vocal se lleva cabo mediante un modelo que describe el proceso del habla clasificando las señales en dos tipos: las señales sonoras y las señales no sonoras o sordas.

Las señales sonoras se caracterizan por tener alta energía y contenido frecuencial en el rango de los 300 Hz a 4 KHz. Se generan por intermedio de las cuerdas vocales y además presentan cierta periodicidad.

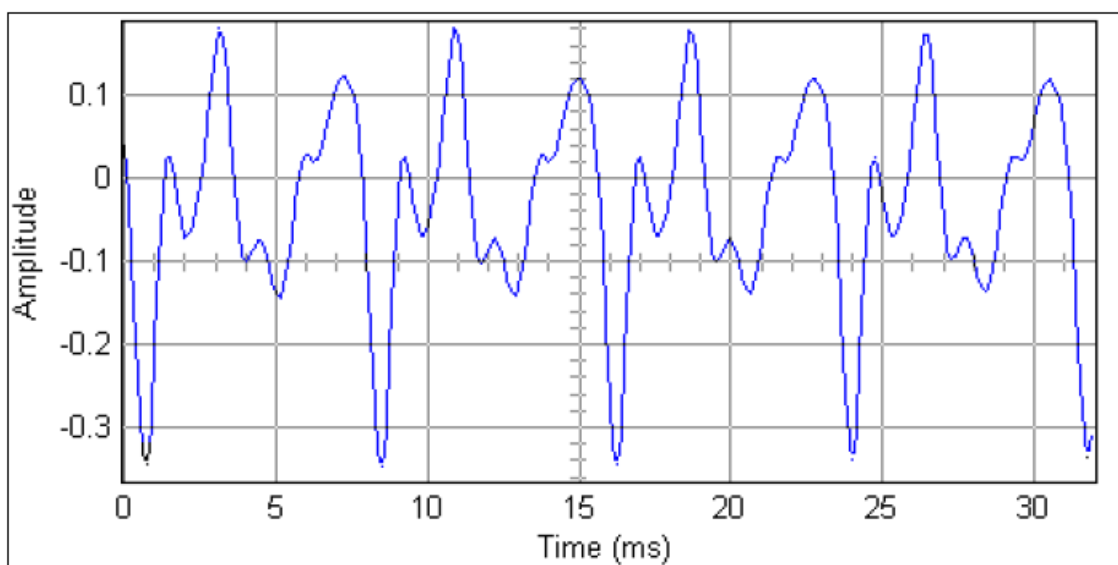


Figura 5.4 - Señal sonora de la vocal 'a'.

Las señales no sonoras se caracterizan por tener baja energía y componente frecuencial uniforme presentando aleatoriedad en forma de ruido blanco.

El ruido blanco es una señal aleatoria que se caracteriza porque sus valores de señal en dos instantes de tiempo diferentes no guardan correlación estadística. Como consecuencia de ello, su densidad espectral de potencia (PSD, Power Spectral Density) es una constante, esto significa que la señal contiene todas las frecuencias y todas ellas tienen la misma potencia. Igual fenómeno ocurre con la luz blanca, lo que motiva la denominación.

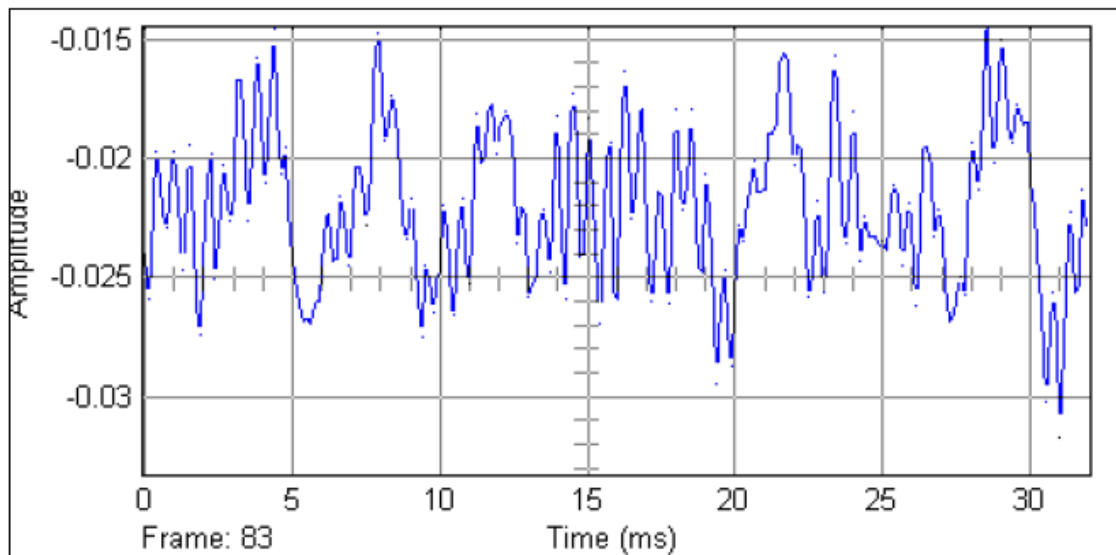


Figura 5.5 - Señal no sonora de la consonante 's'.

Es así como la señal, dividida en dos partes, se modela como dos fuentes: una llamada de excitación $u(n)$ que es un tren de pulsos periódico, de amplitud unitaria y período T (denominado período fundamental), o bien un ruido gaussiano de media nula y varianza unitaria, las que alimentan al sistema acústico que conforma el tracto vocal. Las cavidades resonantes junto al tracto vocal están modeladas mediante un filtro digital con respuesta al impulso $h(n)$. Además, el modelo incorpora una decisión sonoro / no-sonoro para los dos tipos básicos de sonidos y controla el tipo de excitación a generar.

En general durante el proceso de producción de la voz existen variaciones continuas de la excitación y las características del tracto vocal. Sin embargo, esta variación es lo suficientemente lenta como para que la expresión anterior sea válida si se considera un intervalo de tiempo lo suficientemente pequeño. En el caso que la forma del tracto vocal sea fija y la excitación constante, la señal de salida $s(n)$ se puede expresar en términos de la convolución de la excitación con la respuesta al impulso del tracto vocal.

ANÁLISIS LOCALIZADO

La producción de las señales vocales es un proceso transitorio. El sistema fonador altera sus características cuando pasa de un fonema al siguiente; en transiciones de un fonema a otro, el sistema participa de las características de los dos fonemas. Por esta razón, el análisis del habla sólo puede ser un análisis localizado, considerando intervalos de tiempo durante los cuales este proceso de habla tienda a ser estacionario. Un análisis localizado conduce directamente a dos conceptos auxiliares llamados segmentación y ventanas.

SEGMENTACIÓN

La segmentación de una señal de voz (cuyo término en inglés es “frame” y se traduce comúnmente como trama) consiste en separar dicha señal en “trozos” que contengan información y cuya longitud es un problema que puede tomar dos soluciones:

- Si las tramas tienen la misma longitud, conllevaría la ventaja de exigir un esfuerzo uniforme de procesamiento. Presenta la desventaja de que las tramas obtenidas pueden no presentar características uniformes, esto es, que capten secciones pertenecientes a diferentes fonemas.
- Si se adoptan tramas de longitud variable que representen secciones aproximadamente estacionarias del habla, se tiene ventaja en la homogeneidad del segmento y la desventaja en su longitud variable que exige un procesamiento variable.

VENTANAS

Este término (que se traduce al inglés “windowing”), implica la multiplicación de la señal del habla discretizada que se designará por $s(n)$, de duración indefinida, por otra señal $w(n)$, de duración finita igual a N muestras, la cual puede tener diversas formas de acuerdo al propósito del procesamiento de dicha señal. Desplazando la ventana, de duración finita, respecto del habla, de duración indefinida, se podrán procesar las diferentes secciones de esta última.

La definición matemática de la ventana rectangular escrita de forma binaria está dada por la expresión:

$$w(n) = r(n) = \begin{cases} 1 & \text{para } 0 \leq n \leq N-1 \\ 0 & \text{de otra manera} \end{cases}$$

En esta forma se limita el análisis a N muestras, todas las cuales tienen la misma ponderación.

Una alternativa muy utilizada por sus propiedades en señales de voz es la ventana de Hamming.

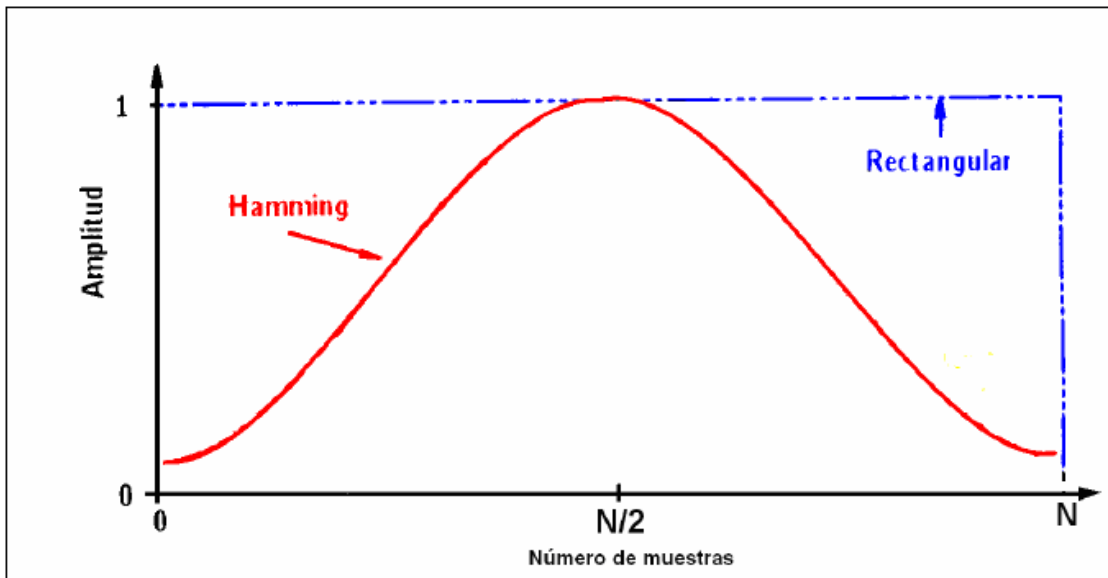


Figura 5.6 – Ventana de Hamming.

El adelgazamiento de los bordes de la ventana permite el corrimiento periódico de la trama de análisis de la señal, sin tener grandes fluctuaciones en los parámetros del habla debido a los comienzos de periodo o a otros cambios súbitos en la señal del habla.

Como los valores de la ventana de Hamming en los extremos quedan muy reducidos, se suele solapar este tipo de ventanas para eliminar el efecto mencionado.

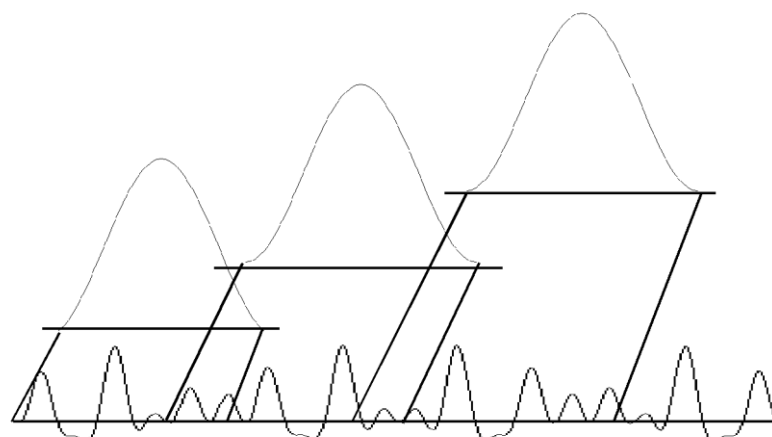


Figura 5.7 – Mecanismo de establecimiento de ventanas.

Una vez establecido el mecanismo de ventanas, las características temporales más comunes que se utilizan en la señal de voz se exponen en los siguientes apartados.

ANÁLISIS EN EL DOMINIO DEL TIEMPO

En el análisis en el dominio del tiempo, el concepto clave es lo que se llama “Análisis de Corta Duración”. Las propiedades de segmentos de habla de entre 10 y 40 ms van a permanecer casi invariables. Se suele hacer análisis de corta duración a intervalos de tiempo espaciados uniformemente (denominados generalmente tramas).

Este análisis se utiliza para realizar mediciones de picos, de energía, de cruce de cero, ocurrencia de sonidos sonoros o no sonoros, de habla y silencio.

MEDICIONES DE ENERGÍA

El cálculo de la energía enfatiza las zonas de gran amplitud de la señal, mientras que la evaluación del módulo evita dicho énfasis y puede ser más sencillo de calcular. Estos valores pueden ayudar a segmentar la señal de voz en unidades fonéticas más pequeñas que podrían corresponder a sílabas o fonemas. Las grandes variaciones de amplitud entre el habla sonora y no sonora, como también las pequeñas variaciones entre fonemas con diferentes formas de articulación, permiten una segmentación basada en la energía, en sistemas automáticos de reconocimiento.

MEDICIONES DE CRUCE POR CERO

Los cruces por cero ocurren cuando una señal cambia de signo y atraviesa el nivel de umbral de ruido. Es un parámetro cuya implementación en el dominio del tiempo es relativamente fácil y en sistemas de procesamiento digital de voz permite junto con la medición de energía detectar características de la señal muestreada tales como el descubrimiento de bordes. Desde un punto de vista acústico, con los cruces por cero se puede intentar detectar las fricaciones²⁷ del habla.

ANÁLISIS EN EL DOMINIO DE LA FRECUENCIA

Los parámetros más útiles para el procesamiento del habla se encuentran en el dominio de la frecuencia. Las señales del tracto vocal se analizan fácil y consistentemente en el dominio de la frecuencia.

²⁷ Ver definición en Anexo B.

Las emisiones repetidas de una sentencia por un mismo locutor difieren considerablemente en el dominio del tiempo, mientras conservan gran similitud en el dominio de la frecuencia. Por esta razón el análisis espectral es usado preferentemente para extraer parámetros relevantes de la señal del habla.

ANÁLISIS DE FOURIER

La Transformada de Fourier es una potente herramienta de análisis espectral muy utilizada en el campo científico. Transforma una señal representada en el dominio del tiempo al dominio de la frecuencia, sin alterar su contenido de información, sólo es una forma diferente de representarla.

Comprender y utilizar los algoritmos de traspaso de una señal en el dominio del tiempo al dominio de la frecuencia, resulta fundamental para obtener los parámetros básicos a partir de los cuales se podrán realizar caracterizaciones espectrales de los sonidos.

Se parte de la base que toda señal genérica, por compleja que sea, se puede descomponer en una suma de funciones periódicas simples de distinta frecuencia. En definitiva, la Transformada de Fourier visualiza los coeficientes de las funciones sinusoidales que forman la señal original.

La siguiente figura muestra como la señal compleja representada en la parte inferior se puede descomponer en las dos señales simples de la parte superior, o desde otro punto de vista, la señal inferior puede ser creada sumando las dos funciones sinusoidales superiores. En este ejemplo, se podría reconocer a simple vista las frecuencias y amplitudes que dan lugar a la señal compleja, sin embargo, complicando la señal resulta un poco más difícil determinar la descomposición en senos (o cosenos) de la función final.

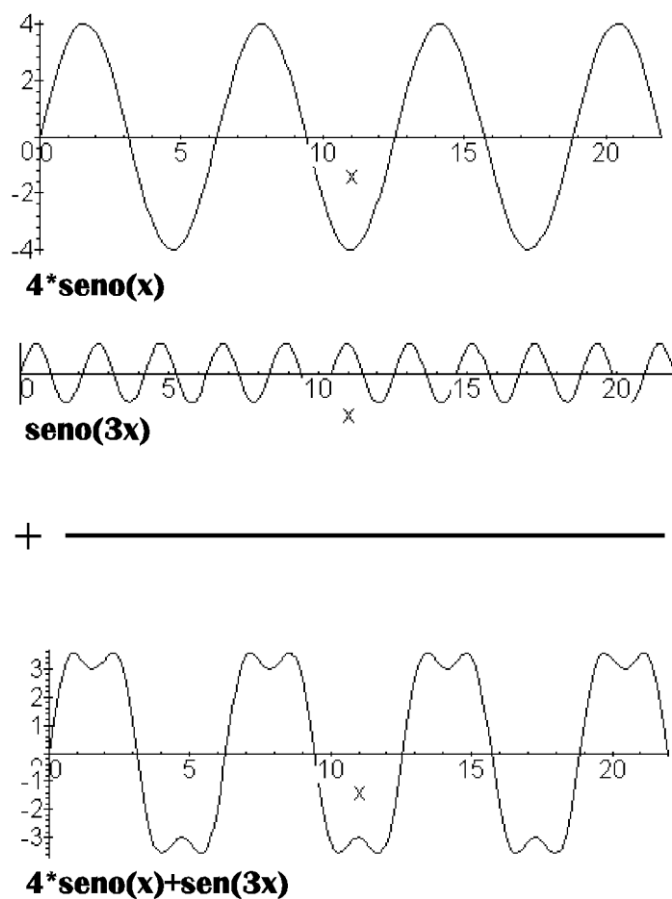


Figura 5.8 – Ejemplo de descomposición de una señal compleja en sumatorias de señales simples.

TRANSFORMADA DISCRETA DE FOURIER

La Transformada de Fourier es una herramienta muy útil cuando se trabaja con modelos matemáticos, pero si se desea trabajar con señales físicas reales y operando mediante computador, se deben utilizar modelos finitos y discretos.

Lo primero que se debe hacer es muestrear la señal de la voz a analizar. Es importante saber la frecuencia de muestreo que se debe aplicar y para ello se debe conocer el ancho de banda de la señal.

Una vez muestreada hay que convertirla en finita. Para ello se limita el número de puntos que se toman. Matemáticamente es multiplicar la señal por una ventana temporal; el efecto que se provoca es convolucionar el espectro de la señal muestreada con el espectro de la ventana, y se produce una distorsión de la transformada de la señal original. Por ello conviene elegir un tipo de ventana que produzca la menor distorsión posible.

La Transformada Rápida de Fourier es un algoritmo que resuelve de una manera más eficiente la transformada discreta de Fourier.

REPRESENTACIONES GRÁFICAS DEL ESPECTRO

Se trata de un instrumento de análisis conocido como espectrograma, que entrega una representación tridimensional de emisiones del habla, empleando la Transformada de Fourier de tiempo corto (jw), la cual se grafica con el tiempo n sobre el eje horizontal, la frecuencia w (de 0 a p) sobre el eje vertical y con la magnitud sugerida con una escala de gris. Solo se representa el módulo del valor complejo de la Transformada de Fourier (en escala logarítmica para tener concordancia con el comportamiento de la sensación auditiva), debido a que la fase no aporta mayor información.

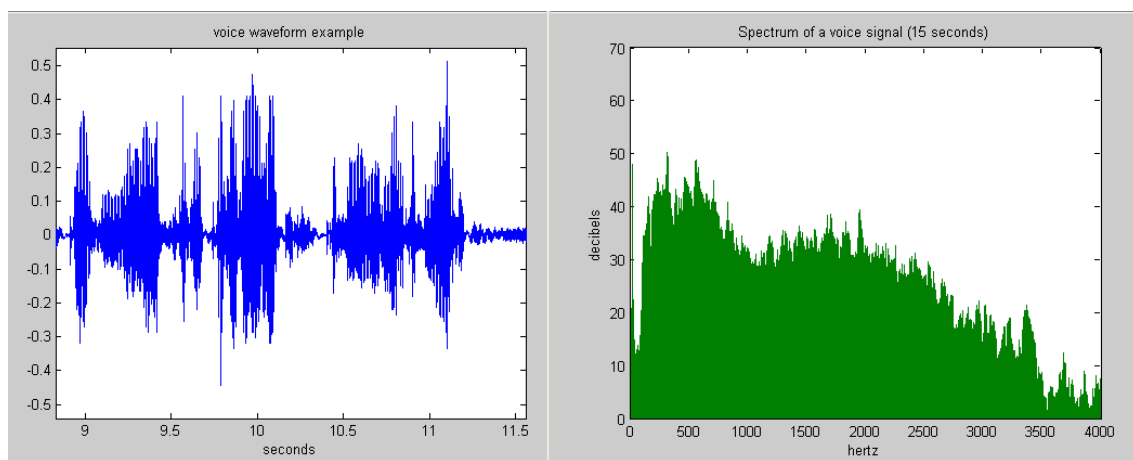


Figura 5.9 – Gráfico de espectro de la señal de la voz.

ANÁLISIS DE PREDICCIÓN LINEAL

Este método se basa en asumir que una muestra de voz puede ser aproximada como una combinación lineal de muestras anteriores. La aplicación del análisis de predicción lineal para estimar parámetros de voz es a menudo llamada Códigos de Predicción Lineal (cuya traducción al inglés sería Linear Predictive Coding LPC) o CPL.

La Predicción Lineal consiste en modelar al tracto vocal humano como un filtro digital constituido únicamente por polos, permitiendo calcular la próxima muestra como la suma ponderada de las muestras previas. Este filtro es manejado por una función de entrada que para sonidos sonoros es un tren de pulsos a la frecuencia fundamental y para los no sonoros es ruido aleatorio con un espectro plano.

PROCESAMIENTO HOMOMÓRFICO

El espectro de una señal de voz refleja tanto las características de la excitación (periodo fundamental), como las características del tracto vocal (formantes). En sistemas de reconocimiento de voz es frecuentemente deseable aislar las dos contribuciones del espectro. Utilizar un filtro sería suficiente si las señales fueran aditivas para separarlas; sin embargo están combinadas de forma convolutiva. Por este motivo, es necesario algún método de análisis que permita la deconvolución de las dos señales. El análisis homomórfico permite realizar este proceso.

En el dominio de la frecuencia en intervalos de corta duración se puede asumir que el espectro del habla es el resultado del producto del espectro de la excitación por la función de transferencia del filtro, compuesto en este caso por los órganos del conducto bucal.

En el dominio del tiempo, lo que se tiene es una convolución de ambos sistemas. El análisis homomórfico del habla es una técnica utilizada para descomponer las dos señales que han intervenido en la convolución temporal.

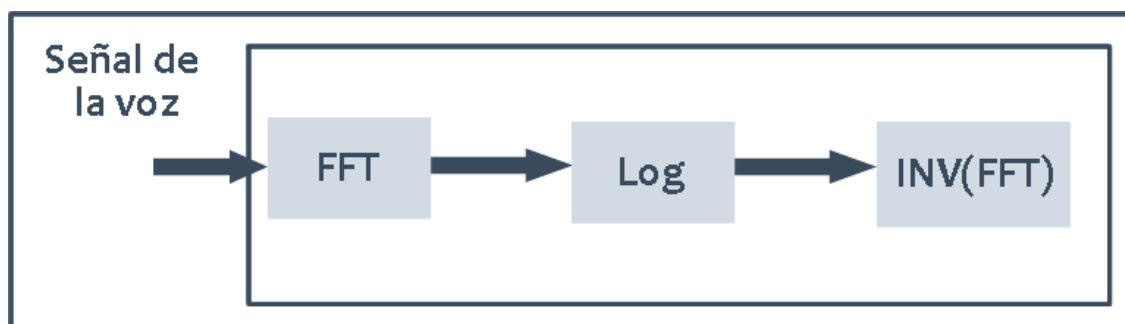


Figura 5.10 – Operaciones del análisis Homomórfico.

En el primer bloque se realiza una transformada de Fourier, con la que se obtiene el espectro frecuencial de la señal de entrada. En este espectro se tiene el producto de otros dos: el de la excitación y el del filtro.

En el segundo bloque se realiza el logaritmo del espectro, transformando el producto en una suma de logaritmos de los espectros: de excitación y del filtro.

La transformada inversa de Fourier realizada en el tercer bloque es el cepstrum²⁸ de la señal de habla y se compone de coeficientes cepstrales altos y bajos.

Se concluye que las componentes cepstrales de la señal de voz son la suma de las componentes cepstrales de la excitación y del filtro que representa el tracto vocal.

²⁸ Ver definición en Anexo B.

Las bajas componentes cepstrales contienen información de la envolvente del espectro que se relaciona con la respuesta en frecuencia del filtro que modela el tracto vocal, mientras que las altas componentes contienen características de la excitación. Para reconocimiento de voz se consideran 12 coeficientes, obtenidos sobre una ventana de 20 ó 30 milisegundos, tiempo en el cual se considera a la señal como estacionaria.

RELACIÓN ENTRE LOS COEFICIENTES CEPSTRUM Y CPL

En muchas ocasiones es deseable combinar la estimación CPL con la utilización de los coeficientes cepstrum, para la caracterización del espectro de una señal de voz. En este caso, se obtienen los coeficientes cepstrales a partir de los coeficientes de predicción.

RECONOCIMIENTO DE PATRONES DEL HABLA

Un patrón se puede definir como una representación que recoge características comunes de una estructura en particular, en nuestro caso, la voz.

Mediante el Procesamiento Digital de Señales es posible el tratamiento de estas señales en computadora ya sea a bajo nivel, donde se las manipula, o también en alto nivel, donde se las interpreta.

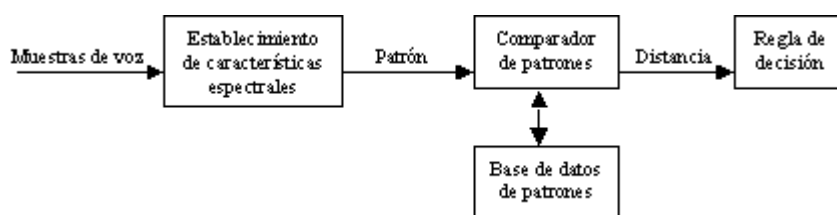


Figura 5.11- Reconocimiento del habla empleando comparación de patrones.

El reconocimiento de patrones trata de diferentes técnicas de clasificación para identificar subgrupos, con características comunes en cada grupo. El primer problema es, dado un vector de mediciones, reducir las dimensiones del espacio de mediciones que suele ser muy grande. Este nuevo espacio debe mantener las propiedades y rasgos distintivos del vector de mediciones, proceso denominado selección de rasgos.

Se tienen tantas clases dentro del espacio de mediciones como el número de individuos que constituyen la población de muestras. El conjunto de patrones que pertenece a la misma clase (propia de un individuo) se agrupa en alguna región del espacio. Como se puede ver en la siguiente figura las clases normalmente suelen estar solapadas.

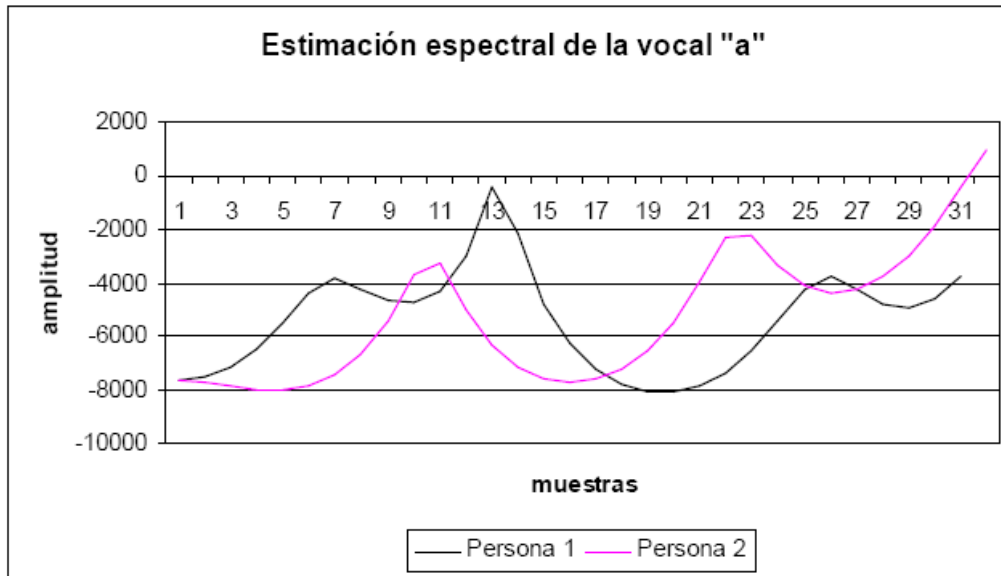


Figura 5.12 – Dos clases de patrones solapados.

TÉCNICA DE CLASIFICACIÓN DE PATRONES

Luego del procesamiento de la señal de voz se obtienen patrones que contienen información respecto a determinada característica de la señal. Es necesario entonces emplear técnicas que permitan clasificar y establecer el proceso de comparación con las muestras posteriormente analizadas (proceso conocido también como “parametrización”). Las técnicas más utilizadas en el reconocimiento de voz son:

- *Técnicas topológicas:* Alabeo Temporal Dinámico (cuya traducción al inglés es “Dynamic Time Warping” o DTW), basado en el cálculo y comparación de distancias. La técnica de Alabeo Temporal Dinámico, conocida también con el nombre de Deformación Temporal Dinámica, empareja dos modelos de longitudes diferentes dinámicamente. En términos generales DTW es un algoritmo para medir la similitud entre dos secuencias que pueden variar en el tiempo o la velocidad. En el caso del habla, el algoritmo encuadra el eje de tiempo de la palabra patrón con el eje de tiempo de la palabra desconocida, mientras considera en lo posible la distancia más pequeña de las componentes de la palabra.
- *Técnicas probabilísticas:* Modelos ocultos de Markov (en inglés “Hidden Model Markov” o HMM), que son modelos generativos de las palabras del vocabulario. En esta técnica se reconoce el habla mediante la estimación de la semejanza de la señal de entrada con una colección de modelos (prototipos) obtenidos en la fase de aprendizaje del sistema. En HMM se asume que la voz es un proceso estacionario. Esto significa que cada pronunciación acústica se planea como una serie de estados estacionarios discretos, con transiciones instantáneas entre ellos. La cadena

resultante de estados diferentes se usa para modelar frases enteras permitidas dentro de un modelo del idioma.

- *Sistemas basados en el conocimiento:* reconocedores por reglas o sistemas expertos. El sistema de expertos tiene dos módulos básicos: la Base de conocimiento, con los datos e informaciones que se usan durante el razonamiento, y el Motor de inferencias, que es la estrategia de razonamiento del sistema.
- *Redes neuronales:* las redes neuronales consisten en una simulación de las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, un ordenador o un conjunto de válvulas). El objetivo es conseguir que las máquinas den respuestas similares a las que es capaz de dar el cerebro, que se caracterizan por su generalización y su robustez. Las arquitecturas en red con un gran número de conexiones entre varias capas de procesadores fueron introducidas para reproducir su estructura. La información contenida en una red neuronal se codifica en la estructura de la red y en los pesos de las conexiones.

Una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por tres funciones:

- Una función de propagación (también conocida como función de excitación), que por lo general consiste en la sumatoria de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina excitatoria; si es negativo, se denomina inhibitoria.
- Una función de activación, que modifica a la anterior. Puede no existir, siendo en este caso la salida, la misma función de propagación.
- Una función de transferencia, que se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que se quiera darle a dichas salidas. Algunas de las más utilizadas son la función sigmoidea (para obtener valores en el intervalo $[0,1]$) y la tangente hiperbólica (para obtener valores en el intervalo $[-1,1]$).

En las cuatro técnicas de reconocimiento mencionadas se puede hablar de una fase de "entrenamiento" (cálculo de los patrones de referencia, cálculo de los parámetros de los modelos de Markov, creación de estructuras de datos para los sistemas expertos o

entrenamiento de las redes neuronales) y de otra fase de "reconocimiento" propiamente dicho.

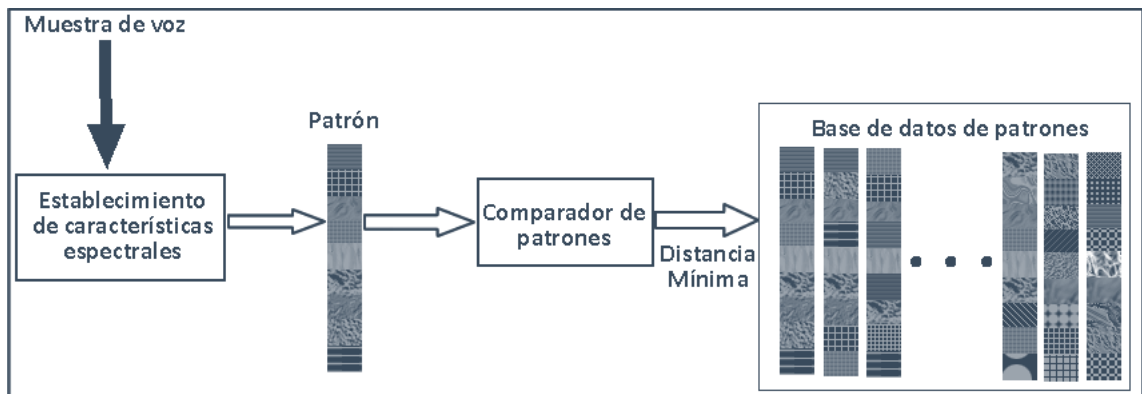


Figura 5.13 – Funcionamiento general del reconocimiento de patrones.

CONCLUSIÓN

La tecnología de reconocimiento por voz ha revolucionado la forma en la que las personas interactúan con las computadoras. Esta tecnología está a la vanguardia de brindar más oportunidades a la educación, al empleo y al ocio.

La tecnología de reconocimiento de voz ayuda a las personas con ciertas discapacidades a interactuar con la computadora con más facilidad. Las personas con limitaciones motoras, que no pueden usar el teclado y el mouse, pueden usar sus voces (si esto fuera posible) para navegar en la computadora.

En este capítulo se describieron distintos aspectos del reconocimiento de voz, con el objetivo de comprender un poco mejor como funciona esta tecnología y qué procesos son llevados a cabo para efectuar el reconocimiento de voz por parte de una máquina. En el siguiente capítulo se introducirá una amplia lista de softwares que pueden ser incorporados a cualquier aplicación para proveer reconocimiento por voz.

CAPÍTULO 6

ANÁLISIS DE SOFTWARE PARA EL RECONOCIMIENTO DE VOZ

INTRODUCCIÓN

En el capítulo anterior se introdujo el concepto de reconocimiento de voz, se analizaron sus características y la forma en que esta tecnología trabaja, obteniendo así una base para la comprensión y análisis más consciente de lo que es un software de reconocimiento de voz.

En el presente capítulo, se introducirán algunos software y frameworks de reconocimiento de voz que se encuentran en el mercado. Cabe aclarar, que estos, varían mucho unos con otros, brindando así un amplio contenido sobre lo que se puede encontrar cuando se quiere introducir o utilizar reconocimiento de voz. Se describirán sus características principales, licencias y demás, utilizando como base los fundamentos vistos en el capítulo anterior.

También se estudiará la Java Speech API, una interfaz multiplataforma que sirve para soportar comandos, reconocimiento y síntesis de voz, y de esta forma poder incorporar reconocimiento por voz en los programas. JSAPI sólo provee la interfaz, la cual deberá ser implementada por terceros, lo que resulta muy común, pues esta interfaz es ampliamente usada.

Además, se describirán determinados criterios de evaluación para productos de reconocimiento de voz, los cuales se listarán en la siguiente sección.

Finalmente, se presentará una sección de descripción de los softwares y frameworks de reconocimiento de voz utilizando la información recolectada sobre dichos productos. Se tendrán en cuenta algunos de los criterios de evaluación antes mencionados

CRITERIOS DE EVALUACIÓN DE PRODUCTOS DE RECONOCIMIENTO DE VOZ

Estos productos pueden beneficiar a todos, incluyendo personas con discapacidades, pero tienen diferencias en su exactitud, interfaces de usuario y usabilidad. Para cualquier persona que esté interesado en el reconocimiento de voz, es importante entender estas diferencias, y cómo va a repercutir el éxito del uso de estos productos. Se proponen algunos criterios para evaluar estos productos de reconocimiento de voz, entre ellos los siguientes²⁹:

- ¿El software permite entrenamiento?
- ¿Cuánto entrenamiento se necesita para que el software de reconocimiento de voz resulte productivo?

²⁹ Esta lista ha sido extraída de:

<http://www.dcp.ucla.edu/dcppresents/Choosing%20and%20Using%20Speech%20Recognition%20Technology.htm>

- ¿Es la interfaz de usuario fácil de entender y lógicamente organizada para el nuevo usuario del producto?
- ¿Con qué facilidad puede el usuario interactuar con la computadora usando solamente su voz?
- ¿Es la formación y corrección de cada uno de los archivos de discurso fácil e intuitiva para una amplia gama de usuarios?
- ¿El software soporta un amplio rango de patrones y acentos?
- ¿Se puede personalizar el software para permitir el ajuste para los diferentes estilos de habla de las personas?
- ¿El software se integra de manera transparente en una amplia variedad de programas?
- ¿Puede la velocidad de discurso de una persona variar sin reducir la precisión del software de reconocimiento?
- ¿El software incluye salida de voz o la integración con otros programas de salida de voz para ayudar a personas con discapacidades de aprendizaje o deficiencias visuales?
- ¿La compañía provee soporte para el software?
- ¿Bajo qué licencia se distribuye el software?
- ¿El software sigue siendo desarrollado para incrementar su usabilidad?

Estos criterios permiten obtener un análisis del producto en cuestión, pudiendo ver sus ventajas y desventajas a la hora de aplicarlo a un determinado contexto, pues no siempre las necesidades y posibilidades del destinatario son las mismas.

ANÁLISIS DE DIFERENTES HERRAMIENTAS DE RECONOCIMIENTO DE VOZ

En esta sección se hará una revisión de diferentes softwares que permiten abordar el reconocimiento de voz. Por supuesto, no es una lista acabada de lo disponible en el mercado, pero se han analizado algunos que han sido seleccionados por diferentes razones: por haber cierta información general disponible o por ser recomendados por gente vinculada al área de reconocimiento de voz.

LOQUENDO

Loquendo ASR (“Automatic Speech Recognition”) es un software independiente del locutor y que reconoce fiablemente un amplio vocabulario de discurso continuo, aún en ambientes ruidosos.

Loquendo es un proveedor de tecnología de voz que brinda una línea completa de productos para servidores, celulares, computadoras de escritorios y dispositivos embebidos, permitiendo la misma amplitud de rango de lenguajes y la misma precisión y performance en todos estos ambientes (**Loquendo, sitio oficial – 2010**).

ASR HERRAMIENTAS DE PUESTA A PUNTO

El software mencionado provee un paquete de herramienta que automáticamente analiza la información recolectada en el área para mejorar el desempeño del servicio, incluyendo:

- **Aprendizaje Fonético:** el cual automáticamente analiza la información de la aplicación para identificar formulaciones frecuentes que no han sido cubiertas y variantes de pronunciación adicionales para mejorar las gramáticas de reconocimiento de voz.

Dado que resulta difícil predecir confiablemente, a priori, las formulaciones de los usuarios, se desarrolló un paquete de software para detectar, a partir de los datos de campo, las formulaciones/peticiones de usuario que no fueron previstas por los diseñadores de la aplicación. Los objetivos principales de esta herramienta son la individualización de formulaciones frecuentes no cubiertas y el estudio de variantes de pronunciación adicionales para las formulaciones previstas. Esto es particularmente útil en el caso de hablantes no nativos /extranjeros y acentos regionales.

Loquendo ha desarrollado un modelo ASR acústico mixto para castellano español y catalán. El desarrollador de la aplicación debe especificar dentro de una gramática qué palabras han sido catalogadas sólo como catalán o como castellano. La precisión es similar a la performance del modelo acústico de un único idioma. Esto permite en países bilingües proveer servicios de discurso eficiente, manejando ambos idiomas al mismo tiempo.

- **Modelo de adaptación acústica:** incrementa el rendimiento del reconocimiento usando material de audio grabado en el área (ambiente, hablante, canal de adaptación), donde una aplicación vocal se usa en un contexto particular.

En la nueva versión, se ha publicado recientemente una herramienta de adaptación innovadora; el modelo de adaptación acústica de Loquendo, basado en un

tecnología

patentada.

Los modelos acústicos publicados en cualquier producto ASR comercial, son modelos generales derivados de grandes corporaciones de telefonía, involucrando muchos locutores seleccionados para representar una distribución típica en edad, sexo y área geográfica. Ellos representan un promedio en discursos telefónicos para cada idioma, y garantiza un muy buen rendimiento en la mayoría de los casos. Cuando una aplicación de voz es usada en un contexto no estándar, el modelo acústico de adaptación puede ser utilizado para incrementar el desempeño del reconocimiento.

Una aplicación que es siempre usada por el mismo hablante, por ejemplo, puede ser mejorada si los modelos acústicos son adaptados a ese hablante. De la misma manera, una aplicación que es usada en un ambiente particular (ejemplo in-car) o con determinado tipo de dispositivo de entrada (ejemplo, un PDA particular) puede mejorar si los modelos generales son adaptados grabando material de voz adecuado extraído de la interacción con el usuario.

Los escenarios más comunes de adaptación son:

- Locutor
- Canales de entrada de audio y multimedia. Ejemplo: Micrófonos específicos, PDA
- Ambientes. Ejemplo: en el automóvil, en una fábrica
- El estilo del hablante. Ej.: acentos regionales, velocidad del discurso.
- Vocabulario dependiente de la aplicación. Ejemplo: jerga, término científicos.

En todos los casos, la adaptación es realizada usando material de audio grabado en el ambiente adecuado o desde la fuente adecuada, y puede mejorar el rendimiento del reconocimiento. El módulo del modelo de adaptación acústica publicado con Loquendo ASR permite a los modelos acústicos ser adaptados con sólo una pequeña cantidad de información de voz recolectada fácilmente. De hecho, recolectar grandes cantidades de grabaciones de voz de hablantes específicos o en condiciones específicas para entrenar los modelos acústicos desde cero, no es práctico.

LOQUENDO ASR - ESPECIFICACIONES TÉCNICAS

Principales características

- Independiente del hablante
- Vocabulario abierto
- Robusto frente al ruido (por ejemplo, en el coche, con telefonía móvil, etc.)
- Optimizado para uso telefónico

Tecnología básica

Una combinación de Redes Neuronales y Modelos Ocultos de Markov de Densidad Continua.

Idiomas soportados

Inglés Británico y Americano, Francés Canadiense, Portugués Brasileño, Norte Americano, Argentino, Chileno, Colombiano, Mexicano, Español Castellano, Catalán, Valenciano, Gallego, Holandés, Francés, Alemán, Griego, Italiano, Polaco, Portugués Europeo, Sueco, Turco y Ruso.

Sistemas Operativos soportados

MS Windows (XP, Vista, Server 2003, Server 2008*), Red Hat Enterprise Linux (3, 4, 5*), SUSE Linux Enterprise 10.0 * también disponible para versión 64 bit.

Requisitos de CPU

Reconocimiento de cifras conectadas: 80 canales en un Intel Pentium 3.2 GHz CPU
Gramáticas con 10,000 palabras: 20 canales en un Intel Pentium IV 3.2 GHz CPU.

Requisitos de memoria

15 MB por idiomas compartidos entre canales.
Unos pocos MB por canal dependiendo de la tarea de reconocimiento (por ejemplo 5 MB para reconocimiento de cifras conectadas, 15 MB para gramáticas con 10.000 palabras).

LICENCIA

Es un programa de licencia propietaria, con lo cual sólo se puede adquirir mediante el pago del producto al momento de su compra.

XVOICE

Permite el dictado de un discurso continuo y control por voz de la mayoría de las aplicaciones X30. Para convertir el habla del usuario a texto, usa el motor de reconocimiento

³⁰ Ver definición en Anexo B.

de voz de IBM, Viavoice, el cual se distribuye de manera separada (**XVoice, sitio oficial – 2010**).

Cuando el programa está en modo de dictado, pasa el texto directamente a la aplicación X que tiene el foco. Cuando está en modo comando, compara lo hablado con secuencias o comandos predefinidos que pueden ser modificados por el usuario. Por ejemplo "list" se va igualar a "ls -l", cuando se esté dando un comando a la consola, de manera que cuando el usuario dice "list" "ls -l", se va a enviar a la consola como si el usuario lo hubiera escrito.

EL MOTOR DE IBM VIAVOICE

Xvoice es un software gratuito, pero el motor IBM del cual depende, no lo es. IBM Via Voice utiliza como tecnología básica los Modelos Ocultos de Markov. Para poder ejecutar Xvoice, es necesario tener una versión con licencia de ViaVoice para Linux. El equipo de desarrollo de Xvoice se encuentra investigando motores de reconocimiento de voz de código abierto.

Para instalar este software a través de RPM³¹, sólo es necesario el Run Time Kit (ViaVoice_runtime-3.x-x.x.rpm). IBM ya no provee este RPM a través de una descarga gratuita. El Runtime RPM está disponible como parte del paquete IBM ViaVoice Dictation for Linux , el cual se puede comprar online.

Para compilar a partir del código, es necesario contar con el ViaVoice Speech Recognition (ASR) SDK (ViaVoice-sdk-3.x-x.x.rpm). Este RPM ya no está disponible en IBM.

Debido a que el IBM ViaVoice SDK del cual depende Xvoice no se encuentra disponible, se ha iniciado un proyecto para reemplazarlo utilizando Sphinx, software que se analizará más adelante.

VIAVOICE Y XVOICE: RECONOCIMIENTO DE VOZ

El control por voz en Linux es posible usando dos paquetes de software. IBM ViaVoice para Linux suministra el motor de reconocimiento de voz. XVoice, disponible bajo licencia GPL, utiliza las librerías de ViaVoice para proveer el control de las aplicaciones de escritorio.

REQUISITOS DE VIAVOICE

Para sistemas Linux requiere un 233MHz Pentium MMX o superior, con al menos 128MB de memoria RAM y una placa de sonido de 16-bit.

³¹ Ver definición en Anexo B.

ENTRENANDO VIAVOICE

Una instalación básica de ViaVoice, así como ocurre con otros software de reconocimiento de voz, no provee un alto grado de precisión al comienzo. Cada usuario debe entrenar ViaVoice para que reconozca mejor su voz.

Un método de entrenamiento es leer el texto que ViaVoice muestra. Este proceso es fácil de llevar a cabo, pero puede que no refleje las palabras o frases que el usuario tiende a usar a menudo, haciendo que sea poco efectivo.

Una mejor alternativa es usar la aplicación ViaVoice Dictation Java cuando se trabaja con documentos. A medida que se realiza el dictado, algunas palabras o frases no son correctamente reconocidas. Cuando esto ocurra, se utilizan las facilidades de corrección para mejorar los errores. Así ViaVoice modifica sus modelos para adaptarse mejor a la voz del locutor. Este método requiere más trabajo, pero por lo general, estas correcciones se pueden hacer con comandos por voz.

DESCRIPCIÓN GENERAL DE XVOICE

XVoice permite al usuario asociar un conjunto de funciones con comandos de voz predefinidos. Las gramáticas pueden ser asociadas a aplicaciones específicas, ventanas o modos dentro de una aplicación. También pueden ser generales y accesibles desde cualquier contexto. Las acciones invocadas pueden ser combinaciones de teclas, eventos del mouse, llamadas a programas externos o cualquier combinación de estas.

XVoice usa las librerías de ViaVoice para reconocer comandos o texto. Los comandos están definidos en un archivo de configuración que se llama xvoice.xml. En principio, usa un archivo de configuración estándar /usr/share/xvoice/xvoice.xml, hasta que el usuario cree el suyo ~/.xvoice/xvoice.xml.

La ventana de XVoice muestra qué gramáticas de comandos están activas e incluye un panel que muestra las palabras dictadas más recientes. Si se detecta algún probable comando, pero la respuesta no es exacta, entonces el texto se muestra en gris para alertar al usuario, y no se ejecuta la acción del comando.

El programa puede estar en cuatro estados diferentes para cualquier ventana dada. En modo comando, escucha sólo comandos. En modo de dictado, no escucha comandos específicos de aplicaciones (aunque si escucha comandos más generales) y simplemente escribe lo que crea que el usuario haya dicho. En modo ocioso, sólo escucha comandos generales. Finalmente, en modo de dictado y comando, escucha por ambos simultáneamente. Los comandos se distinguen del texto haciendo una pausa pequeña antes y después de decir el comando.

Cuando se hace foco en una aplicación, automáticamente empieza en modo comando. Para activar el modo dictado, simplemente se debe decir "dictate mode". Para detener el dictado, se debe decir "stop dictation".

IDIOMAS SOPORTADOS

Únicamente inglés, aunque también alemán en este caso será necesario recompilar Xvoice.

LIMITACIONES

Algunas aplicaciones, principalmente juegos como TuxRacer y otros, no se pueden controlar con software. Aplicaciones en las que es necesario usar mayoritariamente el mouse, tales como The Gimp o Netscape, pueden ser automatizadas, pero resulta extremadamente tedioso tratar de controlar el mouse a través de la voz para esos casos.

El reconocimiento de voz en general funciona muy bien para los comandos y bien para el texto. Sin embargo, aún existen tasas pequeñas de errores. Mientras que XVoice y ViaVoice otorgan mucho control, no es posible controlar el escritorio Linux por completo utilizando la voz.

No se encuentra disponible para sistemas operativos Windows.

NICO TOOLKIT

El NICO Toolkit es un toolkit diseñado y optimizado para aplicaciones con tecnología de reconocimiento de voz (**NICO Toolkit, sitio oficial – 2010**).

El nombre surge de "Neural Inference COmputation", y se desarrolló entre los años 1993 y 1997. Fue realizado por Nikko Strom del departamento de "Speech, Music, and Hearing" en el Royal Institute of Technology (KTH).

CARACTERÍSTICAS

Este software utiliza Redes Neuronales Artificiales y fue desarrollado utilizando el lenguaje C. Todas las herramientas que provee este toolkit son comandos que pueden ser ejecutados desde consola de comandos, y se pueden dividir en tres grupos principales:

- Building Tools: el cual contiene herramientas para definir la topología de la red, especificar formatos de entrada y salida, etc.

- Training Tools: provee herramientas para la normalización de los datos de entrada, algoritmos de entrenamiento, etc.
- Evaluation Tool: permite ejecutar la red, realizar evaluaciones, etc.

REQUISITOS DEL SISTEMA

Fue testeado solamente en sistemas operativos Linux y se necesita un compilador gcc³² para ejecutar los comandos de este software.

LICENCIA

Este software se distribuye bajo licencia BSD³³.

CMU SPHINX

Es un sistema diseñado en la Universidad de Carnegie Mellon (CMU), el cual es uno de los mejores y más versátiles, sistemas de reconocimiento del mundo actual (**Sphinx, sitio oficial - 2010**).

Existen varias versiones, siendo la última de ellas Sphinx-4, el cual fue creado en colaboración conjunta entre el grupo de Sphinx de la Universidad de Carnegie Mellon, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL), y Hewlett Packard (HP), y con contribuciones de la Universidad de California en Santa Cruz (UCSC) y el Massachusetts Institute of Technology (MIT).

Sphinx-4 comenzó como un puente de Sphinx-3 al lenguaje de programación Java, pero evolucionó en un reconocedor diseñado para ser mucho más flexible que Sphinx-3, convirtiéndose así, al decir de muchos usuarios, en una excelente plataforma para la investigación del reconocimiento por voz.

CARACTERÍSTICAS

Es un sistema basado en los modelos ocultos de Markov (HMM), como muchos otros sistemas de reconocimiento de voz, para su funcionamiento primero se debe aprender las

³²Ver definición en Anexo B

³³Ver definición en Anexo B

características (o parámetros) de un conjunto de unidades de sonido. Luego utiliza lo que ha aprendido, para encontrar la secuencia de unidades de sonido más probables para una señal de voz dada. El proceso de usar el conocimiento adquirido, para deducir la secuencia de sonido más probable en una señal dada se llama decodificación, o simplemente reconocimiento.

La última versión de Sphinx, Sphinx- 4, está completamente desarrollada en tecnología Java, haciendo uso de la Java Speech API (JSAPI).

Puede ser ejecutado en modo “batch” o modo “live”. Es capaz de reconocer discursos continuos o discretos.

Incluye soporte para modelos de lenguajes ASCII y versiones binarias de Java Speech API Grammar Format (JSGF), y ARPA-Format FST.

COMPONENTES PROVISTOS PARA EL RECONOCIMIENTO

El decodificador consiste en un conjunto de programas, que han sido compilados en un único ejecutable que va a realizar la tarea de reconocimiento, dadas las entradas correctas. Las entradas que se necesitan son: los modelos acústicos, un archivo índice modelo, un modelo de lenguaje, un diccionario de lenguaje, y un conjunto de señales acústicas que deben ser reconocidas, también llamada datos de prueba.

En resumen, los componentes previstos para la decodificación son:

1. El código fuente del decodificador.
2. El diccionario de lenguaje.
3. El diccionario de relleno.
4. El modelo de lenguaje.
5. Los datos de prueba.

REQUISITOS DEL SISTEMA

Sphinx-4 fue desarrollado y testado en sistemas operativos Solaris, Mac OS X, Linux y Windows. Ejecutar, compilar y testear Sphinx-4 requiere de software adicional. Por lo que antes de comenzar a usarlo, es necesario disponer de los siguientes softwares en la computadora:

- Java SE 6 Development Kit o superior.

- Ant 1.6.0 o superior. Sólo es necesario si se quiere compilar Sphinx-4 a partir del código fuente.

LICENCIA

Los términos de licencia para el motor y herramientas se derivan de la licencia BSD.

DRAGON NATURALLY SPEAKING

Es un software de reconocimiento de voz el cual es comercializado por Nuance Communications para sistemas Windows (**Experto en TICs y discapacidad, 2009**).

El objetivo de este programa es facilitar la interacción con la PC, sin necesidad de usar el teclado ni el mouse. Para ello, el programa debe ser capaz de analizar un flujo entrante de sonidos y de interpretar tales sonidos como comandos y dictado.

Para lograr una buena precisión del reconocimiento, este motor recurre a varias fuentes de información: un modelo acústico, un vocabulario y un modelo de lenguaje.

Cuando se crea y se entrena un usuario, se empieza con un conjunto de modelos estándar que después se adapta a la forma de hablar del usuario (modelo acústico) y al modo que éste usa las palabras (vocabulario y modelo de lenguaje relacionado). Al usar Dragon Naturally Speaking, el programa trabaja con los archivos personales de cada usuario para averiguar las palabras que pronuncia.

Este software cuenta con un kit para desarrolladores, el cual permite a los programadores incluir reconocimiento de voz utilizando la funcionalidad completa.

PERSONALIZACIÓN DE DRAGON

Antes de utilizarlo por primera vez, el usuario aporta información sobre su pronunciación.

A medida que se va usando, puede que el usuario se tope con errores de reconocimiento, ocasionalmente, pero dichos errores irán disminuyendo con el tiempo.

- Si el programa reconoce mal una palabra común, se puede corregir el error de reconocimiento, de forma que el programa va depurando la comprensión del modo en que el usuario pronuncia palabras o frases concretas.
- Si el programa reconoce mal una palabra poco común, como un término técnico o un nombre propio, se puede añadir esa palabra al vocabulario del programa para que, la siguiente vez que se pronuncie, el reconocimiento sea correcto.

REQUISITOS MÍNIMOS DEL SISTEMA

El proceso de instalación verifica que el sistema cumple con los requisitos mínimos:

CPU: Intel Pentium4 o AMD Athlon 64 1 GHz o superior. Se recomienda 2.4 GHz (1.6 GHz dual core) o el procesador AMD equivalente. (Nota: el set de instrucciones SSE2 es requerido.)

RAM: se recomienda un mínimo de 512 MB (1 GB para Windows Vista, and 2 GB para Windows 7). Se recomienda 1 GB (2 GB para Windows 7 32-bit, y 4 GB para Windows 7 64-bit).

Espacio de disco libre: 1 GB

Sistemas Operativos soportados:

- Windows Server 2000.
- Windows Server 2003.
- Windows 2000 SP4.
- Windows XP SP2 y SP3, 32 bit.
- Windows Vista SP1 y SP2, de 32 y 64 bits.
- Windows 7, de 32 y 64 bits.

LICENCIA

Es un producto propietario patentado, el cual sólo se puede adquirir a través de la compra del mismo.

JAVA SPEECH API

La Java Speech API fue desarrollada por Sun Microsystems Inc., en colaboración con compañías líderes en la tecnología de reconocimiento de voz como Apple Computer Inc., AT&T, Dragon Systems (empresa que originalmente desarrolló Dragon Naturally Speaking), IBM Corporation, Novell Inc., Philips Speech Processing y Texas Instruments Incorporated. Sun trabaja con otras compañías para fomentar la disponibilidad de distintas implementaciones (**Java Speech API – 2010, Speech Synthesis & Speech Recognition Using SAPI 5.1 – 2010**).

Define una interface estándar, fácil de usar, de software multiplataforma para la tecnología de voz. Soporta dos núcleos aplicaciones de voz: reconocimiento de voz y síntesis de voz.

Las interfaces de voz otorgan a los programadores de aplicaciones Java la oportunidad de implementar aplicaciones personalizadas y diferenciar sus productos.

La Java Speech API se desarrolló a través de un proceso de desarrollo abierto. Con la participación activa de compañías líderes en tecnología de voz, los aportes de los desarrolladores de aplicaciones, meses de revisiones y comentarios públicos, se pudo alcanzar un cierto grado de excelencia en la especificación. Además, Sun brindará soporte y mejorará la Java Speech API para mantener su capacidad.

Esta herramienta es una extensión de la plataforma Java. Las extensiones son paquetes de clases escritos en Java (y cualquier código nativo asociado), que los desarrolladores de aplicaciones pueden usar para extender la funcionalidad de la plataforma Java.

OBJETIVOS DE DISEÑO PARA LA JAVA SPEECH API

Junto con las otras Java Media APIs, la Java Speech API permite a los programadores incorporar interfaces de usuario avanzadas a las aplicaciones Java. Los objetivos de diseño son:

1. Proporcionar soporte para sintetizadores de voz y reconocedores de voz.
2. Proporcionar una robusta interfaz multiplataforma y multiproveedor para síntesis y reconocimiento de voz.
3. Soportar la integración con otras capacidades de la plataforma Java, incluyendo la Java Media API.

Una característica importante es que su diseño es simple, compacto y fácil de aprender.

CARACTERÍSTICAS DE LA API

Las capacidades existentes de la plataforma Java la hacen atractiva para el desarrollo de un amplio rango de aplicaciones. Con la adición esta API, los desarrolladores de aplicaciones Java pueden extender y complementar interfaces de usuario existentes con E/S de voz. Para los desarrolladores de aplicaciones de voz, la plataforma Java ofrece ahora una alternativa atractiva con:

- Portabilidad: el lenguaje de programación Java, las APIs y la máquina virtual están disponibles para una amplia variedad de plataformas hardware y sistemas operativos y son soportadas por los principales navegadores web.

- Ambiente potente y compacto: ya que provee a los programadores con un lenguaje potente, orientado a objetos y con "garbage collection" que permite un desarrollo rápido y de mayor fiabilidad.

EL RECONOCIMIENTO DE VOZ Y LAS JAVA APIS

La Java Speech API es una de las Java Media APIs, un conjunto de interfaces de software que proveen acceso multiplataforma a audio, video y otros archivos multimedia, gráficos 2D y 3D, animación, telefonía, imágenes avanzadas, y más. La Java Speech API, en combinación con otras Java Media APIs, permite a los programadores enriquecer las aplicaciones y applets Java con capacidades de comunicación que cumplen con las expectativas de los usuarios actuales, y mejoran la comunicación persona a persona.

Aprovecha las capacidades de otras Java APIs. Las características de internacionalización del lenguaje de programación Java más el uso del conjunto de caracteres Unicode simplifica el desarrollo de aplicaciones de voz multilingüe. Las clases e interfaces de la Java Speech API siguen los patrones de diseño de JavaBeans™. Por último, sus eventos se integran con los mecanismos de eventos de AWT, JavaBeans y la Java Foundation Classes (JFC).

REQUERIMIENTOS

Para usar la Java Speech API, un usuario debe tener un mínimo de software y hardware disponibles. Lo siguiente detalla algunas de estas necesidades (que si bien muchas son obvias, se las incluye también en el listado).

- **Software de voz:** se requiere un reconocedor o sintetizador de voz compatible con JSAPI.
- **Requerimientos del sistema:** la mayoría de los reconocedores de voz de escritorio y algunos sintetizadores requieren computadoras relativamente potentes para ejecutarse efectivamente.
- **Hardware de audio:** los sintetizadores de voz requieren salida de audio. Los reconocedores requieren entrada de audio. La mayoría de las computadoras de escritorio y portátiles actuales, proveen un buen soporte para audio. Los sistemas de dictado funcionan mejor con una placa de sonido de buena calidad.
- **Micrófono:** los sistemas de reconocimiento de voz de escritorio obtienen la entrada de audio a través del micrófono. Algunos reconocedores, especialmente sistemas de dictado, son sensibles al micrófono, y la mayoría de los productos de reconocimiento los recomiendan con ciertas características. Los micrófonos con auricular

incorporado generalmente proveen mejor rendimiento, especialmente en ambientes ruidosos. Los de mesa pueden ser usados en ciertos ambientes y para ciertas aplicaciones.

CONCLUSIÓN

En el presente capítulo, se ha cubierto la descripción de algunos softwares y frameworks para proveer reconocimiento de voz, lo cual permite junto con los capítulos anteriores, en los que se han incorporado distintos conceptos y herramientas, dejar las bases para capítulos posteriores.

A partir de lo presentado en este capítulo, se realizará la selección de uno de los softwares para ser utilizado en la realización del prototipo de integración que esta tesina tiene por objetivo. En el capítulo 7, se introducirá la herramienta seleccionada junto con la justificación del por qué de su elección. De dicho software se analizará y explicará, en profundidad, su estructura y funcionamiento interno.

CAPÍTULO 7

SPHINX 4

INTRODUCCIÓN

En el capítulo 6 se analizaron distintos softwares de reconocimiento de voz, se estudiaron sus principales características, sus funcionalidades y sus requisitos dando lugar al presente capítulo donde se presentará el software elegido para el prototipo de adaptación de este trabajo.

En este capítulo se analizarán las características principales del software elegido para integrar con JClic. Se revisará en profundidad cada uno de sus componentes y otros aspectos de interés para el presente trabajo, de manera de comprender su funcionamiento.

En la próxima sección se analizará el por qué de esta elección.

MOTIVACIÓN

La elección de Sphinx-4 para llevar a cabo el prototipo de integración fue motivada por varias razones, las cuales se detallan a continuación.

Es un framework ampliamente utilizado por investigadores y desarrolladores que se dedican al área del reconocimiento de voz, y que debido a esto, se encuentra en constante desarrollo y actualización.

Al ser un software de licencia BSD³⁴ es posible utilizarlo libremente en cualquier desarrollo e investigación. Además, también se puede obtener su código fuente, en caso de que sea necesaria alguna modificación, se desee mejorar alguna de sus características o estudiar su funcionamiento a bajo nivel.

Está completamente desarrollado con tecnología Java, al igual que JClic. De manera que se cree que se podrán integrar ambos componentes sin dificultades provocadas por incompatibilidad de lenguajes.

En la sección siguiente se comenzará a estudiar el framework elegido, analizando su estructura interna, módulo por módulo, de manera de comprenderlo en profundidad.

SPHINX-4

Es un sistema diseñado en la Universidad de Carnegie Mellon (CMU). Constituye uno de los más versátiles sistemas de reconocimiento de voz que existe actualmente (**Sphinx, sitio oficial - 2010**).

³⁴ Ver definición en Anexo B.

Fue creado en colaboración con Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL), y Hewlett Packard (HP), con contribuciones de la Universidad de California en Santa Cruz (UCSC) y el Massachusetts Institute of Technology (MIT).

En las siguientes secciones se estudiarán las características más importantes de este framework.

CARACTERÍSTICAS

Es un framework flexible, modular y fácil de incorporar a otras aplicaciones. De esta manera, ha ayudado a fomentar las innovaciones vinculadas a la investigación de los sistemas de reconocimiento, basados en los Modelos Ocultos de Markov.

Fue desarrollado en Java, lo cual hace muy práctico el desarrollo de aplicaciones multi-plataformas. Esto es uno de los cambios más importante que tiene Sphinx-4 respecto a versiones anteriores. Representa uno de los sistemas de reconocimientos de voz más completo, sólido y estable de la actualidad. Incluye muchas maneras diferentes de resolver un mismo problema, lo que lo convierte en un software flexible, capaz de ser utilizado en cualquier situación, desde aplicaciones para personas con algún tipo de discapacidad hasta laboratorios de investigación o dentro de un automóvil. Esta variedad de aplicaciones hace que sea un software altamente adaptable a cualquier ambiente.

Esto es una ventaja respecto a HTK (Hidden Markov Model Toolkit)³⁵, uno de los frameworks de reconocimiento de voz más consolidadas para los estudios universitarios. HTK ha sido programado en C, esto podría ser una desventaja a la hora de tener que exportar el software producido a una plataforma diferente de la utilizada en el desarrollo. Por eso Sphinx-4, al ser Java un lenguaje interpretado, y no un compilado como C, no se tiene necesidad de compilar un código para una plataforma específica.

La mera utilización de Java como lenguaje de programación, le da algunas ventajas:

- La gran variedad de APIs que existen para el desarrollo en lenguaje Java, reduce el tiempo de codificación.
- El soporte integrado para multi-thread hace más sencilla la división de tareas en varios hilos de ejecución.
- El Garbage Collector, que ayuda a que los programadores se concentren en el desarrollo del algoritmo y se olviden del manejo dinámico de memoria.

³⁵ Ver definición en Anexo B.

Como desventaja, la plataforma Java puede ocasionar problemas con la cantidad de memoria principal que un programa utiliza o referencia mientras se está ejecutando. También relacionado con la memoria, algunos motores de voz, acceden directamente a memoria para optimizar el rendimiento durante la decodificación. Con este lenguaje, el acceso directo a memoria no está permitido. A pesar de esto, se han obtenido buenos resultados en el desarrollo y el rendimiento de aplicaciones de reconocimiento de voz.

Este framework es un sistema basado en los Modelos Ocultos de Markov (HMM, Hidden Markov Model), y como todos los demás sistemas de reconocimiento de voz, para su funcionamiento primero debe aprender las características (o parámetros) de un conjunto de unidades de sonido, y luego utilizar lo que ha aprendido de estas unidades para encontrar la secuencia de unidades de sonido más probable para una señal de voz dada. El proceso de aprendizaje sobre las unidades de sonido se llama entrenamiento. El proceso de usar el conocimiento adquirido para deducir la secuencia de sonido más probable en una señal dada se llama decodificación, o simplemente reconocimiento.

En consecuencia, se necesitan dichos componentes del sistema Sphinx que se pueden usar para el entrenamiento y reconocimiento. En otras palabras, son necesarios el SPHINX Trainer y el SPHINX Decoder.

Otra ventaja de este framework es que es un software libre, lo cual evita el problema de tener que desarrollar un sistema completo desde cero. Además posee una estructura modular y de aplicación conectable que incorpora los patrones de diseño de sistemas existentes.

En resumen, es libre para ser usado y mejorado y su plataforma de trabajo se amplió debido a la tecnología Java.

Una última observación que se debe tener en cuenta en el marco de esta tesina, es que el desarrollo planteado será un prototipo de aplicación de software libre independiente de la plataforma, porque se ejecuta en una máquina virtual Java, pero la aplicación de la máquina virtual puede no ser libre.

ARQUITECTURA DE ALTO NIVEL

El framework ha sido diseñado con un alto grado de flexibilidad y modularidad, donde cada elemento en el sistema puede ser fácilmente reemplazado o modificado. De manera que para probar algún módulo del sistema no se necesita modificar otras partes.

La naturaleza modular está dada en principio por la utilización del lenguaje Java. En particular, la capacidad de la plataforma Java de cargar el código en tiempo de ejecución

permite soportar frameworks fácilmente adaptables, y además el dicho lenguaje construye interfaces que permiten separar el diseño del framework de la implementación.

A continuación se mostrará la arquitectura de Sphinx-4. Se respeta el inglés porque así lo hace la bibliografía específica, y además porque entendiendo las diferentes partes en inglés, luego resulta más fácil instanciar cada una, cuando se desee utilizar dicho framework ya que estos nombres se mantienen en la definición de las clases.

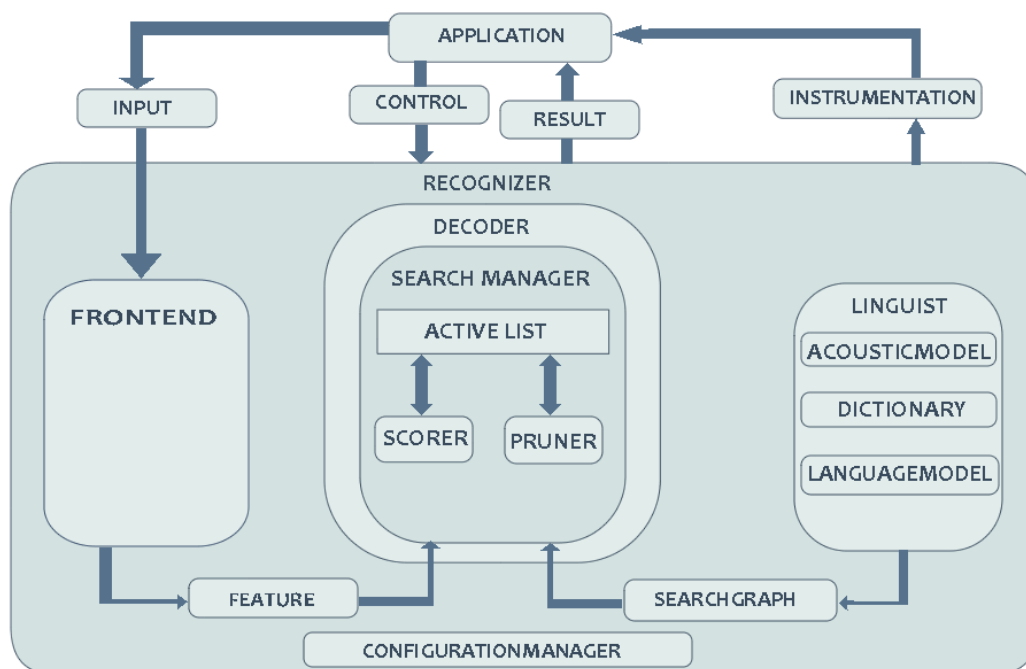


Figura 7.1 – Arquitectura de Sphinx-4.

Como puede verse en la figura 7.1, la arquitectura de alto nivel es relativamente sencilla. Desde el punto de vista de la aplicación, la forma de proceder para invocar al reconocedor es la siguiente: se deben introducir los datos de las señales en el búfer de entrada (Input), ajustar el registro de control (Control) y luego, tomar los resultados obtenidos desde el búfer de resultados (Result). Desde el punto de vista de la aplicación, la forma de trabajar del reconocedor es como la de una caja negra.

Desde el punto de vista del reconocedor, este obtiene los datos de entrada desde el búfer de entrada (Input), los procesa a través de los diversos módulos que componen su estructura interna y mueve los resultados al búfer de Resultados para ser leídos por la aplicación que invocó al reconocedor.

En principio, y de manera más abstracta, se puede decir que hay tres módulos importantes incluidos en el reconocedor. El **FrontEnd**, el **Decoder** y el **Linguist**.

El **FrontEnd** toma los datos de entrada de la señal digitalizada, establece los límites de cada uno de los datos (anotaciones), y parametriza en una secuencia de elementos a ser leídos

por el decodificador (**Decoder**). Las anotaciones que se mencionaron anteriormente, indican el comienzo y el fin de cada segmento de datos, y las operaciones realizadas incluyen, preenfásis (preemphasis), cancelación de ruidos, control automático de mejoras, puntos de fin, análisis de Fourier, filtros Mel-Cepstrum, y demás procedimientos propios de los métodos de reconocimiento de voz, utilizados para mejorar la señales de voz recibidas.

El **Linguist** traduce cualquier tipo de modelo de lenguaje (LanguageModel), junto con información de la pronunciación que toma del diccionario (Dictionary) e información estructural que toma de uno o más conjuntos de modelos acústicos (AcousticModel), y con esta información genera un SearchGraph. El SearchManager en el Decoder, utiliza los Features tomados del FrontEnd y el SearchGraph que le brinda el Linguist para realizar la decodificación, y generar los resultados.

En cualquier momento, antes o durante el proceso de reconocimiento, la aplicación puede facilitar Controls a cada uno de los módulos, mediante el ConfigurationManager, convirtiéndose en partícipe del proceso de reconocimiento. Estos eventos permiten que la aplicación monitoree y afine el proceso de decodificación.

El ConfigurationManager da al framework la habilidad de cargar y configurar módulos dinámicamente, en tiempo de ejecución, convirtiéndolo en un sistema flexible y adaptable.

FRONTEND

El propósito del FrontEnd es parametrizar una señal de entrada (por ejemplo, la voz) en una secuencia de salida (Features). Comprende una o más cadenas secuenciales de módulos de procesamiento de señales de voz que pueden ser reemplazados, y reciben el nombre de DataProcessors. El soporte de múltiples cadenas permite el cómputo simultáneo de diferentes tipos de parámetros sobre la misma señal de entrada. Esto permite la creación de sistemas que pueden decodificar simultáneamente usando diferentes tipos de parámetros.

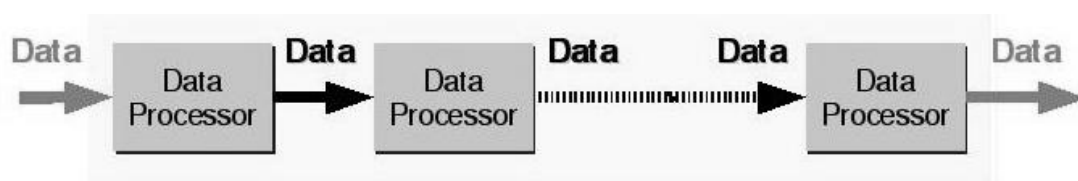


Figura 7.2 – Cadena de DataProcessors.

Cada DataProcessor tiene una entrada y una salida, que pueden estar conectadas a otros DataProcessor, permitiendo arbitrariamente tener largas secuencias de cadenas.

Las entradas y salidas de cada una de estas unidades, son objetos de datos genéricos que encapsulan datos de entrada procesados, así como también, marcadores que indican eventos de clasificación de datos tales como puntos de fin (EndPoint).

El último DataProcessor en la cadena, es responsable de producir un objeto de dato compuesto de señales parametrizadas, llamado Features, que será usado por el decodificador o Decoder.

Sphinx-4 tiene la capacidad de producir secuencias paralelas de Features, y permite un número arbitrario de corrientes (Streams) paralelas.

En este caso, la comunicación entre bloques utiliza el diseño llamado Pull. En un diseño de este tipo, un DataProcessor requiere la entrada desde un módulo anterior sólo cuando lo necesita, de manera contraria al diseño Push convencional, donde un módulo propaga su salida al módulo siguiente en el momento en que es generada.

Este diseño Pull permite a los procesadores realizar buffering, permitiendo a sus consumidores mirar hacia adelante y hacia atrás en el tiempo.

Dentro del framework FrontEnd, Sphinx-4 provee una suite de DataProcessors que implementan técnicas comunes de procesamiento de señales. Estas implementaciones incluyen: leer datos con varios formatos de entrada para modo de operación en lotes (por ejemplo, extraer el lenguaje hablado de una señal previamente grabada), leer desde un dispositivo de entrada de audio para un modo de operación en vivo (por ejemplo: extraer el lenguaje hablado utilizando un micrófono), preénfasis, etc.

Usando el ConfigurationManager es posible modificar la cadena de DataProcessors, como por ejemplo incorporar implementaciones de DataProcessors en cualquier diseño propio.

La naturaleza modular y adaptable de Sphinx-4, no sólo se aplica a la arquitectura de alto nivel, sino también a los módulos que esta incluye. Por ejemplo, el FrontEnd es un módulo adaptable, que también está compuesto por módulos adaptables.

El ConfigurationManger consiste básicamente de un código XML (eXtended Markup Language), el cual contiene toda la información para configurar el sistema correctamente.

```

<config>
  <property name="logLevel" value="WARNING" />
  <component name="epFrontEnd" type="edu.cmu.sphinx.frontend.FrontEnd" >
    <propertylist name="pipeline" >
      <item>microphone </item>
      <item>speechClassifier </item>
      <item>speechMarker </item>
      <item>nonSpeechDataFilter </item>
      <item>preemphasizer </item>
      <item>windower </item>
      <item>fft </item>
      <item>melFilterBank </item>
      <item>dct </item>
      <item>liveCMN </item>
      <item>featureExtraction </item>
    </propertylist>
  </component>
</config>

```

Figura 7.3 – Ejemplo de archivo ConfigurationManager.

En las siguientes secciones se analizará brevemente cada uno los ítems que se muestran en la figura anterior.

MICROPHONE

Un microphone (como componente del FrontEnd) captura el dato de audio desde un sistema de audio subyacente y convierte estos datos de audio en objetos de datos (Data Objects).

```

<component name="microphone" type="edu.cmu.sphinx.frontend.util.Microphone" >
  <property name="closeBetweenUtterances" value="false" />
</component>

```

Figura 7.4 – Definición de microphone en el configurationManager.

Intentará obtener un fragmento de audio con el formato especificado en el archivo de configuración. Si no se obtiene tal información con el formato indicado, tratará de obtener uno con un formato de audio que tenga una tasa de muestreo más alta que la tasa de muestreo configurada, de todas formas se mantiene el resto de los otros parámetros de formato (Por ejemplo, el tamaño de la muestra, la señal, el canal, etc.). Si no encuentra ningún tipo de información se coloca un flag de error.

Existe una propiedad configurable para este periférico que especifica si el micrófono enviará, o no, el audio entre expresiones. En algunos sistemas (GNU/Linux es uno de ellos),

el cierre y reapertura de audio no funciona muy bien, es por esta razón que por defecto esta propiedad se establece en falso.

SPEECHCLASSIFIER

Esta clase implementa un nivel de seguimiento de “Endpointer” inventado por Bent Schmidt-Nielsen³⁶. Esta técnica de “Endpointer” está compuesta por tres pasos:

1. Clasificar el audio en lo que es voz y lo que no lo es.
2. Insertar etiquetas SPEECH_START y SPEECH_END donde comience y termine la señal de la voz, respectivamente
3. Eliminar las regiones que no incluyen voz.

El primer paso, clasifica el audio en lo que es voz y lo que no, para realizar esta tarea, usa el algoritmo de Bent Schmidt-Nielsen. Cuando ingresa el audio el promedio del nivel de la señal y el ruido de fondo se actualizan, usando el nivel de señal del audio actual. Si el promedio del nivel de la señal es mayor que el nivel de ruido de fondo estipulado por un valor umbral configurable, entonces el audio actual es marcado como señal de voz. De lo contrario, se marca como que no representa voz. Un valor más bajo hará que el “Endpointer” sea más sensible, es decir, se va a marcar más audio como voz. Un umbral más alto hará que el “Endpointer” sea menos sensible, es decir, marca menos audio como voz.

```
<component name="speechClassifier"
  type="edu.cmu.sphinx.frontend.endpoint.SpeechClassifier" >
  <property name="threshold" value="13" />
</component>
```

Figura 7.5 – Definición de speechClassifier.

El segundo y tercer paso del “Endpointer” se encuentran documentados en las clases SpeechMarker y NonSpeechDataFilter, que se estudiarán en las siguientes secciones.

SPEECHMARKER

Particularmente, esta clase convierte una secuencia de objetos SpeechClassifiedData, marcados como voz y no voz, y marca las regiones que son consideradas voz. Esto se realiza insertando señales SPEECH_START y SPEECH_END en una secuencia de entrada.

³⁶ MERL Research / Technical Staff - Team Leader Speech Group - B.S. University of California at San Diego, 1971

```
<component name=" speechMarker"  
|         type=" edu.cmu.sphinx.frontend.endpoint.SpeechMarker" >  
|     <property name=" speechTrailer" value=" 50" />  
| </component>
```

Figura 7.6 – Definición de speechMarker.

El algoritmo está siempre en uno de dos estados: “in-speech” y “out-of-speech”.

Si se encuentra en el estado ‘out-of-speech’, leerá audio hasta que se alcance una señal de voz. Si la cantidad de habla que recibe es mayor que el “startSpeech”, se considera que se comenzó a hablar, y se inserta un SPEECH_START como “speechLeader” antes de que comience el discurso. El estado del algoritmo cambia a “in-speech”.

Ahora, se considerará el caso en que el algoritmo esté en estado “in-speech”. Si lee un audio que es considerado de voz, será emitido. Si el audio no es voz, se lee hasta que se adquiere una cantidad de “endSilence” mayor al sonido continuo que no es voz. En este punto se considera que la voz ha terminado y se inserta un “SPEECH_END” después de cumplido el tiempo del “speechTrailer” (indica la cantidad de tiempo, en milisegundos, que se incluirá como datos de voz, después de finalizado el discurso). El algoritmo retorna al estado “out of speech”. Si se encuentra en el medio algún audio de voz, la cuenta comienza nuevamente.

NONSPEECHDATAFILTER

La clase NonSpeechDataFilter, dada una secuencia de objetos de datos, filtra las regiones marcadas como “sin voz”. Dicha secuencia debería tener con y sin voz marcadas por las clases SpeechStartSignal y SpeechEndSignal, utilizando la clase SpeechMarker.

Tal secuencia de datos para una expresión debe ser similar a uno de los siguientes casos:

1. Sólo una región con voz. En este caso, el “stream” tiene sólo una región de voz. Luego, cuando se aplica el filtro, se remueven las regiones sin voz.

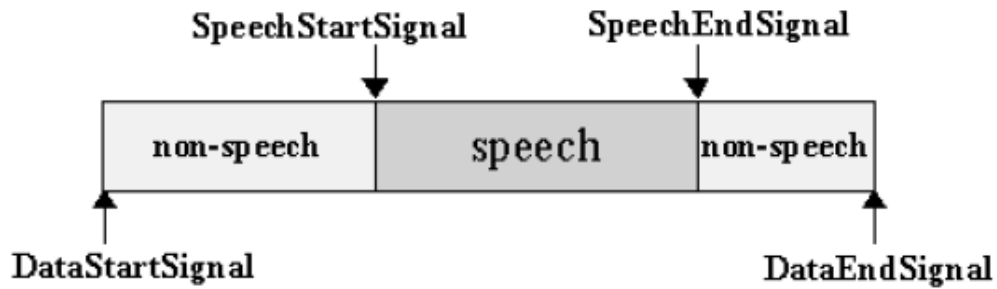


Figura 7.7 – Ejemplo de “stream” con solo una región de voz.

Donde el “stream” resultante sólo contiene regiones de voz, el cual se ilustra en la siguiente figura.

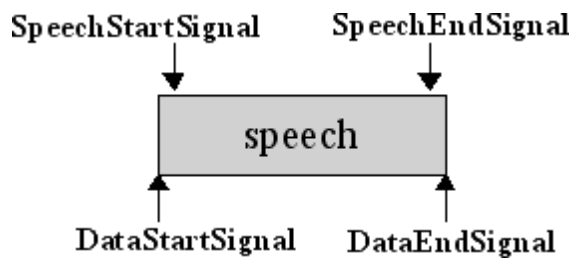


Figura 7.8 - Ejemplo de “stream” con solo una región de voz filtrada.

- Múltiples regiones con voz. Este caso es más complicado que si se tiene una sola región con voz. La propiedad “mergeSpeechSegments” es muy importante en el control del comportamiento de este filtro. Esta propiedad determina si las regiones de voz (y las regiones sin voz que hay entre ellas) podrían ser unidas en una región más grande, o si las regiones de voz son convertidas en expresiones individuales.

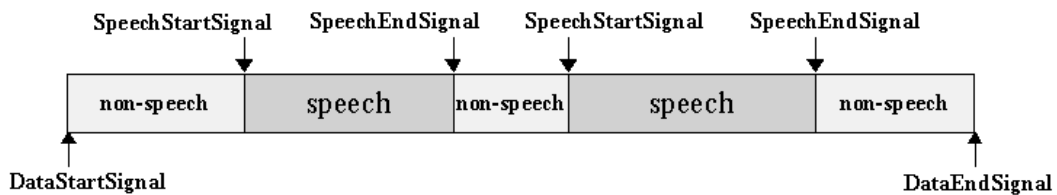


Figura 7.9 – Ejemplo de un “stream” con dos regiones de voz.

Si la propiedad “mergeSpeechSegments” es verdadero, todos los objetos de datos desde el primer SpeechStartSignal hasta el último SpeechEndSignal serán considerados como una única expresión y estarán encerrados entre etiquetas de DataStartSignal y DataEndSignal. Todas las regiones que no contienen voz, tales como SpeechStartSignals y SpeechEndSignals se eliminan de la secuencia.

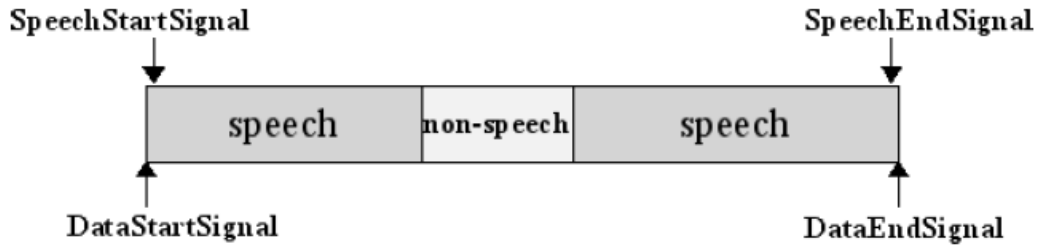


Figura 7.10 – Ejemplo de un “stream” con dos regiones de voz después del filtrado cuando “mergeSpeechSegments” es verdadero.

Por otra parte, si “mergeSpeechSegments” se establece en false (valor predeterminado), entonces cada región de voz se convertirá en un “stream” independiente.

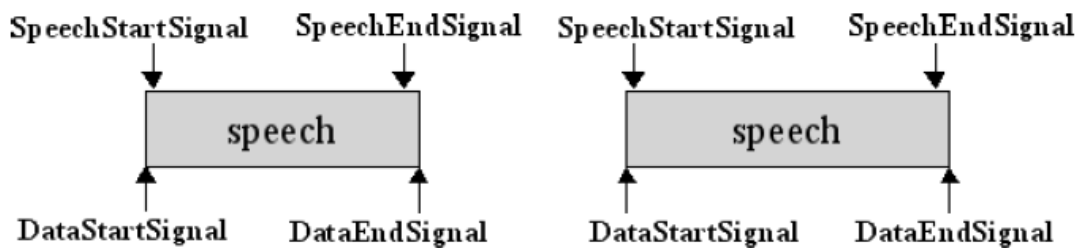


Figura 7.11 – Ejemplo de un “stream” con dos regiones de voz después del filtrado cuando “mergeSpeechSegments” es falso.

PREEMPHASIZER

Esta clase implementa un filtro “high-pass” (o pasa altos) que compensa la atenuación en el dato de sonido. Las señales de la voz tienen una atenuación (un decremento en la intensidad de la señal) de 20 dB/dec³⁷. Aumenta la magnitud relativa de las frecuencias más altas con respecto a las frecuencias más bajas ya que normalmente contienen mucha menos energía, aunque siguen siendo importantes para el reconocimiento de voz.

Dicha clase toma un objeto de dato que generalmente representa datos de audio como entrada, y la salida es el mismo objeto pero con el proceso de Preemphasis aplicado.

³⁷ Ver definición en Anexo B.

WINDOWER

Esta clase divide el objeto de datos en un número de ventanas superpuestas (usualmente llamadas frames). Con el objetivo de minimizar las discontinuidades de la señal en los límites de cada trama (frame), cada trama se multiplica por una función ventana de coseno. Sin embargo, el sistema usa ventanas superpuestas para capturar la información que puede ingresar en los límites de la ventana. Estos eventos no podrían estar bien representados si las ventanas estuvieran simplemente yuxtapuestas.

El número de ventanas resultantes depende del tamaño de la ventana y del tamaño de la parte que se solapa (comúnmente conocido como frame shift).

FFT(FAST FOURIER TRANSFORM)

Esta clase calcula la transformada de Fourier discreta (DFT) de una secuencia de entrada, utilizando el algoritmo de la Transformada Rápida de Fourier (FFT). La Transformada de Fourier (FT) es el proceso de análisis de una señal por sus componentes de frecuencia.

La transformada de Fourier Discreta (DFT) (como ya fue explicado en el capítulo 4) es la representación discreta de la transformación de Fourier general y, la Transformada Rápida de Fourier, es un algoritmo optimizado en términos de carga computacional y tiempo de procesamientos para secuencias cuyo tamaño corresponde a una potencia de dos.

El discurso es analizado a una tasa constante por una función de ventana. Esta ventana es el producto de aplicar una ventana deslizante de Hamming a la señal. Además, dado que la amplitud es mucho más importante que la fase de reconocimiento de voz, esta clase retorna la potencia del spectrum de una ventana de datos en lugar del spectrum complejo. Cada valor en el espectro retornado representa la fuerza de esa frecuencia, en particular, para esa ventana de datos.

Por defecto, el número de puntos de FFT es la potencia de dos más próxima (que es igual o mayor que el número de muestras en la ventana de datos de entrada). La longitud de la potencia retornada es el número de puntos de FFT, dividido dos, más uno. Dado que la señal de entrada es real, la FFT es simétrica, y la información contenida en todo el vector ya se encuentra presente en su primera mitad.

MELFILTERBANK

La presente clase filtra como entrada una potencia spectrum a través de un banco de cierto número de filtros mel (mel-filters). La salida es un arreglo de valores filtrados, llamados típicamente mel-spectrum, cada uno correspondiente al resultado de filtrar la entrada con un filtro individual. Por lo tanto, la longitud del arreglo de salida es igual al número de filtros creados.

DCT

Esta clase aplica un logaritmo y luego una Transformada de Coseno Discreta (DCT) a los datos de entrada, los que normalmente son los mel spectrum obtenidos por el módulo anterior. Ha sido probado que, para una secuencia de números reales, la DCT es equivalente a la Transformada de Fourier Discreta (DFT). Esta clase corresponde a la última etapa de la conversión de una señal a cepstra, definida como la Transformada de Fourier inversa del logaritmo de la Transformada de Fourier de una señal. Las dimensionalidades de los coeficientes que se retornan por defecto son 13. Cuando la entrada es mel-spectrum, el vector retornado es el vector MFCC (Mel-Frecuencia Cepstral Coefficient), donde el elemento 0 es el valor de energía.

LIVECMN

Esta clase extrae la media de todos los Objetos de Datos de entrada. CMN son las siglas de Cepstral Mean Normalization (Normalización de media de Cepstral). No lee la entrada de Objetos de Datos completamente antes de calcular la media, sino que estima la media desde ventanas previas y lo extrae desde los Objetos de Datos en tiempo real.

Por lo tanto, no se presenta ningún retraso por el uso de “liveCMN”.

Las propiedades del Sphinx-4 que afectan este procesador están definidas por el promedio inicial cepstral que está establecido en 12, la medida de la ventana “liveCMN” que está establecida en 100 y la ventana cambiante de CMN, que especifica después de cuántos cepstrum se recalcula la media cepstral, y su valor por defecto es 160, lo cual nuevamente es un valor interno no limitado por una unidad específica.

cmnWindow

CmnWindow + número de frames desde la última recalculación

Figura 7.12 – Ecuación CMN.

La media de todos los cepstrum de entrada calculados hasta un momento no se reestima por cada cepstrum. Esta media se recalcula después de cada ventana cambiante cepstra de CMN y se estima dividiendo la suma de todos los cepstrum de entrada ya procesados. Luego de obtener la media, la suma decae exponencialmente multiplicándose por el radio dado por la ecuación mostrada en la figura 7.12.

Este proceso se realiza en el contexto de la clase `DeltasFeatureExtractor` que computa el delta o delta doble de cepstrum de entrada. El delta es el derivado de primer orden y el doble delta (también conocido como delta delta) es el derivado de segundo orden del cepstrum original. Estos ayudan a modelar la dinámica de las señales del discurso (por ejemplo velocidad y aceleración). El dato de salida es un objeto `FloatData` con un arreglo de decimales, de longitud igual al triple del cepstrum original, formado por la concatenación de cepstra, delta cepstra y doble delta cepstra. La salida es el vector de Features usado por el decodificador. La figura 7.13 muestra la disposición del arreglo de salida.

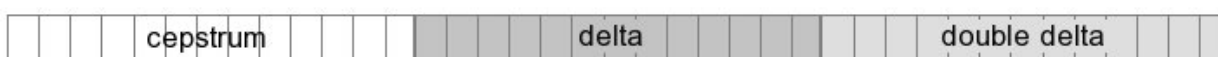


Figura 7.13: Diseño de las características retornadas/devueltas.

Suponiendo que el cepstrum original tiene una longitud N , entonces los primeros N elementos del arreglo son el cepstrum original, los segundos N elementos son el delta cepstrum, y los últimos N elementos son el doble delta cepstrum.

La figura 7.14 ilustra el cálculo/cómputo del delta y doble delta de un vector de cepstrum, usando las últimas 3 cepstra y las próximas 3 cepstra.

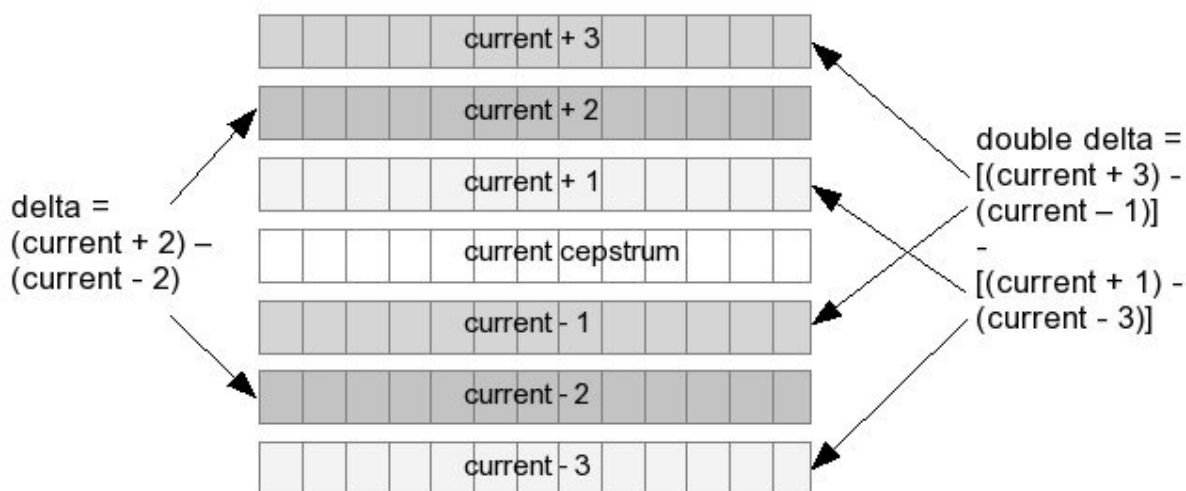


Figura 7.14 Cálculo/Cómputo del vector Delta y Doble Delta.

Refiriéndonos a la Figura 7.14, el delta se calcula restando el cepstrum, que está dos frames más atrás que el cepstrum actual, del cepstrum que está dos frames adelante del cepstrum actual.

El cálculo del Doble Delta es similar. Se calcula restando el delta cepstrum de un frame más atrás al delta cepstrum de un frame más adelante. Reemplazar delta cepstra por cepstra,

lleva a una fórmula que involucra al cepstra, que está 1 y 3 antes y después del cepstrum actual.

LINGUIST

El Linguist genera el SearchGraph (grafo de búsqueda) que es usado por el Decodificador durante la búsqueda, mientras que al mismo tiempo oculta las complejidades involucradas en la generación de este grafo.

Una típica implementación del Linguist construye el SearchGraph usando la estructura de lenguaje como el representado en el LanguageModel dado y la estructura topológica del AcousticModel. El Linguist puede usar también un Diccionario para mapear palabras del LanguageModel con secuencias de elementos de AcousticModel. Cuando se genera el SearchGraph, el Linguist puede también incorporar sub-palabras de longitud arbitraria, si es proporcionada.

Sphinx-4 provee una implementación del Linguist que representa estáticamente el espacio de búsqueda como un grafo plano, donde cada palabra en el vocabulario tiene su propia rama, y su nombre es la clase FlatLinguist. La clase FlatLinguist toma un grafo de la gramática (devuelto por la gramática configurable subyacente), y genera un grafo de búsqueda para esta gramática.

LANGUAGEMODEL

El módulo LanguageModel del Linguist provee una estructura de lenguaje a nivel palabra, el cual puede ser representado por cualquier número de implementaciones adaptables. Estas implementaciones generalmente pertenecen a una de las siguientes categorías: gramáticas basadas en gráficos y modelos estocásticos N-Gram. La primera representa un grafo dirigido de palabras, donde cada nodo representa una palabra y cada arco representa la probabilidad de que exista una transición entre palabras. Los modelos estocásticos N-Gram proveen probabilidades de palabras dada la observación de las n-1 palabras previas.

DICTIONARY

El Dictionary (Diccionario) provee las pronunciaciones de las palabras que se encuentran en el LanguageModel. Las pronunciaciones parten las palabras en secuencias de sub-palabras que se encuentran en el AcousticModel. La interface del Dictionary también soporta la clasificación de palabras y permite a una palabra estar en múltiples clases.

Sphinx-4 actualmente proporciona implementaciones de la interface del Dictionary para admitir el CMU Pronouncing Dictionary (Diccionario de Pronunciación).

El módulo AcousticModel proporciona un mapeo entre una unidad de discurso y un HMM que puede ser marcado contra las componentes (Features) de entrada proporcionadas por el FrontEnd. Como con otros sistemas, el mapeo también puede tener en cuenta información contextual y de posición de la palabra. Por ejemplo, en el caso de los trifonemas, el contexto representa los fonemas simples a la izquierda y a la derecha del fonema dado, y la posición de la palabra representa si el trifonema está al principio, al medio o al final de la palabra(o si es una palabra en sí mismo).

La definición contextual no está fijada por el Sphinx-4, teniendo en cuenta la definición de los AcousticModels que contienen alófonos³⁸ así como también los AcousticModels cuyos contextos no necesitan ser contiguos a la unidad.

Generalmente, el Linguist parte cada palabra del vocabulario en una secuencia de unidades dependientes del contexto. Entonces, el Linguist pasa las unidades y sus contextos al AcousticModel, recuperando los grafos HMM asociados a esas unidades. Luego usa estos grafos HMM conjuntamente con el LanguageModel para construir el SearchGraph.

A diferencia de otros sistemas de reconocimiento de voz, que representan los grafos de HMM como una estructura fija en memoria, el HMM de Sphinx-4 es simplemente un grafo dirigido de objetos. Cada nodo corresponde a un estado del HMM y cada arco representa la probabilidad de pasar de un estado a otro en el HMM. Al representar al HMM como un grafo dirigido de objetos, en lugar de representarlo como una estructura fija, una implementación del AcousticModel puede suministrar fácilmente diferentes topologías a los HMM. Por ejemplo, las interfaces del AcousticModel no restringen a los HMMs en cuanto a número de estados, el número transiciones de cualquier estado, o la dirección de una transición (hacia adelante o hacia atrás). Además, Sphinx-4 permite que el número de estados varíe de una unidad a otra en el mismo AcousticModel.

Cada estado del HMM es capaz de producir una puntuación desde una característica (Feature) observada. El código actual para calcular la puntuación está realizado por el estado HMM mismo, ocultando así su implementación del resto del sistema.

SEARCHGRAPH

Sin importar como esté implementado el Linguist, los espacios de búsqueda son todos representados como un SearchGraph, ilustrado en la Figura 7.15. Este es la estructura de datos primaria usada durante el proceso de decodificación. Está compuesta por SearchStates opcionalmente emitidos y SearchStateArcs con probabilidades de transición. Cada estado en el grafo puede representar componentes del LanguageModel (palabras en

³⁸ Ver definición en Anexo B.

rectángulos), Dictionary (unidades de la palabra en círculos oscuros) o AcousticModel (HMMs).

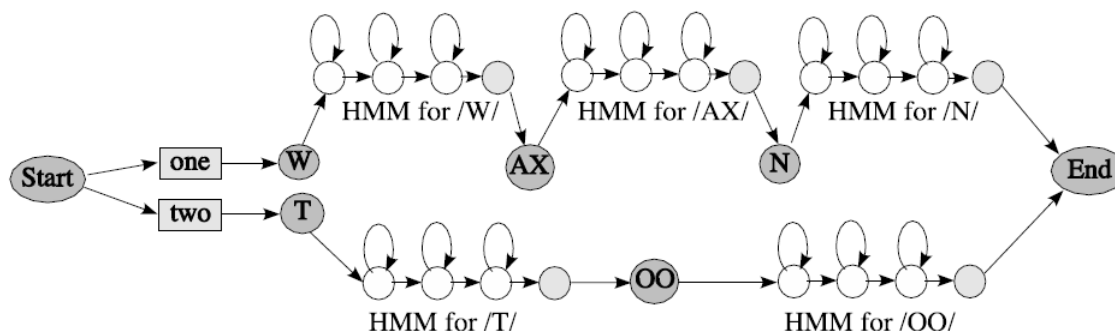


Figura 7.15 - Ejemplo de SearchGraph.

El SearchGraph es un grafo dirigido donde cada nodo, llamado SearchState (un estado de búsqueda), representa tanto estados emitidos como estados no emitidos. Los estados emitidos pueden ser marcados contra componentes acústicos de entrada mientras que los estados no emitidos se utilizan generalmente para representar modelos lingüísticos de mayor nivel como palabras y fonemas que no están directamente marcados contra componentes de entrada. Los arcos entre estados representan los posibles estados de transición, cada uno de los cuales tiene una probabilidad para representar la posibilidad de hacer la transición a lo largo del arco.

La interface es deliberadamente genérica para permitir una amplia gama de opciones de implementación, liberando los supuestos y limitaciones del cableado (hard-wired) encontradas en sistemas previos de reconocimiento. En particular, el Linguist no tiene restricciones inherentes sobre lo siguiente:

- Topología del espacio total de búsqueda.
- Tamaño del contexto fonético.
- Tipo de gramática (estocástica o basada en reglas).
- Profundidad del modelo del lenguaje N-Gram.

Una característica fundamental del SearchGraph es que la implementación del SearchState no necesita ser fijo. Como tal, cada implementación del Linguist generalmente proporciona su implementación concreta del SearchState que puede variar en función de los componentes del Linguist particular. Por otro lado, un Linguist representa un vocabulario muy grande y complejo, sin embargo, puede construir una representación compacta interna del SearchGraph y dinámicamente expandir esta representación compacta según la demanda. La elección entre la construcción dinámica y estática del lenguaje depende mayoritariamente en el tamaño del vocabulario, complejidad del modelo de lenguaje y la huella (memoria que utiliza o referencia el software) deseada en la memoria del sistema.

El rol fundamental del bloque Decoder es usar los Features del FrontEnd conjuntamente con el SearchGraph del Linguist para generar hipótesis de resultado (Result). El bloque Decoder se compone de un SearchManager y un código de soporte que simplifica el proceso de decodificación para una aplicación. Así, el componente más interesante del bloque Decoder resulta ser el SearchManager.

El Decoder simplemente le informa al SearchManager que reconozca un conjunto de frames de Feature. En cada paso del proceso, el SearchManager crea un objeto Result que contiene todos los recorridos que han terminado en un estado no-emitado. Para procesar el resultado, Sphinx-4 también proporciona utilidades capaces de producir un retículo y reportes confiables del Result.

Como el Linguist, el SearchManager no está restringido a ninguna implementación en particular. Por ejemplo, implementaciones del SearchManager pueden realizar algoritmos de búsqueda como frame sincrónico Viterbi, A^* , bidireccionales, etc.

La implementación del SearchManager usa un algoritmo de “token passing”. Un token es un objeto que está asociado con un SearchState y contiene la acústica y reportes del lenguaje generales del recorrido en un punto dado, una referencia al SearchState, una referencia a un frame Feature de entrada, y otra información relevante. La referencia al SearchState le permite al SearchManager relacionar un token con su distribución de salida de estado, unidad de fonética dependiente del contexto, pronunciación, palabra y estado gramatical. Cada hipótesis parcial finaliza en un token activo.

Como es una técnica común, Sphinx-4 proporciona un sub-framework para respaldar al SearchManager compuesto de un ActiveList, un Pruner y un Scorer. Este sub-framework genera ActiveLists de los tokens que están actualmente activos en la cuadrícula de búsqueda mediante el método de poda haciendo uso del Pruner.

La implementación del Pruner es enormemente simplificada por el garbage collector de la plataforma Java. Con el garbage collector, el Pruner puede cortar un camino completo simplemente eliminando el token terminal del camino del ActiveList. El acto de la eliminación del token terminal identifica el token y cualquier token no compartido para ese camino como inactivo, permitiendo que el garbage collector recupere la memoria asociada.

El sub-framework del SearchManager también se comunica con el Scorer, un módulo de estimación de probabilidad de estado que proporciona valores de densidad de salida de estado según la demanda. Cuando el SearchManager solicita un reporte para un estado dado, a un determinado tiempo, el Scorer accede al vector de componentes para ese momento y realiza las operaciones matemáticas para calcular el reporte. El Scorer retiene la información perteneciente a las densidades del estado de salida. Así, el SearchManager no

necesita saber si el reporte fue hecho con HMMs continuos, semi-continuos o diferenciados.

ENTRENAMIENTO Y CONFIGURACIÓN

En esta sección se presentará una introducción a la herramienta que se provee para realizar el entrenamiento y se describirán brevemente los archivos necesarios para la configuración del framework.

ENTRENAMIENTO

Es posible entrenar Sphinx-4 de manera que reconozca las palabras o frases que se desean, además de obtener una probabilidad de acierto mayor en el reconocimiento. Además Sphinx por defecto está disponible para el idioma inglés y si se desea que se reconozca un vocabulario en español se deberá entrenarlo para que esto sea posible. Aunque hoy en día existen modelos, ya entrenados, disponibles en español.

Para realizar el entrenamiento, la Universidad de Carnegie Mellon (CMU), provee una herramienta, llamada SphinxTrain, la cual permite generar archivos para la configuración de Sphinx-4 que permitirán reconocer en el idioma deseado.

SphinxTrain genera modelos acústicos con el formato utilizado por Sphinx-3, aunque es posible utilizar estos archivos para entrenar Sphinx-4.

ARCHIVO DE CONFIGURACIÓN

Como ya ha sido mencionado, Sphinx-4 utiliza un archivo de configuración, el cual tiene dos objetivos principales³⁹:

- Determinar que componentes se van a usar en el sistema: debido a que es un framework extremadamente flexible es posible reemplazar un componente por otro en ejecución.
- Determinar la configuración determinada de cada uno de estos componentes: como la mayoría de los sistemas de reconocimiento tiene un gran número de parámetros que controlan el funcionamiento del sistema.

³⁹ Información sobre el archivo de configuración - <http://cmusphinx.sourceforge.net/sphinx4/javadoc/edu/cmu/sphinx/util/props/doc-files/ConfigurationManagement.html>

Además, este archivo sirve para indicarle al reconocedor todo acerca de cómo se desea que funcione el reconocimiento. Para ello, define lo siguiente:

- El nombre y tipo de todos los componentes del sistema.
- La forma de conectarse de estos componentes, en otras palabras, cuáles componentes interactúan con otros.
- La configuración detallada de cada componente.

Se denomina *properties* (o propiedades) a la configuración de un componente. Estas propiedades varían dependiendo del componente que se está definiendo.

DICCIONARIO

Es un archivo que contiene la correspondencia entre las palabras que se desea que se reconozcan y su transcripción fonética.

En la siguiente sección se nombrarán algunas aplicaciones que utilizan Sphinx-4.

APLICACIONES QUE USAN SPHINX

Existen actualmente una gran cantidad de aplicaciones que utilizan Sphinx, en alguna de sus versiones. A continuación se nombrarán algunas de ellas (**Listado de aplicaciones basadas en Sphinx4 – 2010**).

- **Lumenvox:** compañía de software comercial que utiliza una versión modificada de Sphinx-2 para proveer una solución de reconocimiento de voz orientada a la Respuesta de Voz (IVR).
- **Cairo:** provee un Media Resource Control Protocol Version 2 (MRCPv2) el cual utiliza Sphinx-4.
- **Asterisk plugin:** integra PocketSphinx con Asterisk.
- **Freeswitch:** es un software alternativo a Asterisk, que utiliza PocketSphinx para proveer reconocimiento de voz.
- **jvoicexml:** interprete Free VoiceXML para Java con una arquitectura abierta que permite extensiones personalizadas.
- **Gnome-Voice-Control:** es un sistema de diálogo que se utiliza para controlar el GNOME Desktop.

- **perlbox:** provee soluciones de reconocimiento de voz para poder controlar el escritorio de Linux y Unix.
- **Zanzibar:** es un software de Respuesta de Voz Interactiva (IVR) de código abierto.
- **ASTRA (Advanced Sphinx Trainer):** es un proyecto desarrollado bajo licencia GPL v3 para entrenar modelos CMU Sphinx-3.
- **Arabisc:** es un reconocedor de voz continua para lengua árabe independiente del hablante, publicado bajo licencia GNU. Es también una colección de herramientas de código abierto que permite a los investigadores y desarrolladores para construir sistemas de reconocimiento de voz para árabes.
- **Voicekey:** teclado controlado a través del habla para sistemas GNU/Linux que utiliza Sphinx-3.
- **Magnus:** software libre de reconocimiento de voz para controlar la funcionalidad del mouse a través de comandos de voz, disponible en Catalán utilizando Sphinx-4.

CONCLUSIÓN

En el presente capítulo se comenzó presentando el software de reconocimiento de voz elegido para llevar a cabo el prototipo de integración, para luego introducir algunas de las características más importantes de este framework. Las cuales proporcionan un entendimiento del funcionamiento de Sphinx-4 necesario para el presente trabajo de investigación.

Además, se introdujo al lector al mecanismo de entrenamiento del mismo, haciendo mención y una breve descripción de las herramientas necesarias. Este tema se analizará en profundidad en el capítulo 9, en el cual, también se expondrá los pasos realizados para lograr un prototipo de aplicación, incluyendo el código que se debió agregar para integrar ambos componentes, como la configuración y entrenamiento de Sphinx-4.

CAPÍTULO 8

JCLIC

INTRODUCCIÓN

Habiendo realizado un análisis de diferentes softwares de autor en el capítulo 3, y resaltado las distintas características de los mismos, se llegó a la elección de la herramienta a integrar con comandos por voz.

Para la elección, se tuvo en cuenta que sea un software con uso en la comunidad educativa, lo que conlleva a encontrar un programa que permita la posibilidad de elaborar distintas actividades que se pueden aplicar en diferentes niveles, dependiendo del contenido con el que se construyan. Además se tuvo en cuenta que sea multiplataforma de manera que sea posible utilizarlo en distintos ambiente, y que su código fuente esté disponible para poder llevar a cabo la adaptación. También se ha tenido en cuenta la experiencia de las autoras en el uso de estos softwares.

En el presente capítulo, se introducirá el software que se eligió para integrar con reconocimiento por voz. De manera que, en las siguientes secciones, se estudiarán sus características, objetivos, componentes, estructura y otros aspectos de interés para el trabajo.

En el siguiente apartado se presentará el software elegido y el porqué de esta elección.

MOTIVACIÓN

Para la realización del prototipo que esta tesina tiene como objetivo se eligió JClíc⁴⁰, un proyecto de software libre, que se distribuye bajo los términos de la Licencia Pública General de GNU (GPL).

El proyecto de desarrollo de JClíc se encuentra alojado en el portal LaFarga.org, del Departamento de Universidades, Investigación y Sociedad de la Información (DURSI) de la Generalitat de Cataluña.

Algunas de las características que llevaron a la elección de este software, fueron que está desarrollado bajo licencia GPL, lo que provee la oportunidad de contar con el código fuente del programa para poder estudiarlo y analizarlo. De esta manera, se puede llevar a cabo la integración propuesta, la cual se analizará en el capítulo 9, el cual enmarca los detalles de la integración, dificultades encontradas y demás características.

También es uno de los software más utilizados para realizar actividades educativas (ya lleva años de uso en el ámbito), de manera que se creyó que sería interesante comenzar el desarrollo de un prototipo para ampliar la diversidad de usuarios de dichas actividades.

⁴⁰ Página oficial de JClíc - <http://clíc.xtec.cat/es/jclíc/index.htm>

Otro punto fuerte en esta elección es que es posible utilizar JClic en distintos sistemas operativos, tales como Windows, Linux, Solaris y Mac OS X.

Esta cualidad viene dada porque JClic se encuentra enteramente desarrollado con tecnología Java, la cual es multiplataforma.

¿QUÉ ES JCLIC?

JClic es un entorno para la creación, realización y evaluación de actividades educativas multimedia. Está formado por un conjunto de aplicaciones informáticas que sirven para realizar diversos tipos de actividades educativas: rompecabezas, asociaciones, ejercicios de texto, palabras cruzadas, etc. (**JClic, sitio oficial – 2010**).

Las actividades, por lo general, no se presentan solas, sino empaquetadas en proyectos. Un proyecto está formado por un conjunto de actividades y una o más secuencias, que indican el orden en que se han de mostrar.

JClic está desarrollado en la plataforma Java, es un proyecto de código abierto y funciona en diversos entornos y sistemas operativos.

VERSIÓN Y OBJETIVOS

El proyecto JClic es una evolución del programa Clic 3.0, una herramienta para la creación de aplicaciones didácticas multimedia con más de 10 años de historia. A lo largo de este tiempo, han sido muchos los educadores que lo han utilizado para crear actividades interactivas, donde se trabajan aspectos procedimentales como diversas áreas del currículum, desde educación infantil hasta secundaria.

Los objetivos perseguidos al iniciar el proyecto eran:

- Hacer posible el uso de aplicaciones educativas multimedia "en línea", directamente desde Internet.
- Mantener la compatibilidad con las aplicaciones Clic 3.0 existentes.
- Hacer posible su uso en diversas plataformas y sistemas operativos, como Windows, GNU/Linux, Solaris o Mac OS X.
- Utilizar un formato estándar y abierto para el almacenaje de los datos, con el fin de hacerlas transparentes a otras aplicaciones y facilitar su integración en bases de datos de recursos.
- Ampliar el ámbito de cooperación e intercambio de materiales entre escuelas y educadores de diferentes países y culturas, facilitando la traducción y adaptación tanto del programa como de las actividades creadas.

- Recoger las sugerencias de mejoras y ampliaciones que los usuarios habían ido enviando.
- Hacer posible que el programa pueda ir ampliándose a partir del trabajo cooperativo entre diversos equipos de programación.
- Crear un entorno de creación de actividades más potente, sencillo e intuitivo, adaptándolo a las características de los actuales entornos gráficos de usuario.

CARACTERÍSTICAS

COMPONENTES

JClic está formado por cuatro aplicaciones, dos de las cuales, se utilizan para la resolución de las actividades:

- **JClic Player**

Esta componente puede ser presentada de dos maneras diferentes: como applet o como aplicación JClic.

Applet: Un "applet" que permite incrustar actividades JClic en una página web para ejecutarlas en cualquier navegador. Se descarga, automáticamente, la primera vez que se visita alguna página que contenga un proyecto JClic incrustado.

Aplicación JClic: Un programa independiente que una vez instalado permite realizar las actividades desde el disco.

- **JClic autor**

La herramienta de autor que permite crear, editar y publicar las actividades de una manera sencilla, visual e intuitiva.

- **JClic reports**

Es el módulo encargado de recopilar los datos (tiempo empleado en cada actividad, intentos, aciertos, etc.), y presentarlos después en informes estadísticos de diversos tipos. JClic reports se basa en un esquema cliente - servidor. El servidor puede ser cualquier ordenador de una red, y los clientes son de dos tipos: las aplicaciones JClic (applet y player), que envían al servidor las puntuaciones obtenidas por los usuarios al realizar las actividades, y los navegadores web (Firefox, Opera, Explorer...), desde los que se pueden consultar los resultados y administrar la base de datos.

Como se mencionó, el primer módulo (applet) se descarga automáticamente la primera vez que se visita alguna página que contenga un proyecto JClic incrustado. Los otros tres, se pueden instalar en el ordenador mediante Java WebStart desde la página de descargas.

COMPATIBILIDAD Y NUEVAS POSIBILIDADES

El desarrollo de JClic se ha hecho intentando respetar al máximo la compatibilidad con el programa Clic 3.0, de manera que los paquetes de actividades existentes puedan ser automáticamente reconocidos por la nueva plataforma.

Éstas son algunas de las novedades del JClic con respecto a Clic 3.0:

- Uso de entornos gráficos de usuario ("skins") personalizables, que contienen los botones y el resto de elementos gráficos que enmarcan las actividades.
- Uso de gráficos BMP, GIF, JPG y PNG.
- Incorporación de recursos multimedia en formato WAV, MP3, AVI, MPEG, QuickTime y Flash 2.0, entre otros, así como de GIFs animados y con transparencia.
- Sonidos de eventos (hacer clic, relacionar, completar, acertar, fallar, etc.) configurables para cada actividad o proyecto.
- Generadores de formas ("shapers") que controlan el aspecto de las casillas de las actividades.
- Mejoras visuales: Posibilidad de escribir código HTML en las casillas, incrustación de fuentes "TrueType", texto con estilos, uso de gradientes y colores semitransparentes.
- Nuevas características de las actividades: tiempo máximo, número máximo de intentos, orden de resolución, actividades de memoria con dos bloques de contenido, etc.

ARQUITECTURA ABIERTA

JClic tiene una arquitectura abierta que permite ampliar o adaptar sus funcionalidades en diversos aspectos. Se pueden crear nuevos módulos Java que extiendan el funcionamiento del programa en:

- Nuevos tipos de actividades, extendiendo la clase abstracta "Activity".
- Sistemas de creación automática de contenidos, a partir de la clase abstracta "AutoContentProvider" (en estos momentos la única implementación de esta clase es el módulo "Arith").

- Entornos gráficos de usuario, extendiendo la clase "Skin" o suministrando nuevos esquemas XML a "BasicSkin".
- Motores primarios de ejecución de actividades, extendiendo "ActivityContainer".
- Generadores de recortes y formas, extendiendo la clase abstracta "Shaper".
- Sistemas de informes, extendiendo la clase "Reporter".

JClic es un proyecto de software libre que el Departamento de Educación de la Generalitat de Cataluña pone a disposición de la comunidad bajo los términos de la Licencia Pública General de GNU (GPL). Eso permite utilizarlo, distribuirlo y modificarlo libremente siempre que se respeten determinadas condiciones, entre las que cabe destacar el reconocimiento de autoría y la persistencia de la licencia GPL en cualquier obra derivada. El código fuente de JClic está disponible en la plataforma de desarrollo.

FORMATOS DE DATOS

Los datos de JClic se almacenan en formato XML. Eso permite su integración en bancos de recursos de estructura compleja, así como la reutilización de los proyectos JClic en otras aplicaciones.

JClic trabaja con dos tipos de archivos:

ARCHIVOS CON EXTENSIÓN .JCLIC

Son documentos XML que contienen la descripción completa de un proyecto JClic. La estructura de estos documentos está descrita en el esquema XML jclic.xsd.

El elemento raíz de los documentos jclic tiene el nombre <JClicProject> y contiene cuatro elementos principales:

- **<Settings>**
Información sobre los autores/se del proyecto, descriptores temáticos, revisiones, etc.
- **<Activities>**
Contiene elementos del tipo <activity> que definen el funcionamiento y las características propias de cada actividad.
- **<Sequence>**
Describe la orden en que se tienen que presentar las actividades y el comportamiento de los botones de avanzar y retroceder.

- **<MediaBag>**

Relación del nombre y la ubicación de todos los ingredientes necesarios para ejecutar las actividades: imágenes, sonidos, vídeo, MIDI, fuentes TTF, etc.

ARCHIVOS CON EXTENSIÓN .JCLIC.ZIP

Son archivos ZIP estándar que contienen un único documento .jclic y algunos o todos los ingredientes (imágenes, archivos multimedia...) necesarios para ejecutar las actividades. El uso de éste formato permite encapsular un proyecto en un único fichero.

Además de estos dos formatos, JClíc es capaz también de importar los ficheros PAC y PCC de Clíc 3.0.

INTERNACIONALIZACIÓN

Todos los textos y mensajes de JClíc se encuentran en ficheros externos, con el fin de simplificar su traducción a otros idiomas.

Tanto los textos del programa como los de las actividades, se encuentran en formato Unicode. Eso permite utilizarlo con alfabetos y sistemas de escritura no occidentales. El programa soporta también la representación y escritura bidireccional (derecha-izquierda o izquierda-derecha) cuándo el idioma utilizado así lo requiera. El proyecto de desarrollo está abierto a la participación de todos los que quieran elaborar traducciones del programa a otros idiomas.

TIPOS DE ACTIVIDADES

JClíc permite realizar siete tipos básicos de actividades:

Las **asociaciones** pretenden que el usuario descubra las relaciones existentes entre dos conjuntos de información.

Los **juegos de memoria** donde hay que ir descubriendo parejas de elementos iguales o relacionados entre ellos, que se encuentran escondidos.

Las actividades de **exploración**, **identificación** e **información**, que parten de un único conjunto de información.

Los **puzzles**, que plantean la reconstrucción de una información que se presenta inicialmente desordenada. Esta información puede ser gráfica, textual, sonora o combinar aspectos gráficos y auditivos al mismo tiempo.

Las actividades de **respuesta escrita** que se resuelven escribiendo un texto (una sola palabra o frases más o menos complejas).

Las **actividades de texto**, que plantean ejercicios basados siempre en las palabras, frases, letras y párrafos de un texto que hay que completar, entender, corregir u ordenar. Los textos pueden contener también imágenes y ventanas con contenido activo.

Las **sopas de letras** y los **crucigramas** son variantes interactivas de los conocidos pasatiempos de palabras escondidas.

Algunos de los tipos de actividades presentan diversas modalidades, dando lugar a 16 posibilidades diferentes que pueden verse en la Figura 9.1a y b

TIPOS		DESCRIPCIÓN
Asociación	simple	Se presentan dos conjuntos de información que tienen el mismo número de elementos. A cada elemento del conjunto origen corresponde un elemento del conjunto destino.
	compleja	También se presentan dos conjuntos de información, pero éstos pueden tener un número diferente de elementos y entre ellos se pueden dar diversos tipos de relación: uno a uno, uno a varios, elementos sin asignar ...
Juego de memoria		Este tipo de actividades consiste en descubrir parejas de elementos entre un conjunto de casillas inicialmente escondidas. Las parejas pueden estar formadas por dos piezas idénticas, o por dos elementos relacionados. En cada intento se destapan dos piezas, que se vuelven a esconder si no forman pareja. El objetivo es destapar todos los elementos del panel.
Actividad de exploración		Se muestra una información inicial y al hacer clic encima suyo se muestra, para cada elemento, una determinada pieza de información.
Actividad de identificación		Se presenta sólo un conjunto de información y hay que hacer clic encima de aquellos elementos que cumplan una determinada condición.
Pantalla de información		Se muestra un conjunto de información y, opcionalmente, se ofrece la posibilidad de activar el contenido multimedia que lleve cada elemento.
Puzzle	doble	Se muestran dos paneles. En uno está la información desordenada y el otro está vacío. Hay que reconstruir el objeto en el panel vacío llevando allí las piezas una por una.
	de intercambio	En un único panel se mezcla la información. En cada intento se conmutan las posiciones de dos piezas, hasta ordenar el objeto.
	de agujero	En un único panel se hace desaparecer una pieza y se mezclan las restantes. En cada intento se puede desplazar una de las piezas hacia el agujero, hasta que queden todas en el orden original.

Figura 9.1a - Actividades JClic.

TIPOS		DESCRIPCIÓN
Texto	Completar texto	En un texto se hacen desaparecer determinadas partes (letras, palabras, signos de puntuación, frases) y el usuario debe completarlo.
	Rellenar agujeros	En un texto se seleccionan determinadas palabras, letras y frases que se esconden o se camuflan, y el usuario debe completarlo. La resolución de cada uno de los elementos escondidos se puede plantear de maneras distintas: escribiendo en un espacio vacío, corrigiendo una expresión que contiene errores o seleccionando diversas respuestas posibles de una lista.
	Identificar elementos	El usuario ha de señalar con un clic de ratón determinadas palabras, letras, cifras, símbolos o signos de puntuación.
	Ordenar elementos	En el momento de diseñar la actividad se seleccionan en el texto algunas palabras o párrafos, que se mezclarán entre sí. El usuario ha de volver a ponerlo en orden.
Respuesta escrita		Se muestra un conjunto de información y, para cada uno de sus elementos, hay que escribir el texto correspondiente.
Palabras cruzadas		Hay que ir rellenando el panel de palabras a partir de sus definiciones. Las definiciones pueden ser textuales, gráficas o sonoras. El programa muestra automáticamente las definiciones de las dos palabras que se cruzan en la posición donde se encuentre el cursor en cada momento.
Sopa de letras		Hay que encontrar las palabras escondidas en un panel de letras. Las casillas neutras del panel (que no pertenecen a ninguna palabra) se rellenan con caracteres seleccionados al azar en cada jugada. Puede tener un contenido asociado. En este caso se irá desvelando un elemento de un conjunto de información (texto, sonidos, imágenes o animaciones) cada vez que se localice una palabra nueva.

Figura 9.1b - Actividades JClíc.

OBTENER JCLIC

JClíc es una aplicación Java que se distribuye mediante la tecnología WebStart. La primera vez que se hace clic en los enlaces de la página de JClíc se descargarán los programas, que quedarán instalados en el ordenador. A partir de ahí se podrá poner en marcha las aplicaciones tantas veces como se desee, ya sea desde la ventana de control de Java WebStart o mediante los iconos, que se crean en el escritorio y/o en el menú inicio. La conexión a Internet sólo es necesaria la primera vez.

Una de las ventajas del sistema WebStart es que los programas se instalan y se actualizan cuando hay alguna mejora disponible, sin que haya que descargarlos manualmente. A pesar de todo, si se prefiere no utilizar este sistema o si se necesita hacer una instalación manual (por ejemplo, en ordenadores sin conexión a Internet), existe también la posibilidad de

descargar un instalador EXE (para sistemas Windows) o un ZIP en el que se encuentran los archivos y las instrucciones de instalación para Linux, Mac y Solaris.

ESTRUCTURA DE CLASES

JClic es un programa formado por clases Java organizadas en paquetes de acuerdo a su funcionalidad. En la imagen continuación se muestra la composición de paquetes de JClic con sólo un nivel de profundidad.

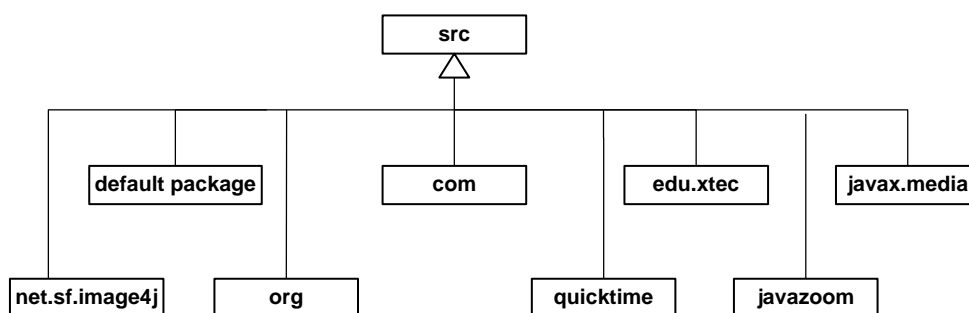


Figura 9.2 Estructura interna del paquete "src".

A continuación se dará una breve explicación del contenido de los paquetes más utilizados para llevar a cabo la integración que se presentará en el capítulo 9, comenzando por el paquete "default package".

El paquete "**default package**", no es en realidad un paquete creado por el autor de la aplicación, son clases dentro "src" que no están empaquetadas. Estas clases son:

- JClicApplet
- JClicAuthor
- JClicAuthorApplet
- JClicInstaller
- JClicPlayer

De este conjunto de clases se destacan para el objetivo de esta tesina JClicAuthor y JClicPlayer.

- **JClicAuthor** es la clase que instancia a través del método main() a la ventana mediante la cual se crean los proyectos y actividades.
- **JClicPlayer** es la clase que instancia a través del método main() la ventana mediante la cual se pueden resolver las actividades.

En el paquete “**edu.xtec**” se encuentran las actividades, contenedores, reportes, y demás. A continuación una imagen ilustrativa de la estructura interna de paquetes.

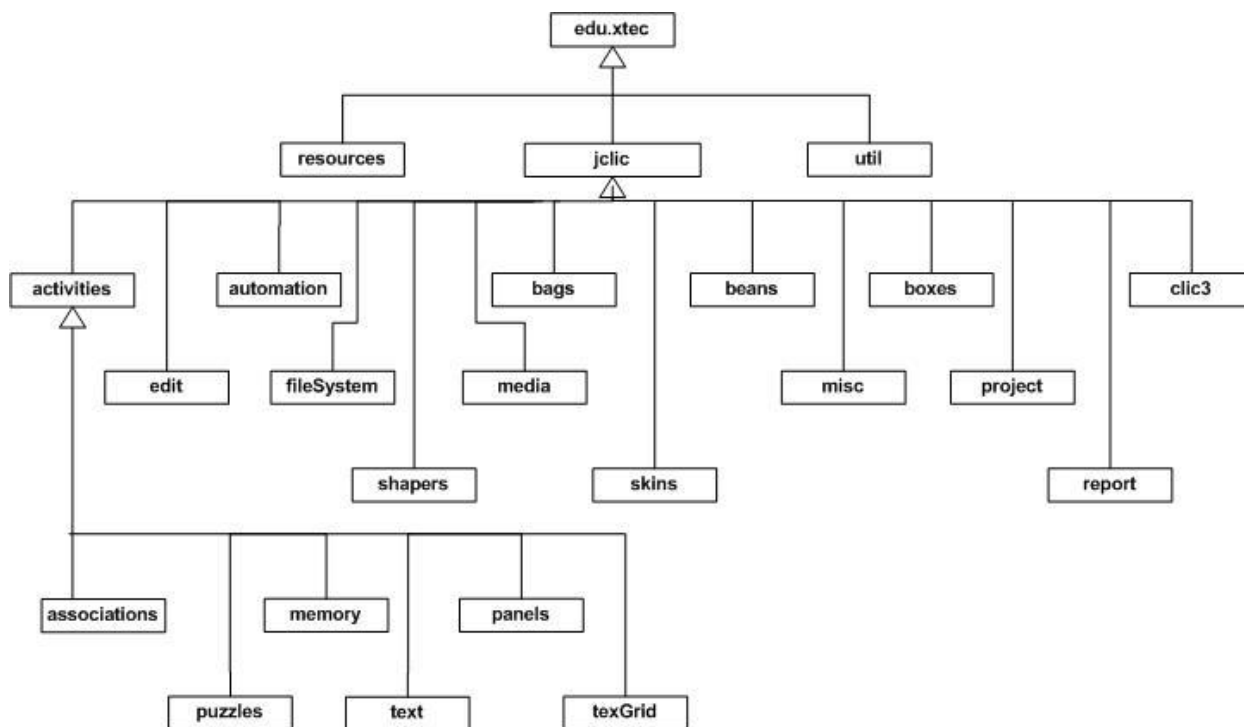


Figura 9.3 Estructura interna del paquete “**edu.xtec**”.

El paquete “**jclic**” contiene clases relacionadas a las actividades y a la resolución de las mismas, entre ellas se encuentran las siguientes:

- **Activity**: es la clase base de las actividades de JClíc. Contiene dentro una clase llamada “**Panel**”, la cual es responsable de la interacción del usuario con el contenido de la actividad.
- **Player**: es una de las principales clases de JClíc. Para poder ejecutar las actividades, esta clase provee a las actividades de todos los recursos necesarios: imágenes y contenidos media, interfaz de usuario, etc.

El paquete “**activities**” contiene las clases que representan los tipos de actividades que se pueden crear con JClíc.

En el paquete “**bags**” se encuentran clases como “**ActivityBag**” que se utiliza para almacenar todas las actividades incluidas en un proyecto JClíc, almacenadas en un vector de objetos de tipo “**ActivityBagElement**”, clase que también está contenida en el paquete “**bags**”.

El paquete “**boxes**” contiene clases usadas para el manejo de las actividades y su contenido. Algunas de estas clases son:

- **AbstractBox:** es la clase base para la mayoría de los componentes gráficos de JClíc. Describe un área con determinadas propiedades que indican cómo se debe dibujar el componente en la pantalla.
- **BoxBase:** esta clase contiene todos los atributos visuales que se necesitan para dibujar objetos de tipo “AbstractBox”: colores de fondo, tipo de letra, bordes, etc.
- **ActiveBoxContent:** esta clase define un contenido que puede ser mostrado por un objeto “ActiveBox”. Este contenido puede ser un texto, una imagen, un fragmento de una imagen o una combinación de texto e imagen. El estilo se puede definir en un objeto “BoxBase”. También almacena información sobre el tamaño y posición del “ActiveBox”.
- **ActiveBox:** es una clase muy usada en las actividades de JClíc, las celdas en los rompecabezas y asociaciones, mensajes y otros objetos son “active boxes”. El contenido, tamaño y posición del “ActiveBox” está determinado por sus miembros de tipo “ActiveBoxContent”.
- **ActiveBagContent:** esta clase almacena una colección de objetos de tipo “ActiveBoxContent”, y provee métodos para manejarlos.
- **ActiveBoxBag:** esta clase contiene objetos “ActiveBox” e implementa métodos específicos para manejar objetos de tipo “ActiveBagContent” y “ActiveBox”.
- **BoxConector:** permite al usuario conectar visualmente dos objetos de tipo “ActiveBox” en un panel de actividad.

Con esta revisión y análisis del software Jclíc, se establecen las bases como para encarar la integración planteada en este trabajo, de manera de ofrecer al lector un marco conceptual para los capítulos siguientes.

CONCLUSIÓN

En el presente capítulo se ha estudiado el software elegido para el desarrollo del prototipo, introduciendo sus conceptos más importantes, los cuáles servirán de base al capítulo 9 en el cual, como ya se ha dicho, se hablará del prototipo de adaptación.

En el próximo capítulo se presentará la herramienta de reconocimiento de voz elegida para realizar la integración con JClíc.

CAPÍTULO 9

MODIFICACIÓN DEL SOFTWARE
EDUCATIVO JCLIC PARA LA
INCORPORACIÓN DE COMANDOS POR
VOZ

INTRODUCCIÓN

En los primeros capítulos de este trabajo se estudiaron varios conceptos, algunos softwares y fundamentos teóricos que forman una base para introducir el presente capítulo.

En los capítulos 7 y 8, se explicaron las características y la arquitectura de Sphinx-4 y de JClic, respectivamente. En el presente capítulo se explicará cada uno de los pasos que fueron llevados a cabo, para implementar el prototipo de aplicación que surge de la integración del reconocedor de voz con la aplicación JClic.

En particular se detallarán todas las posibles alternativas analizadas para la configuración del reconocedor de voz, respecto al conjunto de palabras a ser reconocidas y otras adaptaciones. Luego se explicarán las modificaciones realizadas al programa educativo antes mencionado, para incorporar el reconocedor a la resolución de las actividades.

En primer lugar, se hará un resumen de los pasos previos que fue necesario realizar y los temas investigados que constituyen la base para las decisiones tomadas en el desarrollo del prototipo.

Luego, se explicará el funcionamiento del prototipo desarrollado, incluyendo la resolución de una actividad de asociación simple a modo de ejemplo.

Posteriormente, se plantearán posibles formas de adaptar el resto de actividades disponibles en JClic, de manera de poder resolverlas utilizando reconocimiento de voz. Estas posibles reformas serán propuestas en el presente desarrollo y quedará pendiente su implementación como líneas de trabajo futuro.

En otra sección, se listarán algunos pasos a tener en cuenta si se desea integrar la funcionalidad de reconocimiento de voz a cualquier software sea o no con fines educativos.

Por último, se describirá una evaluación realizada a expertos en el tema de tecnología aplicada a la educación especial y reconocimiento de voz, además de las conclusiones de la encuesta.

PROPUESTA DE PROTOTIPO

La propuesta de prototipo de adaptación que se presenta en este trabajo incluye varias etapas de análisis, las cuales se plantean en las secciones siguientes.

De manera general, lo que se propone es realizar las modificaciones necesarias en el programa JClic de modo que se puedan utilizar comandos por voz para la resolución de actividades, tecnología que es provista por el framework Sphinx-4.

Cabe aclarar, que estas adaptaciones se realizarán sobre la herramienta JClicPlayer, que fue explicado en el capítulo 8.

En primera instancia, el prototipo solamente abarca la adaptación de un tipo de actividad que JClic provee, que es la Asociación Simple.

En la sección siguiente se describirá conceptualmente, cómo se realiza una actividad de este tipo.

En las próximas dos secciones se detallará el análisis abordado sobre cada componente en particular.

TRABAJOS REALIZADOS PREVIOS AL PROTOTIPO

Para llegar a la realización del prototipo, se tuvieron que estudiar, investigar y analizar diferentes temas, algunos de los cuales se detallan a continuación.

En primer lugar se analizaron varias herramientas de autor y programas educativos con el fin de tomar conocimiento de qué está disponible para utilizar en el ámbito educativo y revisar aspectos funcionales. Cabe aclarar que, por supuesto, esta lista de herramientas estudiadas no es exhaustiva. Esta tarea se mostró en el capítulo 2 y un estudio más detallado de la herramienta elegida para la integración se realizó en el capítulo 8. Previo a comenzar con las modificaciones del código para la integración, fue necesario estudiar la arquitectura de diseño y desarrollo de la herramienta elegida, JClic, interiorizándose en sus componentes, clases y cuestiones específicas del lenguaje JAVA.

Luego se estudiaron conceptos bases sobre reconocimiento de voz para entender la lógica y el funcionamiento de diversas herramientas de RV, lo cual es clave para su utilización. Esto incluye conceptos como: modelos acústicos, modelos de lenguaje, diccionarios, fonemas, etc. Además se estudiaron diferentes motores de reconocimiento de voz, analizando sus características y tomando decisiones sobre las conveniencias para la adaptación propuesta. Este análisis se describió en el capítulo 6. Se eligió un motor de RV, se analizó en detalle. En particular, se abordó el estudio del motor de reconocimiento de voz Sphinx, versión 4.

Posteriormente a la elección de las herramientas a utilizar, se propuso una estrategia de adaptación para JClic, que permitiera integrar los comandos por voz en la resolución de actividades. Para esto, se comenzó con el análisis y desarrollo de la integración, estudiando temas específicos de implementación, como el desarrollo con threads dentro de JClic y el reconocedor. Luego, se tomaron decisiones de configuración para el RV, tales como: revisar diferentes fonéticas, armar un diccionario, un modelo acústico, etc.

Por último, se decidió consultar a expertos sobre el ámbito particular de alumnos con discapacidad motora y realizar un informe al respecto.

Todos los pasos correspondientes a la integración y consulta a expertos se irán describiendo en detalle a lo largo de este capítulo.

PROPUESTA PLANTEADA PARA EL PROTOTIPO – EL CASO DE LA ASOCIACIÓN SIMPLE

En este tipo particular de actividad que JClic permite crear, permite al usuario descubrir las relaciones existentes entre dos conjuntos de información. Es decir, se presentan dos grupos de datos que tienen el mismo número de elementos, donde a cada elemento del origen le corresponde un elemento del destino. Es por ello que se la denomina simple, a diferencia de la asociación compleja, donde a cada elemento del origen puede corresponderle 0, 1, o más elementos del destino. En la Figura 9.1 se puede ver un ejemplo de asociación simple.

$? + 10 = 53$	$? + 1 = 81$	$70 \times ? = 280$	$64 - 8 =$
$46 + ? = 48$	$79 ? 4 = 83$	$80 : 1 =$	$45 - 8 =$
37	43	56	80
4	2	+	80

Figura 9.1 – Actividad de Asociación Simple.

En principio, lo que se propone es modificar las actividades de JClic de manera que se puedan realizar a través de comandos por voz.

Para ello se tomó como inicio este tipo particular de actividad. Como primera medida para llevar a cabo esta integración se debieron tomar algunas decisiones, las cuales se detallan a continuación.

Una de las primeras cuestiones que se plantearon fue decidir qué mecanismo proveer para identificar cada elemento de la actividad mostrado en pantalla. Para esto se agregó en cada

elemento a asociar, un número identificatorio de manera que el usuario pueda nombrarlo. A este número se lo denominará etiqueta en las siguientes secciones.

El segundo tema a resolver fue el de conocer cuándo el usuario termina de nombrar los dos elementos a unir. Para ello se pensó en que se deba decir, por ejemplo, “Uno con Tres Aceptar”; lo que se interpreta de esta sentencia es lo siguiente:

El primer número representa un casillero del primer conjunto de información, la palabra “con” indica que se va a nombrar el casillero del segundo conjunto, representado por el segundo número de la frase. La palabra “aceptar” indica que el usuario quiere realizar la unión de los casilleros nombrados.

Además, se agregó el código necesario para que el programa pida confirmación sobre la información recolectada mostrando un cartel en pantalla.

Para poder lograr la adaptación planteada, se agregó el código necesario de manera que el motor de reconocimiento por voz esté funcionando junto con JClic cuando el usuario va a realizar la actividad.

También, se tuvo que decidir que el programa pida al usuario indicación sobre si quiere realizar la actividad con reconocimiento por voz o no, en el momento en que se carga un proyecto.

En esta sección se ha querido plantear a grandes rasgos la estrategia propuesta. En las siguientes secciones se mostrará la justificación de por qué se eligió esta estrategia de solución y qué tareas fueron necesarias realizar para llevarla a cabo.

ALGUNAS DECISIONES PREVIAS Y CONSIDERACIONES DE JCLIC

Al momento de iniciar este segmento del trabajo fueron surgiendo diferentes cuestiones que presentaban varias alternativas, de las cuales, hubo que seleccionar la más indicada(a nuestro criterio) para llevar a cabo el objetivo expresado.

Dado que la actividad a modificar, en primera instancia y en el marco del prototipo, es la de asociación simple, las cuestiones planteadas en esta sección aplican sólo a este tipo de actividad. En secciones siguientes se planteará cómo resolver la adaptación del resto de los tipos de consignas disponibles en JClic y las cuestiones que ellas involucran.

Una de las decisiones que se consideró fue, cómo tomar conocimiento de que se desea realizar la actividad utilizando comandos por voz.

Para ello el usuario deberá contar con la asistencia del docente, debido a que es éste quien tomará la decisión para cada alumno en particular. El programa, por su parte, mostrará un

mensaje en pantalla al momento de comenzar la actividad. Este pedirá que se indique si se desea utilizar reconocimiento de voz.

Con respecto a las etiquetas, cabe mencionar, que se generan cuando se ejecuta JClicPlayer, sólo si se indica que se desea realizar la actividad con comandos por voz. Se debió además, agregar el código necesario para que se inserte la etiqueta en el componente que representa al casillero con la información. Esto es así, debido a que cuando el docente crea la asociación simple y la guarda, se almacena manteniendo el orden con el que fue generada, y luego JClicPlayer las desordena antes de presentarla al alumno. Si se hubiera decidido asignar una etiqueta a cada celda en el momento de su creación se mantendría la correspondencia entre celdas, aún cuando estén desordenadas, ya que conservarían la misma etiqueta y la actividad se mostraría resuelta. Dada esta situación, se consideró como alternativa apropiada, que cuando JClicPlayer carga la actividad a ser resuelta por el alumno, desordene aleatoriamente los casilleros, momento en el cual es posible etiquetarlos sin mostrar las respuesta correcta de antemano. Las decisiones correspondientes al tipo de etiqueta a utilizar, se verán en la próxima sección.

Por último, se agregó el código necesario de manera que la aplicación muestre un mensaje pidiendo confirmación de lo dicho. Así, cuando el usuario nombra los casilleros que desea unir, el programa presentará un mensaje mostrando las palabras reconocidas. Para dar confirmación positiva al mensaje se deberá decir “Aceptar” y en caso contrario “Cancelar”.

De la misma manera, se incorporó la posibilidad de navegar entre las diferentes actividades, implementando los comandos por voz “siguiente” y “anterior”, los cuales dirigen la secuencia a la actividad posterior y anterior, respectivamente.

ELECCION DE LAS ETIQUETAS PARA IDENTIFICAR CASILLEROS

Uno de los primeros pasos para la inclusión del reconocedor, es definir el conjunto de palabras a ser utilizadas como etiquetas en cada una de las celdas, para permitir al alumno escoger una de ellas.

Una de las cuestiones a resolver fue la forma de identificar un casillero determinado dentro del conjunto de información. Es necesario contar con algún mecanismo de identificación para que sea posible referirse unívocamente a cada uno de los casilleros utilizando reconocimiento de voz. De esta manera, se pensó en numerar los casilleros con dígitos, los cuales resultarían más que apropiados desde el punto de vista del tamaño de los conjuntos.

Cabe aclarar, que se han revisado una cantidad de actividades de asociación simple diseñadas en JClic y se ha podido comprobar que en promedio no superan los 10 casilleros por conjunto, con lo cual se decidió realizar la identificación de casilleros con dígitos del 0 al 9, para esta primera etapa del prototipo. Sin embargo, JClic en la creación de la actividad permite un número mayor de casilleros.

En primer instancia, se pensó en utilizar las letras del alfabeto para definir el diccionario, pero al momento de llevarlo a la práctica, se encontró la dificultad de que ciertas letras, tales como la “b” y la “d”, eran muy similares en su pronunciación, por lo que la tasa de aciertos del reconocedor disminuía considerablemente. Estas dificultades de pronunciación han sido estudiadas teóricamente e informadas en el capítulo 4.

Por otro lado, si se ampliaba el número de casilleros a utilizar, resultaba más natural usar combinaciones de dígitos (por ejemplo 10) que utilizar letras (por ejemplo ab). También, se debían utilizar letras alternadas quitando del diccionario aquellas que causaban conflictos como los ya mencionados o aquellas que resultaban muy complejas en cuanto a su pronunciación (por ejemplo, el caso de la letra ‘r’).

Considerado esto, se evaluó la posibilidad de utilizar números para la creación de las etiquetas. Esta solución presenta ciertas ventajas, respecto a la planteada anteriormente. Además, se realizaron las adaptaciones necesarias para evitar la utilización de letras de difícil pronunciación. Para esto, se tuvieron en cuenta otras palabras alternativas a la correcta, por ejemplo, se admite que el usuario diga “tes” en lugar de “tres”, “tínco” en lugar de “cinco”, “acetar” en lugar de “aceptar”, entre otras. Esta información acerca de las dificultades de pronunciación también proviene del estudio plasmado en el capítulo 4.

Si bien esta decisión implica un diccionario de mayor tamaño, presenta consecuencias positivas en cuanto al aumento de usuarios que podrían utilizar el prototipo. Así, se intenta lograr un equilibrio entre performance de la aplicación y usabilidad del producto.

También se podría utilizar como etiqueta el nombre del elemento dentro del casillero. Por ejemplo, si en el casillero se encuentra la imagen de un animal, la etiqueta podría ser el nombre del animal. Si bien esta opción, podría tener un impacto positivo desde el punto de vista educativo, complejiza la implementación ya que se debería disponer de una herramienta que permita la creación dinámica del diccionario y la gramática.

Más allá de que, en el marco del prototipo, se decidió utilizar dígitos, se propone, como línea futura de trabajo, que las etiquetas a utilizar sean configurables desde la aplicación de autor. Así se podrán utilizar números, letras o palabras, dependiendo de la configuración del docente.

CONFIGURACIÓN Y PUESTA A PUNTO DEL RECONOCEDOR SPHINX-4

De igual manera que se plantearon las consideraciones para JClic, en esta sección se analizarán las surgidas para Sphinx-4, incluyendo su descarga, los pasos que se fueron analizando y las decisiones tomadas.

En primer lugar, para utilizar Sphinx, se debe descargar la aplicación desde el sitio oficial⁴¹. Allí está disponible el código fuente de la herramienta, aunque si no se desea modificar código (como en nuestro caso), alcanza con incluir el archivo .jar en la aplicación donde se va a integrar.

Como primer paso en la configuración, resulta necesario decidir el modelo acústico a utilizar para que el reconocedor sepa cómo se pronuncia cada fonema y así poder reconocerlo.

Actualmente, Sphinx-4 dispone de modelos que han sido creados utilizando SphinxTrain y pueden descargarse desde el sitio de cmusphinx.org. Esta herramienta genera modelos acústicos en el formato utilizado por Sphinx-3. Más adelante, en esta misma sección, se explicará cómo convertir esos archivos para ser incorporados a la versión utilizada en este trabajo.

Dos de los modelos más utilizados por Sphinx-4 son TDIGITS y Wall Street Journal (WSJ), que ya están incluidos en el directorio “lib” de la distribución binaria (**Sphinx, modelos acústicos – 2010**).

Cada modelo acústico implementa la interface de AcousticModel. Por ejemplo, el modelo WSJ está envuelto en una clase llamada `edu.cmu.sphinx.model.acoustic.WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz`, la cual implementa la interface de AcousticModel. La implementación de esta clase está en el archivo .jar del modelo. Para usar un modelo acústico en particular, en principio se necesitan dos pasos:

1. Incluir el archivo .jar en el classpath.
2. Especificar la clase que implementa el modelo en el archivo de configuración.

En primer lugar, se pensó como una alternativa válida crear el diccionario utilizando el modelo `WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz` que viene incluido con la distribución de Sphinx-4 y, si bien está entrenado para el idioma en inglés, reemplazando los fonemas puede reconocer español. Esta afirmación está basada en el desarrollo de Magnus⁴² y pruebas propias desarrolladas.

WSJ posee un diccionario en inglés de 129.247 palabras, utilizando 39 fonemas que se muestran a continuación.

⁴¹ CMU Sphinx, Sitio Oficial - <http://cmusphinx.sourceforge.net/sphinx4/>

⁴² Alexandre Trilla Castelló - <https://forja.rediris.es/projects/csl2-magnus/>

Phoneme Example Translation

Phoneme	Example	Translation
AA	odd	AA D
AE	at	AE T
AH	hut	HH AH T
AO	ought	AO T
AW	cow	K AW
AY	hide	HH AY D
B	be	B IY
CH	cheese	CH IY Z
D	dee	D IY
DH	thee	DH IY
EH	Ed	EH D
ER	hurt	HH ER T
EY	ate	EY T
F	fee	F IY
G	green	G R IY N
HH	he	HH IY
IH	it	IH T
IY	eat	IY T
JH	gee	JH IY
K	key	K IY
L	lee	L IY
M	me	M IY
N	knee	N IY
NG	ping	P IH NG
OW	oat	OW T
OY	toy	T OY
P	pee	P IY
R	read	R IY D
S	sea	S IY
SH	she	SH IY
T	tea	T IY

TH	theta	TH EY T AH
UH	hood	HH UH D
UW	two	T UW
V	vee	V IY
W	we	W IY
Y	yield	Y IY L D
Z	zee	Z IY
ZH	seizure	S IY ZH ER

Si bien los fonemas pertenecen al idioma inglés, en un primer momento, fueron utilizados para generar el diccionario para la integración con JClíc.

A continuación se mostrará parte del diccionario generado con los fonemas mostrados:

1	<sil>	SIL
2	CERO	S EH R OW
3	UNO	UH N AO
4	UNO(2)	UW N OW
5	DOS	D AO S
6	DOS(2)	D OW S
7	TRES	T R EH S
8	TRES(2)	T EH S
9	CUATRO	K UH AA T R AO
10	CUATRO(2)	K UH AA T OW
11	CINCO	S IY N K AO
12	CINCO(2)	T IY N K AO
13	SEIS	S EH IH S
14	SEIS(2)	S EH IY S
15	SIETE	S IH EH T EH
16	SIETE(2)	S IY EH T EH
17	OCHO	AO CH AO
18	OCHO(2)	AO CH OW
19	NUEVE	N UH EH V EH
20	NUEVE(2)	N UH EH B EH
21	ACEPTAR	AH S EH P T AH ER
22	CON	K OW N
23	CANCELAR	K AH N S EH L AH ER

Figura 9.2 – Diccionario creado con el modelo WSJ.

Esta solución fue parcialmente válida, ya que el reconocedor funcionaba con un alto porcentaje de acierto. Pero, a pesar de esto, se encontraron dos falencias. Por un lado, había errores en la precisión del reconocedor en ambientes ruidosos. Esto sería un problema en los casos en que la adaptación fuera utilizada en escuelas donde las aulas se comparten entre varios alumnos. Por otro lado, si se desea extender el diccionario y utilizar palabras con la letra “ñ”, no existe un fonema en el idioma inglés que lo represente.

A partir de estas conclusiones, se decidió utilizar un modelo basado en el idioma español. Luego de investigar sobre el tema, surgieron dos alternativas viables. Por un lado, se podía entrenar el reconocedor, utilizando la herramienta SphinxTrain, y por otro, utilizar modelos entrenados y testeados.

En el presente desarrollo se optó por utilizar un modelo ya entrenado, pero se hará además un resumen de los pasos a seguir si se quiere entrenar un modelo acústico para su futura utilización. De hecho, se realizaron pruebas con el entrenador, de manera tal, de entender y estudiar su funcionamiento.

UTILIZAR MODELOS YA ENTRENADOS

Como ya se dijo, en el presente trabajo se optó por trabajar con un modelo ya entrenado, disponible en la web para su libre utilización. El proyecto se llama Diálogos Inteligentes Multimodales en Español (DIME), dentro del cual hay más de un modelo acústico (varían en la amplitud del diccionario). El modelo elegido para este trabajo recibe el nombre de DIMEx30-T22⁴³, el cual fue desarrollado por investigadores del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) perteneciente a la Universidad Nacional Autónoma de México. Estos archivos fueron creados con el software SphinxTrain y el corpus DIMEx100. Dicho Corpus tiene por objetivo hacer posible la construcción de modelos acústicos y diccionarios de pronunciación para la creación de sistemas computacionales para el reconocimiento del español hablado en México. Este tipo de sistemas permiten transcribir una señal de voz en su representación textual y la creación de aplicaciones de reconocimiento de habla de propósito general, con un vocabulario amplio y locutores diversos. El corpus es también útil para la investigación en reconocimiento de locutores, y para la creación de voces para sistemas de síntesis de voz.

El modelo elegido consta de 22 unidades alofónicas que representan el alfabeto español completo, es por esto que a su nombre se le agrega T22. Las últimas dos unidades fonéticas representan el silencio y el ruido, y no son transcripciones del habla (ver figura9.3).

⁴³Información del proyecto disponible en <http://leibniz.iimas.unam.mx/~luis/DIME/recursos.html>

1	a
2	b
3	d
4	e
5	f
6	g
7	i
8	k
9	l
10	m
11	n
12	n~
13	o
14	p
15	r
16	r1
17	s
18	t
19	tS
20	u
21	x
22	Z
23	+BN+
24	SIL

Figura 9.3 – Listado de fonemas de DIMEx30-T22.

A partir de esta lista de unidades fonéticas se creó el diccionario a utilizar en la integración con JClíc. Podría haberse utilizado el diccionario, tal cual lo presenta DIMEx30, pero había palabras que no se encontraban en él, por lo que se optó por redefinirlo, respetando las unidades fonéticas presentadas. Respecto del modelo de lenguaje, la definición del modelo acústico y su arquitectura, se respetó el proporcionado por DIMEx30. En la figura 9.4 se muestra parte del diccionario modificado.

3	ACEPTAR	a s e p t a r1
4	ACEPTAR(2)	a s e t a r1
5	ANTERIOR	a n t e r1 i o r1
6	CANCELAR	k a n s e l a r1
7	CERO	s e r1 o
8	CERO(2)	t e r1 o
9	CINCO	s i n k o
10	CINCO(2)	t i n k o
11	CON	k o
12	CON(2)	k o n
13	CUATRO	k u a t r1 o
14	CUATRO(2)	k u a t o
15	DOS	d o s
16	NUEVE	n u e b e
17	OCHO	o tS o
18	SEIS	s e i s
19	SIETE	s i e t e
20	SIGUIENTE	s i e n t e
21	SIGUIENTE(2)	s i g i e n t e
22	TRES	t r1 e s
23	TRES(2)	t e s
24	UNO	u n o

Figura 9.4 - Diccionario de palabras del prototipo.

Para incorporar estos archivos a la aplicación Jclic, se debió crear en primer lugar un archivo .jar que, por convención, debía respetar la estructura de directorios de los modelos provistos por Sphinx-4. La estructura utilizada es la siguiente:

```
- etc
    |-- variables.def
    |-- dimex30.1000.mdef
    |-- dimex30.lm.DMP
    |-- vocabulario.gram
- dict
    |-- cmudict.0.6d
    |-- fillerdict
- cd_continuous_8gau
    |-- means
    |-- mixture_weights
    |-- transition_matrices
    |-- variances
```

Esta estructura se debió incluir dentro de una carpeta edu/cmu/sphinx/model/acoustic/dimex30/ para generar el archivo .jar.

Como se muestra en la estructura, dentro de la carpeta dimex30 se encuentra un archivo llamado model.props que ha sido creado manualmente con la siguiente información:

```
description = DIMEx30 proyecto de iimas
modelClass = edu.cmu.sphinx.model.acoustic.dimex30.Model
modelLoader = edu.cmu.sphinx.model.acoustic.dimex30.ModelLoader
isBinary = true
featureType = 1s_c_d_dd
vectorLength = 39
sparseForm = false
numberFftPoints = 512
numberFilters = 40
gaussians = 8
minimumFrequency = 130
maximumFrequency = 6800
sampleRate = 16000
dataLocation = cd_continuous_8gau
modelDefinition = etc/dimex30.1000.ci.mdef
```

El archivo variable.def que se muestra dentro del directorio “etc” es un archivo de configuración que, al igual que el anterior, se creó manualmente. El contenido del archivo es el siguiente:

```
set exptname = JclicTrain
set vector_length = 13
set dictionary = $base_dir/lists/JclicTrain.dic
set fillerdict = $base_dir/lists/JclicTrain.filler
```

```
set statesperhmm = 3
set skipstate = no
set gaussiansperstate = 8
set feature = 1s_c_d_dd
set n_tied_states = 4000
set agc = none
set cmn = current
set varnorm = no
```

En la sección de “Entrenar el modelo con SphinxTrain” se verán algunas cuestiones a tener en cuenta en la definición de las variables contenidas en estos últimos dos archivos.

Los archivos Model y ModelLoader que se muestran en la estructura, se tomaron de los modelos que vienen con la distribución binaria de Sphinx-4.

El archivo .gram que está incluido en la carpeta “etc”, no fue proporcionado por DIMEx30, sino que se generó a partir de las palabras que debe aceptar la aplicación. Este archivo se define utilizando un estilo de gramática similar a BNF y se le asigna un nombre. En la figura 9.5 se muestra el archivo que fue generado para el presente trabajo.

```
grammar Jclic;

public <numbers> = ( CERO | UNO | DOS | TRES | CUATRO | CINCO | SEIS | SIETE |
                   OCHO | NUEVE | CON | ACEPTAR | CANCELAR | SIGUIENTE | ANTERIOR )+;
```

Figura 9.5 - Gramática del prototipo.

Luego de armado el archivo .jar, se incluyó en el classpath de la aplicación siguiendo los pasos mencionados anteriormente.

También, se debió configurar Sphinx-4 para incorporar los nuevos archivos del modelo acústico, el diccionario, la gramática y el modelo de lenguaje. Esto se realizó a través del archivo de configuración (configurationManager). En la siguiente figura (9.6) se muestra lo que debió modificarse para incorporar DIMEx30.


```

<component name="trigramModel" type="edu.cmu.sphinx.linguist.language.ngram.large.LargeTrigramModel">
  <property name="unigramWeight" value="0.7"/>
  <property name="maxDepth" value="3"/>
  <property name="logMath" value="logMath"/>
  <property name="dictionary" value="dictionary"/>
  <property name="location" value="resource:/edu.cmu.sphinx.model.acoustic.dimex30.Model!/edu/cmu/sphinx/model/
acoustic/dimex30/etc/dimex30.lm.DMP"/>
</component>

<component name="dictionary" type="edu.cmu.sphinx.linguist.dictionary.FastDictionary">
  <property name="dictionaryPath" value="resource:/edu.cmu.sphinx.model.acoustic.dimex30.Model!/edu/cmu/sphinx/model/
acoustic/dimex30/dict/cmudict.0.6d"/>
  <property name="fillerPath" value="resource:/edu.cmu.sphinx.model.acoustic.dimex30.Model!/edu/cmu/sphinx/model/
acoustic/dimex30/dict/fillerdict"/>
  <property name="addSilEndingPronunciation" value="false"/>
  <property name="allowMissingWords" value="false"/>
  <property name="unitManager" value="unitManager"/>
</component>

```

Figura 9.6 - Archivo de configuración.

Tanto el archivo de gramática como el diccionario, son utilizados por el archivo de configuración, llamado “config.xml” que fue descrito en el capítulo 7.

Con estas modificaciones ya se está en condiciones de utilizar el reconocedor, sólo falta ver cómo invocar a los métodos del reconocedor. Esta información se presentará más adelante en otra sub sección.

ENTRENAR EL MODELO CON SPHINXTRAIN

Con el objetivo de entender la funcionalidad y la lógica del entrenamiento de Sphinx, se realizaron pruebas utilizando la herramienta de SphinxTrain. Estas experiencias fueron llevadas a cabo sobre una plataforma Linux (Ubuntu) (**Entrenamiento usando SphinxTrain – 2010**).

A continuación se explica detalladamente el procedimiento para generar modelos acústicos. Si bien algunos pasos resultan muy sencillos de realizar, se dejan explicados de manera de dar al lector el marco completo acerca del proceso de entrenamiento.

Primero se debe instalar SphinxTrain. Para esto se copia el contenido de la carpeta SphinxTrain descargada desde <http://cmusphinx.sourceforge.net/wordpress/download/> y se debe copiar en la carpeta `/usr/local/src/`.

En segundo lugar, se debe instalar el entrenador. Por ello, se ubica dentro de la carpeta SphinxTrain y se ejecutan los dos comandos siguientes:

```

sudo ./configure
sudo make

```

Se debe generar una estructura de directorios para que las herramientas que ofrece SphinxTrain puedan ejecutarse. Se crea entonces un directorio raíz desde una terminal, y allí la estructura, cuya ruta es:

```
[root@localhost root]# cd /usr/local/src/
```

```
[root@localhost src]# sudo mkdir JclitTrain
```

En este caso el directorio raíz es JclitTrain.

Luego se debe crear una variable de ambiente, la cual contiene la ruta donde se encuentra SphinxTrain y posteriormente ejecutar el comando que crea la estructura automáticamente:

```
[root@localhost JclitTrain]# export SPHINXTRAINDIR=/usr/local/src/SphinxTrain
```

```
[root@localhost JclitTrain]# sudo perl $SPHINXTRAINDIR/scripts_pl/setup_SphinxTrain -task JclitTrain
```

Este último comando genera una estructura de carpetas que luego serán utilizadas para el entrenamiento. La carpeta donde se colocan los archivos de configuración es la carpeta “etc”. A continuación se muestra la estructura que se obtiene luego de ejecutar el comando “setup_SphinxTrain”.

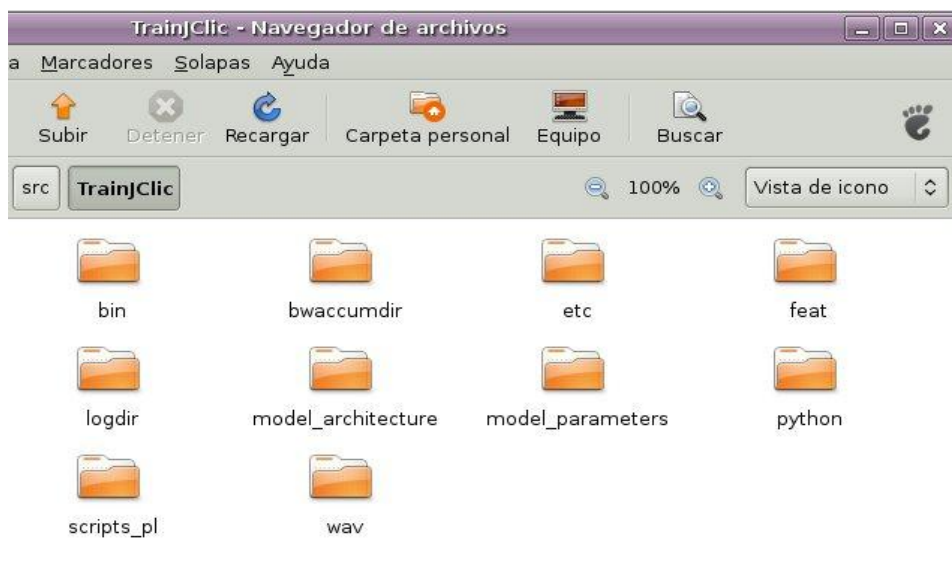


Figura 9.7 – Estructura de carpetas generada por SphinxTrain.

Dicha estructura contiene 10 directorios que se generan automáticamente. En algunos directorios como “bin” y “etc” se copian ciertos archivos de configuración de SphinxTrain.

Por ejemplo, en el directorio “bin” están todos los comandos ejecutables que son necesarios para cada una de las fases de entrenamiento.

En el directorio “etc” se encuentra un archivo de configuración llamado sphinx_train.cfg, y también en este directorio se añaden los datos de entrada al entrenador.

En el directorio wav, se deben colocar copias de los archivos de audio que componen el corpus, estos archivos se van a utilizar como modelo de expresión. De acuerdo a lo dicho

por algunos autores, es recomendable, utilizar al menos 160 wavs, conteniendo cada uno de ellos una única frase. En algunas pruebas que se realizaron para el presente trabajo, se hizo uso del programa Audacity⁴⁴ para generar los wavs.

En los directorios restantes se añadirán datos conforme se vayan ejecutando los scripts de SphinxTrain.

Una vez que está creada la estructura de directorios, el siguiente paso es incorporar la información de entrada al entrenador en el directorio “etc”:

- a) Diccionario (*JclitTrain.dic*)
- b) Archivo de control (*JclitTrain.fileids*)
- c) Lista fonética (*JclitTrain.phone*)
- d) Lista de silencios (*JclitTrain.filler*)
- e) Archivo de transcripción (*JclitTrain.transcription*)

En el archivo .dic se deben colocar la lista de palabras que se desea utilizar, junto con la transcripción fonética. En este caso, no interesan los fonemas utilizados ya que luego se entrenarán para que sean aceptados.

En la figura 9.8 se muestra un ejemplo de archivo del diccionario creado para el entrenamiento.

⁴⁴ <http://audacity.softonic.com/>.

1	<sil>	SIL
2	CERO	S EH R OW
3	UNO	UH N AO
4	UNO(2)	UW N OW
5	DOS	D AO S
6	DOS(2)	D OW S
7	TRES	T R EH S
8	TRES(2)	T EH S
9	CUATRO	K UH AA T R AO
10	CUATRO(2)	K UH AA T OW
11	CINCO	S IY N K AO
12	CINCO(2)	T IY N K AO
13	SEIS	S EH IH S
14	SEIS(2)	S EH IY S
15	SIETE	S IH EH T EH
16	SIETE(2)	S IY EH T EH
17	OCHO	AO CH AO
18	OCHO(2)	AO CH OW
19	NUEVE	N UH EH V EH
20	NUEVE(2)	N UH EH B EH
21	ACEPTAR	AH S EH P T AH ER
22	CON	K OW N
23	CANCELAR	K AH N S EH L AH ER
24	SIGUIENTE	S IY G IY EH N T EH
25	ANTERIOR	AH N T ER IY O ER

Figura 9.8 – Ejemplo de diccionario generado para el entrenamiento.

Además del diccionario de palabras a utilizar, se deberá definir el diccionario de silencios con extensión .filler:

1	<s>	SIL
2	</s>	SIL
3	<sil>	SIL

Figura 9.9 – Ejemplo de diccionario de silencios generado para el entrenamiento.

En este archivo solo se deben especificar los símbolos que se van a utilizar para representar los silencios.

En el archivo .phone se colocan los fonemas utilizados en el diccionario. No deben repetirse las unidades fonéticas. Es importante aclarar que este archivo tiene que contener la unidad fonética SIL, utilizada para representar los silencios. En la figura 9.10, se muestra el archivo .phone generado para la presente prueba.

1	AA
2	AE
3	AH
4	AO
5	B
6	CH
7	D
8	G
9	EH
10	ER
11	IH
12	IY
13	L
14	K
15	N
16	OW
17	P
18	R
19	S
20	SIL
21	T
22	UH
23	UW
24	V

Figura 9.10 – Lista de fonemas generados para el entrenamiento.

Luego de generar estos archivos, también es necesario armar los archivos de .fileids y el de .transcription. Estos se utilizan para ordenar la información que proveen los wav. Aquí no se mostrarán los archivos creados, ya que poseen un gran tamaño porque las pruebas fueron realizadas utilizando 220 wavs de voz, y por lo tanto, ambos archivos tienen 220 líneas. A modo de ejemplo, se mostrará una línea de cada uno para dejar en evidencia el formato a utilizar.

```
1 train/001
```

Figura 9.11 – Formato del archivo .fileids.

El archivo .fileids contiene la ruta y el nombre de los archivos wav que se utilizan en el entrenamiento. La ruta ingresada es relativa a la carpeta wav y el nombre del archivo no debe tener la extensión (esta será especificada en el archivo sphinx_train.cfg).

A continuación se muestra el formato del archivo de transcripción. Aquí, se especifican las palabras que son pronunciadas en el wav. Los tags "<s>" y "</s>" que se muestran en la

figura 9.12, se colocan para definir el silencio antes y después de pronunciar la palabra. El número que está entre paréntesis, luego del tag de cierre, especifica el nombre del archivo .wav.

```
1 <s> UNO(2) </s> (001)
```

Figura 9.12 – Formato del archivo .transcription.

Es importante aclarar que tiene que existir consistencia entre estos dos archivos, ya que en la bibliografía consultada aparecían diferentes formatos que llevaron a introducir errores en el proceso. Es por ello que, se quiere dejar explicitado en detalle cada uno de estos aspectos.

Por último, antes de continuar con la ejecución de los comandos es necesario tener en cuenta los siguientes puntos:

1. El contenido de los archivos .dic, .filler, .phone y .transcription deben estar en mayúscula como se muestra en los ejemplos.
2. No debe haber líneas en blanco en ningún archivo.
3. Los archivos .transcription y .fileids deben tener el mismo número de líneas y cada una de estas se deben corresponder entre sí.
4. El archivo .phone no debe tener entradas duplicadas.

Si bien el diccionario (.dic), el archivo de silencios (.filler) y el de fonemas (.phone) pueden ser creados manualmente, existe una herramienta provista por CMU que realiza esta tarea. Para esto, se debe ingresar al sitio <http://www.speech.cs.cmu.edu/tools/lmtool-adv.html>.

Este sitio web requiere varios archivos, pero alcanza con uno similar al de transcripción (sin los tags <s></s>). Una vez finalizado el proceso, se puede descargar el archivo de sentencia y renombrarlo a JclitTrain_train.transcription. Además, del diccionario y el Modelo de lenguaje con extensión .lm. Este último no es necesario en la etapa de entrenamiento, pero se utiliza cuando se pone en ejecución el reconocedor, incorporándolo desde el archivo de configuración. Este archivo .lm debe ser convertido a extensión .dmp utilizando el paquete lm3g2dmp de Linux, disponible en <http://sisyphus.ru/en/srpm/Sisyphus/lm3g2dmp/get>. Luego de descargar la herramienta, se requiere ingresar al directorio correspondiente a lm3g2dmp y ejecutar el comando make.

Para convertir el archivo a .dmp desde la consola, la forma es la siguiente:

```
#!/lm3g2dmp <archivo_origen>.lm <archivo_destino>
```

Este comando convertirá el archivo .lm a .lm.dmp.

El archivo de configuración "sphinx_train.cfg", contiene los criterios a tomar en cuenta en el proceso de entrenamiento, tales como, la ruta donde se encuentra SphinxTrain, la ruta al directorio raíz, la extensión de los archivos de extracción de características, y el tipo de modelo a crear.

Las variables o criterios que se tienen que cambiar para adaptarlo a la configuración de los archivos generados anteriormente son:

```
$CFG_DB_NAME= 'JclitTrain'  
$CFG_BASE_DIR= '/usr/local/src/JclitTrain'  
$CFG_SPHINXTRAIN_DIR= '/usr/local/src/SphinxTrain'  
  
$CFG_WAVFILES_DIR = "$CFG_BASE_DIR/wav";  
$CFG_WAVFILE_EXTENSION = 'wav';  
$CFG_WAVFILE_TYPE = 'mswav'; # one of nist, mswav, raw'
```

Estas variables deben ser modificadas para el correcto funcionamiento del sistema. Otro aspecto con el que se puede experimentar, es el efecto de la variación en el valor de los parámetros contenidos en variables.def que se nombró en la sección anterior. Para tener mayor conocimiento acerca del impacto que tiene el cambio de valores en esos indicadores sobre el proceso de entrenamiento, se dan a continuación algunos criterios tomados de <http://mit.ocw.universia.net/6.345/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-345Automatic-Speech-RecognitionSpring2003/6563BA98-DD44-4DB1-B28B-9B1CCD0EF7C1/0/assignment8.pdf>:

- **set statesperhmm:** esta variable puede contener valores entre 3-5 en sistemas estándar. El número de estados de un HMM está relacionado con las características variables en el tiempo de las unidades de sonido. Las unidades de sonido que presentan una gran variación temporal necesitan más estados que las representen. La naturaleza de la variación temporal de los sonidos está también, en parte, capturada por la variable "skipstate" (saltar estado) que se describe a continuación.
- **set skipstate:** esta variable se ajusta a los valores "no" o "si". Controla la topología de los HMM. Cuando se ajusta a "si", permite que los HMM salten estados. Si se ajusta esta variable a "no", cualquier estado determinado puede sólo hacer transición al próximo estado. En todos los casos, se permiten auto-transiciones.
- **set gaussiansperstate:** esta variable se puede ajustar a cualquier número entre 4 y 8. La distribución de cada estado de cada HMM se modela por una mezcla de gaussianas. Esta variable determina el número de gaussianas de esta mezcla. El número de parámetros HMM que van a estimarse, aumenta a medida que sube el número de gaussianas de la mezcla. Por tanto, aumentar el valor de esta variable, puede provocar que haya menos datos disponibles para estimar los parámetros de cada gaussiana. Sin embargo, aumentar su valor también produce modelos más elegantes, que pueden conducir a un mejor reconocimiento. Es posible superar el

problema de insuficiencia de datos compartiendo las mezclas de gaussianas entre muchos estados HMM. Cuando múltiples estados HMM comparten la misma mezcla de gaussiana, se habla de estados compartidos o atados (también denominados senones). El número de mezclas que se entrenen serán, en última instancia, exactamente igual al número de estados atados que se especifique, que a su vez puede estar controlado por el parámetro "n_tied_states".

- **set n_tied_states:** esta variable se puede ajustar a cualquier valor entre 500 y 2500. Permite especificar el número total de distribuciones de estado compartidas de su grupo final de HMM entrenados. Los estados se comparten para superar problemas de insuficiencia de datos para cualquier estado de cualquier HMM. La distribución se realiza de tal modo que se mantenga la individualidad de cada HMM, en el sentido de que sólo los estados con las distribuciones más parecidas son atados. El parámetro n_tied_states controla el grado de atadura. Si éste es pequeño, es posible que un gran número de estados sin parecido alguno pueden estar atados, provocando una reducción en el rendimiento del reconocimiento. Por otro lado, si este parámetro es muy grande, pueden existir datos insuficientes para aprender los parámetros de las mezclas de gaussianas para todos los estados atados. Luego de realizado el entrenamiento, se puede observar qué estados ha atado el sistema, en los archivos .mdef de la carpeta "model_architecture". Estos archivos enumeran los fonos y trifonos para los que se entrenan los modelos, y asigna identificadores numéricos a cada estado de sus HMM. Cuando se genere el archivo .props para la generación del .jar se debe asegurar que se referencia al archivo .mdef correcto y a los archivos del modelo acústico.

Una vez definidos todos los archivos de configuración, el primer paso para comenzar con el entrenamiento es extraer las características de los archivos que se encuentran en el directorio wav. Esto se realiza gracias a un programa, escrito en perl, llamado "make_feats". Allí se llama al ejecutable "wave2feat", el cual se encarga de convertir el audio en coeficientes cepstrales (MFCC).

Para ejecutar el comando se debe escribir la siguiente línea en la consola:

```
sudo perl scripts_pl/make_feats.pl -ctl etc/JclicTrain.fileids
```

De ésta forma se obtienen los archivos con extracción de características que se copiarán automáticamente a la carpeta "feat". Si no hay errores en la etapa previa, se puede continuar con el siguiente comando:

```
perl scripts_pl/make_feats.pl -ctl etc/pl1_train.fileids perl scripts_pl/RunAll.pl
```

En este comando es donde se generan el resto de los archivos necesarios para ser utilizados en el reconocedor. Este procedimiento pasa a través de las diferentes fases del modelo de formación acústica y puede tomar algunos minutos, dependiendo de la cantidad de datos

que se tienen. Si hay algún error, se muestra en la consola y además se registran los errores en un archivo html dentro de la carpeta del modelo a entrenar, en este caso es JclitTrain. Este archivo, incluye advertencias y si hay “errores fatales” se brinda información más amplia para que puedan ser solucionados.

Con estos pasos se generan los archivos necesarios para ser utilizados por el reconocedor. Para usarlos se debe proceder como se mostró en la sección anterior. Aquí solo se detallará la ubicación de los archivos que se deben utilizar para la generación del .jar y que están dentro de la carpeta JclitTrain:

1. El archivo .dic, .lm.dmp, .filler se encuentran en la carpeta “etc”.
2. Los archivos main, variances, mixture_weights y transiton_matrices se encuentran dentro de la carpeta “model_parameters/JclitTrain.ci_cont”.
3. Los archivos .mdef se encuentran dentro de la carpeta “model_architecture”.

Con estos datos ya se puede comenzar a generar el archivo .jar como se mostró en la sección anterior.

En resumen, en esta sección se presentó en detalle los pasos a realizar para llevar a cabo el proceso de entrenamiento previo al reconocimiento. Esta información fue recopilada en base a la bibliografía consultada y a las experiencias realizadas por las autoras de la tesina.

INTEGRACIÓN DEL RECONOCEDOR CON JCLIC

En esta sección se describirán cuestiones del prototipo que incluyen a ambos componentes utilizados para la integración.

Una de ellas, es cómo se realizó la incorporación del framework de reconocimiento de voz a JClit. Para ello se creó una clase que representa al reconocedor, llamada VoiceRecognizer, donde se encuentran sus principales métodos, tales como el método que se utilizó para crearlo, así como también, el método que se encarga de realizar el reconocimiento. Se generó un paquete llamado “reconocimiento” dentro del paquete “src” visto en el capítulo 8 para alojar esta clase. Luego, esta clase es utilizada en el método constructor de la clase Player, si el usuario eligió trabajar con reconocimiento de voz. Allí se crea el reconocedor y se puede comenzar a utilizarlo. En la siguiente figura, se muestra la parte del código donde se crea una instancia del reconocedor.

```
int result = JOptionPane.showConfirmDialog((Component) null,
    "Desea realizar la actividad con reconocimiento de voz ? ",
    "Confirmacion", JOptionPane.YES_NO_OPTION);
thisPlayer.playWithRecognition = (result == JOptionPane.YES_OPTION);
if (thisPlayer.playWithRecognition) {
    System.out.println("Comenzando configuracion para reconocimiento de voz...");
    thisPlayer.reconocedor = new VoiceRecognizer();
}
```

Figura 9.13 – Código donde se instancia el reconocedor.

En un primer momento, se pensó en invocar al reconocedor desde la clase SimpleAssociation pero, al llevarlo a la práctica, no permitía navegar entre las diferentes actividades, por lo que se tuvo que pensar en una solución más amplia.

Otro obstáculo que se presentó al momento de incorporar el reconocimiento de voz, es que en la bibliografía de Sphinx, que explica la utilización del framework, lo implementa incluyendo una iteración infinita en el método que lo invoca, lo cual, en el presente desarrollo interfería con la creación y visualización de la interface de JClic. Esta última utiliza hilos para ser dibujada en pantalla y el mecanismo hace uso de una interfaz llamada SwingWorker⁴⁵. Mientras que la clase del reconocedor utilizaba Threads, dando como resultado que ambos se ejecutaran sobre el mismo hilo y no se pudiera lograr el efecto esperado. Para solucionar este problema, se hizo que la clase que representa al reconocedor heredara de SwingWorker aunque no utilice Swing, de manera que JClic y el reconocedor se ejecuten en hilos separados, interactuando entre ellos, para paralelizar tareas. De esta manera, ambos componentes, pueden ejecutarse sin problemas. En la siguiente figura se muestra la declaración de la clase VoiceRecognizer:

```
public class VoiceRecognizer extends SwingWorker {
```

Figura 9.14 – Declaración de la clase VoiceRecognizer

Para llevar a cabo la tarea de resolver una actividad de tipo Asociación Simple, lo que se implementó fue, que al crearse, el reconocedor ejecute un método llamado getCommand() en la clase que representa dicha actividad. Este método es el encargado de procesar la entrada de voz del usuario y tomar las decisiones correspondientes.

Para la implementación de la resolución de la actividad, se utiliza un arreglo (llamado cells) de dos posiciones que representan el par de celdas a unir en cada paso. Este arreglo se encuentra inicializado con valor 1.

Cuando el usuario pronuncia el primer número, luego de que el reconocedor devuelva el valor pronunciado, se coloca el dígito en la posición 0 del arreglo antes mencionado. Si posteriormente, se pronuncian más dígitos, será reemplazado en la posición 0 del arreglo.

La palabra “Con” se utiliza como conector de celdas. Es decir, el alumno deberá pronunciar “Uno con dos” si desea unir la celda 1 con la celda 2. Desde el punto de vista de la implementación, cuando se pronuncia la palabra “Con” se incrementa en uno el índice del arreglo para almacenar en la segunda posición el nuevo dígito que será pronunciado. El índice del arreglo se mantiene en una variable que recibe el nombre de currentCell.

Al recibir la entrada de voz “Aceptar”, el sistema muestra un cartel con los valores que se van a procesar, el usuario deberá confirmar estos valores para que la acción se lleve a cabo.

⁴⁵ SwingWorker - <http://java.sun.com/docs/books/tutorial/uiswing/concurrency/worker.html>

Para la confirmación es necesario pronunciar nuevamente la palabra “Aceptar”. Luego de confirmado, se invoca a un método que se encarga de ejecutar la acción que el usuario desea realizar. En este método se buscan los casilleros nombrados, si existen y no fueron elegidos antes. Luego, se verifica dentro de la estructura interna del elemento, si forman una correspondencia correcta, es decir, si las celdas seleccionadas son parte de la solución. Si es así, se eliminan de los posibles elementos a elegir y se continúa con la próxima correspondencia, hasta llegar a la última. Cuando se llega a la última, se da por terminada la actividad.

Como se mencionó en el capítulo 8, JClic provee un módulo capaz de contabilizar el tiempo empleado en cada actividad, intentos, aciertos, etc. Si bien el tiempo puede variar si se utiliza reconocimiento de voz, se pensó en mantener igualdad en el contador de intentos y aciertos para que el docente pueda evaluar al alumno que está resolviendo la actividad. Es por esta razón, que se decidió agregar un cartel donde el usuario vea y confirme que es lo que desea unir, esto es porque existe, en la mayoría de los reconocedores, cierta tasa de error, con lo cual, podría darse la situación en que se procese una entrada errónea y JClic lo contara como intento fallido, perjudicando la evaluación del alumno. Con los agregados mencionados, el docente que creó la actividad podrá usar el contador de errores que provee, por defecto, JClic.

En la siguiente sección se presentará una guía demostrativa de cómo se realiza una actividad de asociación simple con el prototipo propuesto.

EJECUCIÓN DE UNA ACTIVIDAD USANDO EL PROTOTIPO

A continuación se presenta, a través de un ejemplo, como sería la ejecución de una actividad con el prototipo de adaptación propuesto.

En primer lugar, se repasa lo dicho en secciones previas. Para resolver la actividad se debe pronunciar, por ejemplo, “Uno con Tres Aceptar”; lo que se interpreta de esta sentencia es lo siguiente:

El primer número representa un casillero del primer conjunto de información, la palabra “con” indica que se va a nombrar el casillero del segundo conjunto, representado por el segundo número de la frase. La palabra “Aceptar” indica que el usuario quiere realizar la unión de los casilleros nombrados.

Además se agregó el código necesario para que el programa pida confirmación sobre la información recolectada mostrando un cartel en pantalla.

En la figura 9.15 se muestra la pantalla inicial que muestra JClic cuando se inicia.

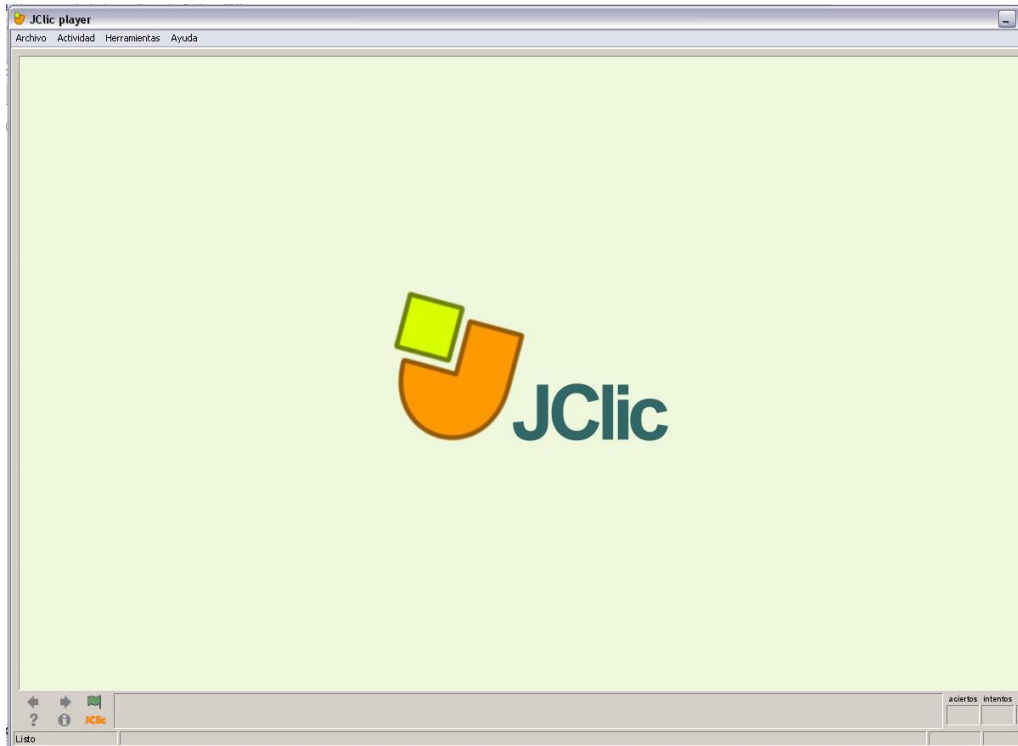


Figura 9.15 – Pantalla inicial de JClic.

Luego desde el menú de Archivo se elige el proyecto que se desea cargar. Antes de comenzar la carga, se muestra un cartel dónde pregunta si se desea utilizar reconocimiento de voz o no.

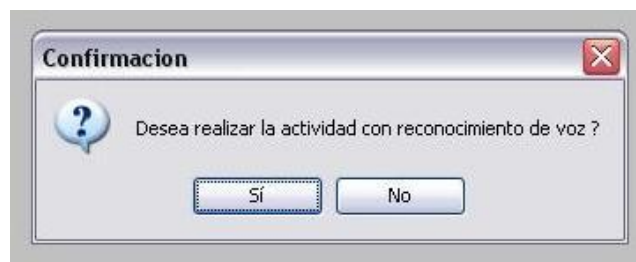


Figura 9.16 – Cartel de inicio de proyecto.

En este momento, se debe elegir como se realizará la actividad, si se desea resolverla con reconocimiento de voz, se debe realizar un clic en el botón si, momento en el cual se pondrá en funcionamiento el motor de reconocimiento de voz y mostrará la actividad.

A continuación se muestra una imagen de la actividad de ejemplo, en la que se presentan imágenes de animales en los casilleros de arriba y sus nombres en orden aleatorio en el conjunto de casilleros debajo. El usuario debe unir la imagen con el nombre correcto.

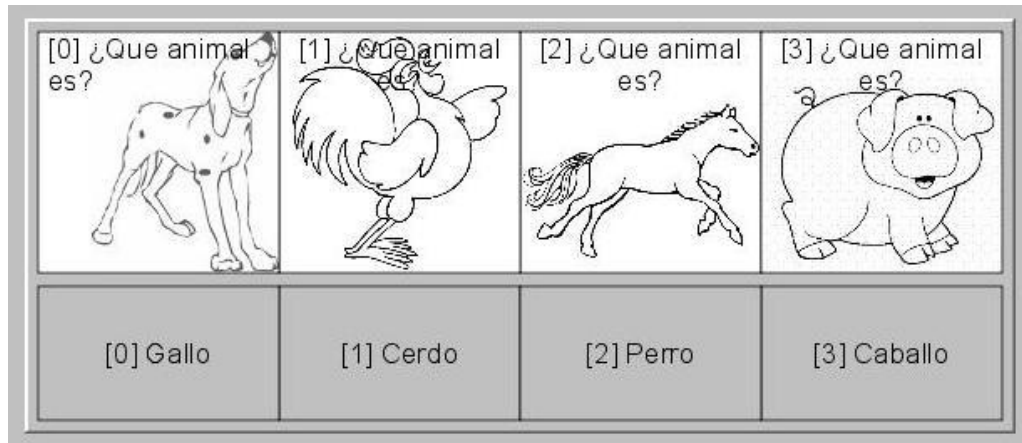


Figura 9.17 – Actividad de asociación simple de ejemplo.

Ahora es el momento de resolver la actividad, de manera que el usuario debe pronunciar una frase como “Tres” “con” “Uno” “Aceptar”. Luego de procesada la palabra “Aceptar”, el programa mostrará un cartel de confirmación como el que se ilustra en la siguiente figura.

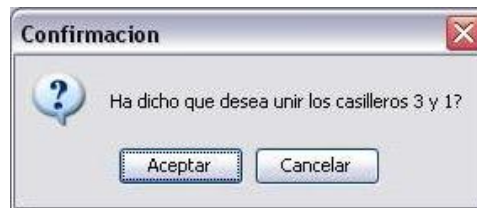


Figura 9.18 – Cartel de confirmación.

Si lo que muestra el cartel es correcto, en otras palabras, es lo que el usuario dijo, deberá decir “Aceptar” nuevamente para que el programa realice la acción. En caso contrario, deberá pronunciar la palabra “Cancelar”. Si el usuario dice “Cancelar”, se cierra la ventana de confirmación y se continúa con la ejecución de la actividad, debiendo el usuario pronunciar nuevamente los casilleros que desea unir. Si dice “Aceptar” y la elección del usuario fue correcta, es decir, los casilleros que nombró se corresponden, la imagen a continuación es la que el programa mostrará indicando que se acertó en la elección.

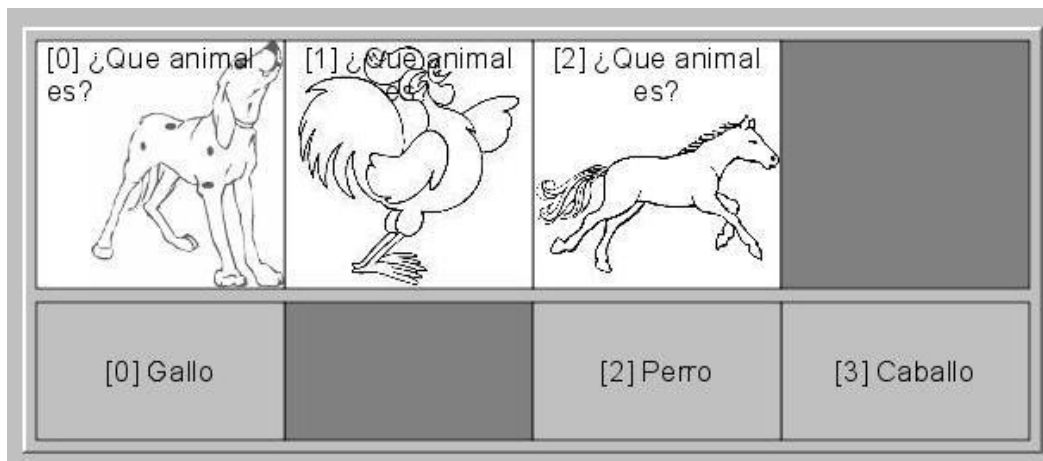


Figura 9.19 – Actividad con una jugada resuelta.

En caso de que la elección haya sido incorrecta, los casilleros se seguirán mostrando, y se reproduce un sonido que indica error.

De esta manera se continúa, encontrando las correspondencias hasta completar la actividad.

Como se dijo anteriormente, una vez finalizada la actividad el usuario podrá pronunciar las palabras “siguiente” y “anterior” para navegar entre las diferentes actividades del proyecto.

Es importante aclarar que si la actividad tiene contenido multimedia, como sonidos de animales, es conveniente utilizar auriculares, ya que la reproducción del sonido podría interferir en el proceso de reconocimiento.

En la siguiente sección, se explicará cómo se podría llevar a cabo la tarea de extender el prototipo, presentado en apartados anteriores, al resto de las actividades que JClic permite crear.

EXTENSIÓN DEL PROTOTIPO

En esta sección se explica la manera en que debieran ser adaptados los diferentes tipos de actividades de JClic, para integrarse con el reconocedor de voz. No se explicará en detalle qué implica cada tipo de consigna, ya que esto fue tratado en el capítulo referido a JClic. Sólo se darán algunas estrategias para la integración.

Se comenzará con las **asociaciones**, de las cuales existen dos tipos, las simples y las complejas. El prototipo presentado abarca las asociaciones simples, de manera que en esta sección, se centrará la atención en las asociaciones complejas. Debido a que son similares a las simples, también se deberán etiquetar los casilleros de la misma manera que se mostró en el prototipo. Así el alumno que está resolviendo la actividad, debe decir una frase como por ejemplo: “Uno con Cinco Aceptar” para lograr la unión entre dos casilleros, previamente identificados. En cuanto a la implementación, se deberá estudiar la forma en que JClic realiza esta tarea y crear un método que utilizando el reconocedor de voz, pueda obtener los elementos que el usuario desea unir. También se debe estudiar en qué momento es conveniente colocar las etiquetas a los elementos, aunque se puede prever que la implementación será análoga a la realizada.

En los **juegos de memoria** hay que ir descubriendo parejas de elementos iguales o relacionados entre ellos, que se encuentran escondidos. Extender el prototipo para que este tipo de actividad se pueda resolver con reconocimiento de voz, sería muy similar a lo que se debe realizar para las asociaciones complejas, pues también se necesita una forma de identificar los elementos. En esta actividad, se debe recordar que el objetivo es descubrir dos casilleros cada vez para encontrar los iguales. Por eso, es necesario nombrar qué casillero se quiere unir con qué otro casillero. Luego, se debe estudiar cómo se resuelve la actividad para crear un método análogo que utilice reconocimiento de voz, en lugar de

eventos del mouse como usa JClic originalmente. Con respecto a identificar los elementos de la actividad, también se deberá realizar con el mismo procedimiento que las asociaciones complejas. En la figura 9.20 puede verse un ejemplo de Juego de memoria.

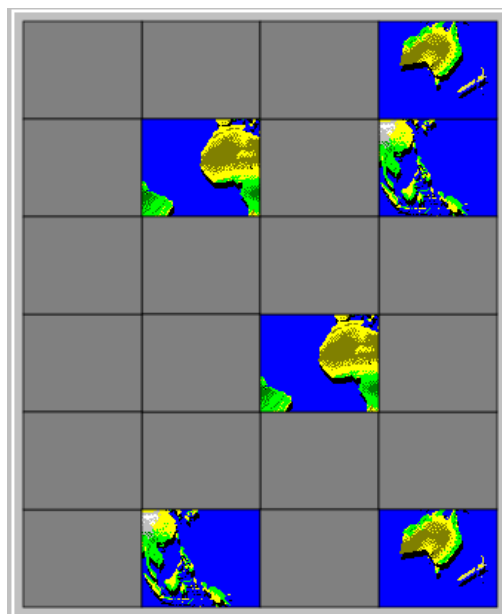


Figura 9.20 – Ejemplo de un juego de memoria.

Las **actividades de exploración, identificación e información** parten de un único conjunto de información. En esta categoría de actividades, también se deberá identificar los elementos a través de una etiqueta, de manera que cuando el usuario quiera activar el contenido de dicho elemento, utilice la misma para realizar la acción. En el caso del ejemplo que se muestra a continuación, una posible forma de resolverlo podría ser etiquetando todas las palabras y que el alumno deba pronunciar solo las que cumplan con la condición que indica el enunciado. Al pronunciar la palabra aceptar, se activa la acción correspondiente al botón “Corregir el ejercicio”. En cuanto a la implementación, se haría de manera similar a cómo se resolvió con las asociaciones simples. En la figura 9.21 para poder señalar los sustantivos dentro del texto, el alumno debiera nombrarlos mediante comandos por voz. Así, por ejemplo, debiera mencionar “parques” para incluirlo en la solución. Cada sustantivo nombrado estará separado por un espacio de silencio. Sólo al decir “Aceptar” se concluirá la resolución del ejercicio.

Señala los sustantivos de este texto

Parques y Jardines etiquetará árboles de Barcelona. Serán entre 5.000 y 10.000 los ejemplares a los cuales se adjuntará un cartel con su nombre común y científico y su lugar de origen. El objetivo de la iniciativa es incrementar el conocimiento de los árboles entre los ciudadanos. La campaña, titulada "Los árboles tienen nombre", abrirá a partir de hoy una consulta pública para seleccionar propuestas sobre la forma, tipo y sistema de rotulación.

"El Periódico"
24 de enero de 1999

Corregir el ejercicio

Figura 9.21 – Ejemplo de actividad para identificar palabras.

Los **puzzles** plantean la reconstrucción de una información que se presenta inicialmente desordenada. Esta información puede ser gráfica, textual, sonora o combinar aspectos gráficos y auditivos al mismo tiempo. En esta categoría se encuentran tres tipos distintos de puzzles. En los puzzles de tipo doble y de intercambio, se deberán etiquetar las piezas del panel desordenado y las posiciones en el panel vacío de la misma manera que las soluciones planteadas antes. Por ejemplo, en el puzzle que se muestra en la figura 9.22, el alumno debiera decir el número que identifique el casillero vacío a completar, junto con el número que identifica la pieza (del panel izquierdo) que se corresponde con dicho espacio. Se deberá estudiar cómo se resuelve esta actividad para trasladar la lógica a un método nuevo que use reconocimiento de voz para resolverla. Con esto se quiere decir que no se ha estudiado aún el detalle de implementación que posee JClic en este tipo particular de consigna. En el puzzle de tipo de agujero, la modificación será análoga a la de los otros tipos de puzzles, pero sólo se deberán etiquetar las piezas y no la posición vacía (agujero). Se podría definir otra forma de procesar la información recolectada por el reconocedor de voz, ya que sólo se necesitará nombrar un elemento. Por ejemplo, en la figura 9.24 se podría etiquetar cada una de las figuras con números entre el 1 y el 11 dejando libre el casillero sin figura. Si se quisiera trasladar la pieza 8 al hueco, sólo habría que nombrar "ocho" y luego la palabra "Aceptar".

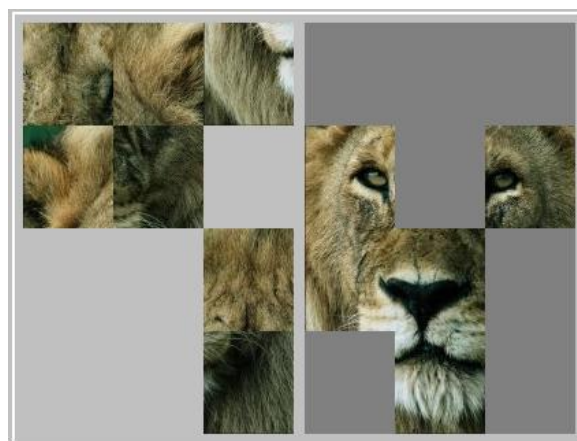


Figura 9.22 – Ejemplo de puzzle doble.

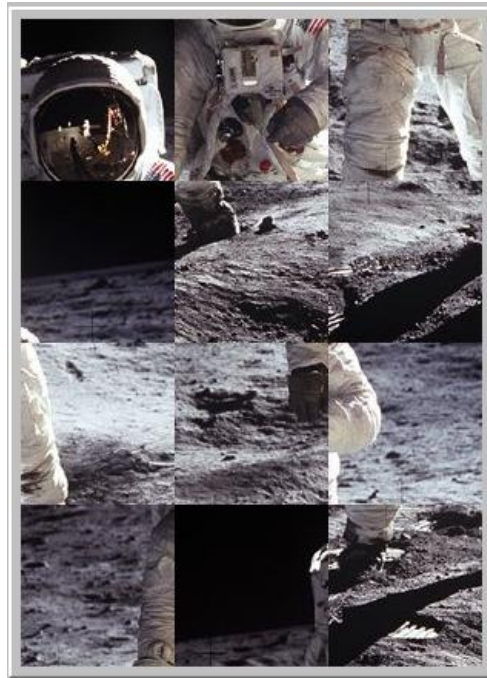


Figura 9.23 – Ejemplo de puzzle de intercambio.

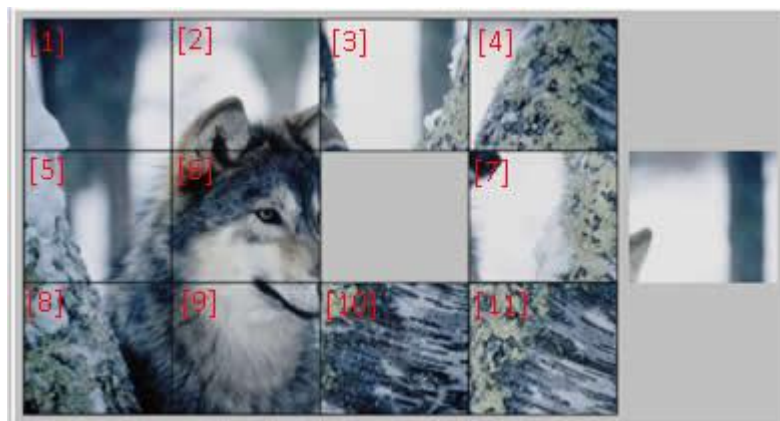


Figura 9.24 – Ejemplo de puzzle de agujero.

Las actividades de **respuesta escrita** se resuelven escribiendo un texto (una sola palabra o frases más o menos complejas). Para este tipo de actividades, la modificación resulta un poco más compleja, pues será necesario modificar el diccionario y la gramática del reconocedor para aceptar una mayor cantidad de palabras. Luego, se deberán etiquetar los espacios a completar por el usuario de manera que se los pueda nombrar. Una vez nombrada la posición que se desea completar, el usuario dirá la frase o palabra que cree que es la solución. Será necesario, estudiar el comportamiento de JClíc para resolver esta actividad y modificarlo para recrear este comportamiento, pero utilizando como dato de entrada la voz del usuario.

Las **actividades de texto**, plantean ejercicios basados siempre en las palabras, frases, letras y párrafos de un texto que hay que completar, entender, corregir u ordenar. Los textos pueden contener también imágenes y ventanas con contenido activo. Para este tipo de actividades, la modificación necesaria será más laboriosa, pues además de proveer un mecanismo para identificar los distintos componentes presentes en la actividad, se deberá contar con un vocabulario más amplio que el del presente prototipo, de manera de cubrir las posibles palabras, signos de puntuación, etc. necesarios para este tipo de actividad. Entonces la clave será abordar la adaptación del reconocedor de manera tal que pueda tener un diccionario más completo.

Tanto en las actividades de respuesta escrita como de texto, será necesario que el docente cuente con una herramienta que le permita editar, de manera transparente, tanto el diccionario como la gramática.

Las **sopas de letras** y los **crucigramas** son variantes interactivas de los conocidos pasatiempos de palabras escondidas. La adaptación para este tipo de actividades será similar a la de actividades de texto, aunque se podría buscar otra solución nombrando las letras. Una alternativa de solución más simple para las sopas de letras, sería que el alumno sólo deba nombrar el casillero inicial y el final de la palabra encontrada, y luego se muestre un cartel en la pantalla para confirmar si la palabra es la elegida. El alumno en esa instancia deberá decir “Aceptar” para confirmar, tal como se ha planteado en el prototipo.

Con este tipo de actividad se ha finalizado el recorrido por los distintos tipos de consigna que provee, al momento, JClic. Se presentó una propuesta de solución para adaptar cada una de ellas, sin entrar en detalles de implementación.

En la siguiente sección se presentarán datos útiles para adaptar cualquier otra aplicación con un motor de reconocimiento.

CONSIDERACIONES GENERALES PARA INTEGRAR RECONOCIMIENTO DE VOZ A UNA APLICACIÓN

Se detallarán en este apartado algunas de las consideraciones a tener en cuenta para lograr la integración de comandos por voz en una aplicación.

En primer lugar, será necesario tener conocimiento sobre la disponibilidad del código fuente de la aplicación y del reconocedor. Además, se deberá tener en cuenta con qué licencia se distribuyen, pues ello influirá en lo que se podrá hacer con ellos. Sería importante considerar, mantener el mismo tipo de licencia luego de adaptar la aplicación, de manera tal de no restringir su disponibilidad.

Resultará de útil conocimiento, también, el lenguaje de implementación utilizado en cada uno, debido a que será necesario integrarlos para que funcionen juntos. Por otro lado, al

igual que con la licencia, es importante que la aplicación resultante mantenga las mismas características de compatibilidad y portabilidad que proveía inicialmente.

Teniendo en cuenta lo anterior, se nombran algunos de los pasos que se deberán considerar si se desea adaptar cualquier programa mediante comandos por voz:

1. Configurar el diccionario, la gramática y demás componentes (modelo acústico y modelo de lenguaje, por ejemplo) que se desean utilizar en el motor de reconocimiento de voz propuesto. Para este punto será de utilidad la lectura de los capítulos 5, 7 y las secciones pertinentes del presente capítulo.
2. Estudiar y planificar una estrategia que permita instanciar al reconocedor, dentro de la aplicación a adaptar. En el caso de este trabajo, se tuvo que generar una nueva clase que represente al RV. Dentro de esta clase, se implementaron métodos, por ejemplo, para que se mantenga en escucha mientras se está resolviendo una actividad.
3. Establecer una correspondencia entre los comandos por voz a utilizar y las acciones dentro del programa adaptado. En este caso, es necesario pensar una estrategia de solución para la aplicación particular que se desea adaptar, similar a lo que hemos realizado en este capítulo para los distintos tipos de consignas que presenta JClic.
4. Modificar el código correspondiente a las acciones del programa a adaptar, para que sean activadas mediante comandos por voz. En este caso se hace necesario estudiar la lógica de la aplicación. Por ejemplo, para lograr esto en JClic se estudiaron los métodos correspondientes a las clases SimpleAssociations y Player, y se modificaron para responder a los comandos por voz.
5. Optimizar la ejecución de la aplicación adaptada, a través del uso de hilos de ejecución. Esto es posible en el caso que los lenguajes correspondientes lo permitan.

Con estas consideraciones se intenta mostrar una base para integrar una aplicación con un motor de reconocimiento de voz, independientemente del producto del mercado seleccionado.

En la siguiente sección, se presentarán algunos resultados de la evaluación realizada al prototipo. Se introducirán las razones por las cuales se escogió la realización de pruebas a través de juicio de expertos, y se definirá la metodología utilizada.

EVALUACIÓN DEL PROTOTIPO DESARROLLADO

Se decidió someter el prototipo presentado a prueba de expertos para que ellos expresaran sus opiniones respecto de este trabajo.

Se consideró más apropiado realizar primero este tipo de prueba, y analizar los resultados para tomarlos como líneas futuras de trabajo e investigación. Después de esta etapa será posible testear el prototipo con los usuarios finales, los cuales serían, docentes y alumnos. Se optó por este tipo de evaluación antes de someter el proyecto a una evaluación por los usuarios finales, debido a que resulta difícil el acceso a los alumnos, en particular, a aquellos con algún tipo de condición especial, pues sus padres/tutores no desean exponerlos a situaciones que puedan resultar negativas para el alumno. Es por ello que es conveniente llegar, en este caso, al usuario final con una versión más avanzada del producto. Este tema fue abordado con gente vinculada al trabajo con alumnos con necesidades especiales.

El juicio de expertos tiene como ventaja la calidad de la respuesta y el nivel de profundización por parte del experto.

Como instrumento de evaluación se eligió una encuesta con preguntas abiertas y cerradas, de manera tal de poder recoger la información que se cree necesaria para someter a juicio el prototipo.

Además de la encuesta, se elaboró una especie de tutorial para contar a los participantes de la evaluación la motivación del trabajo, los objetivos y las características principales de la solución propuesta.

En el anexo A, se presentarán las encuestas realizadas a profesionales de diferentes áreas que se relacionan con distintas temáticas abarcadas en el presente proyecto, utilizando el método de agregados individuales, el cual se explicará en la siguiente sección.

JUICIO DE EXPERTOS

Las opiniones o juicios de los expertos son una alternativa que combina las aproximaciones analíticas y los métodos de estimación subjetiva de probabilidades desarrollados conforme a las reglas de la teoría de la decisión **(ARQUER – 2010)**.

Los expertos se pueden pronunciar y ofrecer sus reflexiones acerca del objeto/s a evaluar.

Mediante el juicio de expertos, se pretende tener estimaciones razonablemente buenas, las mejores conjeturas, en situaciones donde no se pueden o no es conveniente obtener cuantificaciones exactas. Sin embargo, estas estimaciones pueden y deben ser confirmadas o modificadas a lo largo del tiempo, según se vaya recopilando información sobre el objeto de estudio.

JUICIO DE EXPERTOS: PROCESO

Inicialmente, se debe tener una idea clara acerca de qué datos se necesita recopilar a través de este proceso.

Como requisito previo, se necesita disponer de una definición y descripción claras de los objetos, tareas y/o actividades sobre los cuales los expertos deberán emitir un juicio acorde a las reglas que se estipulen a lo largo del proceso. A veces, es recomendable consultar a varios especialistas para asegurarse de que la descripción de la actividad a realizar por los mismos es clara y el nivel de detalle es suficiente.

Para seguir el proceso se debe seleccionar a los expertos. Para que una persona pueda ser etiquetada como “experta”, debe poseer un conocimiento profundo de la tarea o actividad que será objeto de análisis y valoración, y tiene que estar familiarizada con el sistema en el que ésta se desarrolla.

El número de expertos necesario para conseguir el juicio que se busca no es fijo, oscila desde tres o cuatro expertos hasta seis. Sin embargo, Villemeur⁴⁶ menciona que en algunas ocasiones pueden utilizarse más de ocho expertos, según la precisión que se desee.

Se debe tener previsto el modo en que se recogerán los juicios de los expertos y tener preparados, si es el caso, cuestionarios, escalas de evaluación, etc. La precisión en éstos es esencial para que los resultados no queden sesgados por fallos de comprensión. También se deben preparar las instrucciones que se van a dar a los jueces o expertos, de modo que comprendan con claridad cuál es el objetivo de su ejercicio de evaluación.

MÉTODOS PARA LA OBTENCIÓN DE JUICIO DE EXPERTOS

Los juicios de expertos se pueden obtener por métodos grupales o por métodos de experto único. Se pueden seguir, entre otros, el método de agregados individuales, el método Delphi, la técnica grupal nominal y el método de consenso grupal.

A continuación se hace una breve descripción de cada uno.

MÉTODO DE AGREGADOS INDIVIDUALES

Una posibilidad para llevar adelante el juicio, es que se pida individualmente a cada experto que dé una opinión acerca de cada una de las tareas/objetos a someter a evaluación. Luego, pueden tratarse estadísticamente los datos recogidos.

⁴⁶ VILLEMEUR, A. Reliability, availability, maintainability and safety assessment. Vol. 2. John Wiley & Sons, Chichester, 1992.

Éste es un método económico porque, al igual que el método Delphi (que se describirá a continuación), no exige que se reúna a los expertos en un lugar determinado. Puede parecer un método limitado porque los expertos no pueden intercambiar sus opiniones, puntos de vista y experiencia, ya que se les requiere individualmente; no obstante, esta limitación puede ser precisamente lo que se esté buscando, para evitar los sesgos de los datos ocasionados por conflictos interpersonales, presiones entre los expertos, etc.

MÉTODO DELPHI

En este método se debe comenzar, de manera que cada experto responda individualmente y en forma anónima a un cuestionario. Después se analizan las respuestas del conjunto de expertos, se remite a cada uno la respuesta mediana obtenida, y se les pide que reconsideren su juicio anterior, teniendo en cuenta estos datos. En cada una de las tres o cuatro “vueltas” siguientes, se informa a los expertos de cuál es la mediana del grupo y se les propone revisar su juicio anterior. Todo juicio individual que quede fuera del intervalo en que se mueve el grupo de expertos tiene que estar debidamente justificado o argumentado.

Una débil dispersión de los juicios obtenidos en la última vuelta indicaría que se ha alcanzado un consenso. La mediana de las respuestas obtenidas en esta última vuelta es el valor que se estaba buscando. Se emplea este estadístico de tendencia central (mediana), porque se presupone que las puntuaciones posibles de los expertos se distribuyen de forma asimétrica.

Con el método Delphi, los expertos comparten en cierto modo sus opiniones, sin que existan discusiones ni confrontaciones directas entre ellos.

TÉCNICA GRUPAL NOMINAL

El primer paso es reunir a los expertos (entre ocho y diez personas) y pedirles que registren, individualmente y sin intercambiar opiniones, sus propias puntuaciones y consideraciones para cada una de las tareas/objetos que se les detallan. Después, cada experto expone a los demás las puntuaciones y consideraciones registradas y al acabar esta ronda, todos disponen de la relación común del grupo. El paso siguiente consiste en un debate altamente estructurado de cada uno de los apartados de ésta. Finalmente, cada experto, de forma individual y por escrito, puntúa y argumenta las probabilidades de error u opiniones para cada tarea considerada.

En general, se procede como con el método Delphi, con la salvedad de que en esta técnica se permite algún debate entre los expertos, para que aclaren y compartan la información que cada uno está considerando. No obstante, las discusiones son limitadas, y así se limita también la presión de un experto sobre los juicios de otros. Aunque hay intercambio de pareceres, los juicios se emiten de forma individual, y la estimación final suele ser la media aritmética del conjunto de las estimaciones dadas por los expertos.

Para emplear esta técnica conviene que el grupo de expertos sea pequeño, se fomente la libre expresión y se eviten las críticas (discusiones tensas), el sistema de votación y el regateo. El éxito de la técnica depende, por una parte, de la habilidad y la experiencia de quien hace de moderador del grupo, y por otra, de la buena voluntad del grupo de expertos para trabajar juntos en un marco altamente estructurado.

MÉTODO DEL CONSENSO GRUPAL

Se necesita reunir a los expertos en un lugar determinado. Entonces se indica al grupo que su tarea consiste en reflexionar o evaluar una determinada tarea u objeto. Con estas instrucciones se maximizan los intercambios de información y opiniones dentro del grupo de expertos. Si el grupo no logra un consenso, puede intentarse un consenso artificial recogiendo las estimaciones individuales y sintetizándolas estadísticamente. Este método, como el anterior, también precisa que el grupo de expertos sea pequeño, se fomente la libre expresión y se eviten las discusiones tensas y los sistemas de votación.

ANÁLISIS DE LOS RESULTADOS

Cuando no se obtiene de los expertos un valor consensuado en grupo, porque se ha optado, por ejemplo, por el método de agregados individuales, es necesario reunir las estimaciones individuales de los expertos.

Para conseguir un juicio significativo es necesario que haya una cierta homogeneidad entre los expertos en la comprensión de la tarea u objteo que han de valorar y las causas potenciales de errores. Con el fin de obtener un resultado estadístico significativo, se calcula la consistencia interjueces. Obtener muchas opiniones diferentes no es aceptable, pues se corre el riesgo de dar una estimación que no sería representativa de nada; un análisis de varianza. La agrupación de los juicios de expertos se puede hacer estadísticamente.

En el caso de este trabajo se ha aplicado el juicio de expertos consultando un total de 6. En el Anexo A se explicita el cuestionario entregado a cada uno. En particular, se ha elegido el método de agregados individuales, sin embargo, las preguntas del cuestionario son abiertas y no se trabaja sobre resultados cuantitativos si no cualitativos, por el tipo de objeto a evaluar.

RESULTADOS OBTENIDOS EN LA ENCUESTA

Como se muestra en el anexo A, la encuesta se ha dividido en varias secciones, para que los encuestados respondan lo que es de su conocimiento y mantener ordenada la información. Cabe aclarar que, los expertos consultados pueden tener formación solamente en uno de los aspectos consultados en la evaluación.

Para realizar las conclusiones se respeta el mismo criterio, por lo que se dividirá esta sección en tres subsecciones: JClíc, Reconocimiento de voz e integración de comando por voz y JClíc.

A continuación se realiza un breve resumen de las principales opiniones dadas por los expertos en cada una de las tres subsecciones mencionadas. Más adelante, se presentarán algunas conclusiones a las que se ha arribado, a partir de estos resultados.

JCLIC

En la primera sección, la totalidad de los encuestados conocían la herramienta JClíc, coincidieron en que es una herramienta muy utilizada en el ámbito educativo y que podría ser muy utilizada en educación especial, aunque no cuenta actualmente con las adaptaciones necesarias para la utilización por parte de personas con problemas motores. De las opiniones dadas, se rescata la de uno de los expertos que expresó que para la utilización de JClíc, con alumnos con las características descriptas, se hace imprescindible integrar alguna rampa digital (Assistive Technology) como el Kanghoooru, ScreenScanner o el Clic-N-Type para introducir textos.

RECONOCIMIENTO DE VOZ

En esta sección las respuestas fueron más variadas. En la primera pregunta que se refería a si consideraban una buena alternativa la utilización de comandos por voz para la adaptación, cinco de los seis encuestados estuvieron de acuerdo en considerar a los comandos por voz como una solución apta, por diferentes razones. Algunas de estas fueron porque se considera más natural para el usuario, o porque se le podría dar una utilización secundaria en la rehabilitación de personas con trastornos del habla o fundamentaron su opinión en que se podía ampliar el campo de usuarios finales. Sin embargo, uno de los encuestados expresa que, sin tener demasiado conocimiento del tema, si la persona conserva una motricidad mínima, resultará más conveniente el uso de sistemas mecánicos. Manifiesta que los que involucran sonidos requieren ambientes silenciosos. De todas maneras, expresa que lo óptimo sería que cada persona pueda elegir en cada momento la forma de interactuar con el ordenador que le resulte más conveniente.

En cuanto al conocimiento de Sphinx-4, sólo la mitad de los expertos conocían la herramienta. De los cuales, dos pudieron responder si la elección de la herramienta era acertada. Ambos coincidieron en que la elección era correcta, con diferente fundamentación. Uno de ellos consideró la ventaja de que Sphinx-4 es un producto desarrollado en Java y distribuido bajo licencia de software libre, como JClíc, pero expresó su falta de conocimiento en cuanto a prestaciones del motor. El otro fundamentó que la elección era acertada, ya que Sphinx-4 es un software de alta calidad, y cuenta con modelos acústicos para el español.

INTEGRACIÓN DE COMANDOS POR VOZ Y JCLIC

Por último, en cuanto a la integración propuesta, las respuestas resultaron aún más variadas.

En cuanto a la pregunta de qué grupos se verán más beneficiados, hubo varias propuestas, entre las que se pueden nombrar: personas con discapacidad motriz severa que no presenten afectado el área del lenguaje, personas con discapacidad intelectual, personas de la tercera edad y niños; y por último, alumnos con agenesia de miembros superiores o parte de ellos.

Por otro lado, en cuanto a la ventaja de la utilización de comandos por voz, en resumen, los diferentes aportes concluyeron que probablemente es más sencillo y no requiere equipamiento sofisticado. Además, algunos manifestaron que puede resultar más eficiente dependiendo de las deficiencias y capacidades de los alumnos que lo utilizarán. Podría ayudar a los alumnos a dejar en un segundo plano su discapacidad y, resolver las actividades utilizando sus habilidades.

En cuanto a las cosas que se le agregarían a la solución propuesta, la mayoría consideró que la elección de las etiquetas y los conectores debiera ser configurable por parte del docente, para no tomar ninguna convención de antemano. Además, se planteó que sería interesante que desde las opciones generales de JClíc se pueda configurar si se desea o no mostrar la opción de reconocimiento de voz, para que, en el caso que se elija por no, directamente inicie la actividad. Otro aporte fue que se pudiera responder a la opción que pregunta acerca del uso de comandos por voz, también mediante el micrófono, sin requerir de la asistencia del docente.

Respecto a la utilización de etiquetas, en general se considera una elección acertada. Sin embargo, uno de los expertos considera que depende de la situación y que lo ideal sería que también sea configurable.

Para finalizar la encuesta, se preguntó qué otras cosas le agregaría a JClíc para mejorar la interfaz para el grupo de personas mencionado, a lo cual algunos respondieron que sería interesante, la opción de cursor automático que ya incorporaba Clic 3 o un sistema de barrido controlado con monosílabos o palabras cortas.

A continuación se presentan algunas conclusiones acerca de esta evaluación realizada sobre la propuesta de prototipo.

CONCLUSIONES ACERCA DE LA EVALUACION REALIZADA

Como conclusión de las encuestas realizadas a expertos, se considera que se ha realizado una buena elección del software educativo a adaptar, como así también, la utilización de comandos por voz. Como mencionó uno de los expertos, esta opción puede usarse de

forma complementaria con otras herramientas y no necesariamente es mejor o peor que otra adaptación, sino que es una alternativa diferente la cual abre un camino de nuevas posibilidades.

Si bien unos pocos expertos se manifestaron acerca de la elección del motor de reconocimiento de voz, coincidieron en que la misma es acertada. El aspecto fundamental a resaltar es su disponibilidad y sus posibilidades en cuanto a funcionalidad. En el marco de este trabajo, se considera que, la utilización de sphinx-4 resultó conveniente, rescatando también lo manifestado por los expertos.

Por otro lado, muchas de las propuestas realizadas por los consultados, en cuanto a la adaptación, resultaron muy interesantes y enriquecedoras. No se seguirá detallando aquí sino que será producto de un nuevo capítulo donde serán presentadas como líneas futuras.

CONCLUSIÓN

En este capítulo se han presentado las cuestiones principales para llevar a cabo el prototipo de adaptación. Se describieron las consideraciones que se tuvieron en cuenta para ambos componentes, y que concluyeron en decisiones de diseño e implementación imprescindibles para el prototipo.

Se analizaron detalles de la integración y se explicó cómo se la realizó. Particularmente, se describieron qué modificaciones se necesitaron para permitir que el reconocedor de voz y JClic funcionaran juntos para llevar a cabo el objetivo.

Luego, se presentó un ejemplo de ejecución de una actividad utilizando el prototipo para demostrar su funcionamiento final.

En la siguiente sección, se analizó la posibilidad de extender el presente trabajo a todas las actividades provistas por JClic, continuando el capítulo con una lista de pautas a tener en cuenta al momento de iniciar una investigación similar a la presentada en este trabajo.

Por último, se presentaron las razones por las cuales se escogió la prueba de juicio de expertos, como una opción válida, para evaluar las decisiones tomadas durante el presente trabajo. Así como también se detallaron alguno de los resultados recogidos a partir de la aplicación de esta técnica.

En el próximo capítulo, se presentarán las conclusiones de este trabajo, junto con las líneas futuras de investigación y trabajo propuestas, que se desprenden del análisis y desarrollos realizados.

CAPÍTULO 10

CONCLUSIONES Y LÍNEAS DE INVESTIGACIÓN FUTURAS

CONCLUSIONES DE LA TESINA

INTRODUCCION

Como se dijo en capítulos anteriores, las personas con un alto compromiso motriz tienen poca autonomía para la realización de tareas, necesitando en la mayoría de los casos ayuda de terceros, para cambiar de tareas, acceder a información, desplazarse, etc. En la computadora, este problema también se manifiesta, pues suele ser el profesional quien ayuda constantemente a la persona a realizar las tareas.

Un alto nivel de autonomía, haría sentir a estos alumnos, independientes, podría aumentar su autoestima y su autosuficiencia. Esto, influirá considerablemente en su comportamiento, facilitando que puedan ser emprendedores y que puedan realizar actividades nuevas por iniciativa propia.

Las computadoras poseen varias características y capacidades que pueden ser muy bien aprovechadas para trabajar con personas con necesidades especiales. Entre otras cosas, permiten utilizar una gran variedad de recursos como imagen y sonido, cubriendo múltiples necesidades con programas que pueden ser modificados para adaptarse a las necesidades de cada persona.

Algunos de los aspectos que se pueden abordar utilizando programas de computación son: el enfoque de la atención, el estímulo visual, la disminución de la sensación de aislamiento, el entretenimiento, el control del ambiente, entre otras.

Por lo tanto, estas razones motivaron el desarrollo de adaptaciones a un software muy utilizado en el ámbito educativo como es JClic, para facilitar su uso por parte de alumnos con dificultades motoras y posibilidades de comunicación oral, y así incentivar su desarrollo intelectual.

Con la adaptación propuesta, se pretende que el alumno pueda resolver las actividades de JClic de manera independiente, sin necesitar la ayuda del docente durante toda la resolución. De esta manera, obtendrá un cierto nivel de autonomía y, además, facilitará el trabajo del docente en actividades colectivas.

Esta adaptación mantiene las características de licencia GNU y OpenSource, por lo que el código estará disponible para quien desee descargarlo y continuar/modificar el proyecto. Los archivos fuente serán enviados a los creadores de JClic, y cualquier persona interesada puede solicitarlo a cualquiera de las integrantes de esta tesina.

EXPERIENCIA ADQUIRIDA

A lo largo del desarrollo de la tesina, se han estudiado diversos temas relacionados a las TIC en educación, los cuales se fueron desarrollando en diferentes capítulos de este trabajo.

La idea de la adaptación surgió de la necesidad expresada por parte de profesionales en el ámbito de educación especial, vinculados a una de las autoras de la tesina.

A partir de allí, se inició una aproximación al tema para obtener más información, y por ello, se consultó a expertos en el área de TIC aplicada a la educación especial. Como conclusión, las adaptaciones son muy necesarias y resultan complementarias, dada la cantidad de necesidades y particularidades en el área. En resumen, se remarcó la necesidad de crear herramientas para personas con discapacidad motriz, potenciando las posibilidades del habla en aquellos que no tuvieran consecuencias mayores en esta área.

Antes de iniciar con el desarrollo de la tesina, se comenzó a investigar cuáles eran los programas educativos disponibles, revisando las adaptaciones que presentaba cada uno de ellos. Posterior al estudio de las herramientas, se seleccionó JClick para la implementación de la adaptación. También, se estudiaron diferentes rampas digitales, y se pusieron a prueba algunas de ellas junto con JClick, para analizar su funcionamiento.

Es importante destacar, que una de las integrantes del presente trabajo realizó una pasantía en la Escuela Especial 502 de Berisso, y pudo observar las dificultades con las que se encontraban los docentes de dicho establecimiento para incorporar el uso de computadoras en el trabajo áulico. Además, allí se realizaron experiencias con la herramienta JClick, en colaboración con el personal educativo de la escuela.

Paralelamente a estos estudios, se abordó el tema de la discapacidad motriz, y lo que ella representa para las personas que la poseen, tanto a nivel cognitivo como social. Sumado a esto, se investigaron las diferentes causas y consecuencias en el desarrollo de la persona que la padece.

Esta información fue recolectada con el objetivo de entender y comprender las necesidades reales de las personas con dichas características.

Se pensó en el reconocimiento de voz como una buena alternativa para implementar la adaptación deseada de JClick. Se continuó entonces, con el estudio y análisis de diferentes sistemas reconocedores. Se realizaron consultas también a diferentes usuarios de estos sistemas para tomar contacto con sus bondades y dificultades. Luego, se profundizó en los aspectos referidos al reconocimiento de voz, de manera de poder contar con los conocimientos necesarios para lograr introducir esta tecnología, y llevar a cabo el desarrollo del prototipo.

Obtenida la base teórica, se pusieron en práctica estos conocimientos a través del uso y estudio de diferentes motores de RV. Se eligió y se profundizó el estudio de Sphinx 4. En particular, se tuvo que tomar conocimiento acerca de la estructura del motor, su forma de trabajo, y los modelos disponibles para el español. Así como también, aspectos vinculados a su configuración.

Luego de elegidas las herramientas a utilizar, se realizó un estudio del lenguaje Java, en el que están desarrolladas ambas. Se aprendieron otros conceptos particulares asociados a este lenguaje, como el uso de SwingWorkers y manejo de threads.

Estos, fueron algunos de los conocimientos adquiridos, previos a la integración. Este estudio, fue fundamental para el desarrollo del presente prototipo.

Para llevar adelante dicha integración, se realizó el código pertinente que fue producto de todo el camino recorrido previamente.

Como último paso, se realizaron encuestas a expertos para que expresen su opinión sobre el producto desarrollado. Se cree que los resultados de las encuestas, en general, fueron muy positivos. A partir del análisis de las encuestas, se considera que se ha realizado una buena elección del software educativo a adaptar, como así también, la utilización de comandos por voz. Como mencionó uno de los expertos, esta opción puede usarse de forma complementaria con otras herramientas, y no necesariamente es mejor o peor que otra adaptación, sino que es una alternativa diferente la cual abre un camino de nuevas posibilidades. También despertó el interés de uno de los desarrolladores de JClic, con el que se ha tomado contacto.

Se considera que la elección del motor de reconocimiento de voz, también fue una decisión acertada, dentro de las alternativas estudiadas y los objetivos buscados. El aspecto fundamental a resaltar es su disponibilidad, su licencia y sus posibilidades en cuanto a funcionalidad.

Si bien se ha obtenido un amplio conocimiento de diferentes herramientas, tanto educativas como relacionadas con RV, quedan ciertas modificaciones, mejoras y extensiones a desarrollar en la adaptación presentada. En la siguiente sección, se plantean algunas consideraciones a tener en cuenta como líneas de trabajo futuras. Varias de ellas fueron propuestas por los expertos encuestados.

LÍNEAS FUTURAS

A partir de lo realizado hasta ahora, pueden proponerse algunas líneas futuras de trabajo, de las cuales algunas son mejoras al prototipo propuesto y otras se refieren a ampliar conocimientos y desarrollos.

MEJORAS AL PROTOTIPO PROPUESTO

En cuanto al desarrollo propuesto, se plantean algunas mejoras relacionadas con la interfaz y la utilización de la aplicación.

Por un lado, se propone mejorar el aspecto con el que se muestra la etiqueta dentro de cada celda. Sería una buena alternativa que el formato de la misma sea configurable, para

que el docente pueda seleccionar el color (para lograr el contraste que se desee con la imagen/color de fondo), además del tamaño de la letra y otras opciones de formatos. Esta tarea implicaría cambiar la interfaz de JClcAuthor, agregando la funcionalidad necesaria para configurar los aspectos visuales de la etiqueta.

Otro punto a tener en cuenta, que ha sido recomendado por uno de los expertos intervinientes en el análisis de la propuesta, es comenzar la numeración de las celdas desde 1 y no desde 0, ya que considera que, el 0 es un número muy utilizado en la informática pero no es así en aplicaciones con fines educativos.

Por último, se propone extender el prototipo para que puedan resolverse las actividades con un mayor número de casilleros. En el caso de la asociación simple, se pueden crear actividades que contengan hasta 30 casilleros, por lo que habría que extender el diccionario y la gramática para pronunciar 30 números.

Estas son algunas de las mejoras que podrían realizarse sobre el prototipo propuesto. En secciones posteriores se proponen más modificaciones.

PRUEBA CON CASOS CONCRETOS

En el capítulo 9, se mostró que se realizó una evaluación a través del juicio de expertos de la tarea realizada hasta el momento, allí mismo se explicó en qué consistía este método y los resultados obtenidos. Si bien el juicio de expertos es un método de evaluación muy utilizado, sería conveniente realizar pruebas con usuarios finales, luego de realizadas las mejoras antes mencionadas y teniendo una versión más avanzada del producto.

MODIFICAR JCLICAUTHOR PARA LA ELECCIÓN DE UTILIZAR VOZ

Por otro lado, actualmente, siempre que se carga un proyecto desde el JClcPlayer se consulta por medio de una ventana modal, si se desea realizar la actividad con comandos por voz. Según la opinión de expertos, esta decisión no tendría consecuencias negativas, pero se propuso que sea el docente quien configure cuando genera la actividad desde al JClcAuthor, si se debe realizar la consulta o no. Dado que JClc no sólo se usa en el ámbito de educación especial, muchas veces no será necesario utilizarlo con ninguno de los alumnos, y en otras circunstancias, será una decisión a nivel de cada alumno.

CONFIGURAR ETIQUETAS

En lo que respecta a la elección de las etiquetas, la mayoría de las opiniones de expertos sugieren tener etiquetas más variadas, de manera tal de adecuarlas a la circunstancia particular. Por lo que, sería conveniente que en un futuro se pueda configurar, por parte del docente, las etiquetas a utilizar. Se propone la utilización de letras o palabras como alternativas, de acuerdo también a la estrategia didáctica del docente. Para la

implementación de esta modificación, se debería encontrar la manera de realizar una generación automática del diccionario y la gramática, ya que estos archivos no podrían contener todas las alternativas posibles. Es necesario, entonces, que el docente disponga de una herramienta que permita, de manera transparente, adecuar el diccionario y la gramática acorde a las circunstancias.

EXTENDER LA IMPLEMENTACION A TODAS LAS ACTIVIDADES

En el capítulo 9, se plantea una forma de extender la solución a todas las actividades de JClic. Como línea futura se propone implementar los casos que han sido planteados.

Se debe considerar aumentar el diccionario a más de 30 etiquetas considerando además las necesarias para resolver las actividades de texto.

COMO CIERRE DE ESTE TRABAJO

Esta tesina ha permitido que nos vinculáramos con temas interesantes y de impacto para nuestra formación personal y profesional.

En particular, hemos aprovechado desde lo estrictamente vinculado con la disciplina informática, así como también, aquellas cuestiones cercanas a la educación, en particular, la educación especial.

Creemos que este trabajo, puede servir de base para aquellos relacionados a estas temáticas, y sabemos que aún queda mucho camino por recorrer en este sentido.

ANEXO A

ENCUESTAS

INTRODUCCIÓN

En el presente anexo se transcriben las encuestas realizadas a Antonio Sacco, Francesc Busquets, Jaquelina Rosica, Rafael Sánchez Montoya, Ivan Mezza y Rosalía Moralejo, vinculados todos ellos, de alguna manera, a los temas abordados en esta tesina. En cada una de las secciones se hará una breve introducción del área en que se especializa cada uno de los expertos mencionados.

ENCUESTADOS

En las siguientes subsecciones se hace un resumen de la experiencia de cada uno de los encuestados, en el área en que se especializan

ANTONIO SACCO

Graduado en Ingeniería en Sistemas de Información, en la Universidad Tecnológica Nacional. Especialista en Informática Aplicada en la Educación. Actualmente se encuentra finalizando Master Tecnología Informática Aplicada en la Educación en la UNLP. Antonio ha realizado varios aportes en el área de tecnología aplicada a la educación, a nivel internacional. Algunos de ellos han sido mencionados en capítulo anteriores, como por ejemplo, Emuclic, MPB (Mouse Por Barrido), entre otros. En <http://www.antoniosacco.com.ar/> puede encontrarse más información.

FRANCESC BUSQUETS

Profesor en el CEIP "Pompeu Fabra" de Barcelona. Actualmente trabaja como asesor técnico docente en el Departamento de Educación de la Generalitat de Cataluña, coordinando el área de software libre y desarrollo de aplicaciones. Es autor del software educativo "Músic", "Clic" y "JClic", entre otros. Ha participado también en el desarrollo de la distribución de GNU/Linux "Linkat", así como en la creación de diversos interactivos multimedia. Es impulsor del proyecto Clic (<http://clic.xtec.net>), un espacio dedicado a la cooperación entre escuelas y educadores de diversos países mediante el desarrollo e intercambio de aplicaciones y contenidos libres.

JAQUELINA ROSICA

Egresada de la Facultad de Humanidades y Ciencias de la Educación, en el área de Educación Física. Ha trabajado en el área de educación especial, desde ya hace varios años. Actualmente, es docente de la Escuela para niños con problemas motrices María Montessori de la ciudad de La Plata.

RAFAEL SÁNCHEZ MONTOYA

Doctor en Métodos de Investigación e Innovación Educativa, catedrático de Educación Secundaria, profesor de la Universidad de Cádiz (EUEJE Campus Bahía de Algeciras). Representante del Ministerio de Educación (España) en el proyecto SEN-IST-NET (Red Europea de Excelencia sobre necesidades educativas especiales y Tecnologías de la Sociedad de la Información), de la Agencia Europea para el Desarrollo de la Educación Especial (Bruselas).

Secretario general de la Fundación RedEspecial (Fundación Iberoamericana para la Cooperación en Educación Especial y Tecnología Adaptativa) y miembro fundador de la Red VIT@LIS, para la colaboración entre Europa, América Latina y el Caribe en temas de Sociedad de la Información (Bruselas).

Autor del libro “Ordenador y Discapacidad”, el cual utilizamos en el presente trabajo.

IVÁN MEZZA

Investigador en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), Universidad Nacional Autónoma de México. Integrante del proyecto DIME (Diálogos Inteligentes Multimodales en Español), el cual fue utilizado en el presente desarrollo.

ROSALIA MORALEJO

Licenciada en Fonoaudiología, graduada en la Universidad Nacional de San Luis. Actualmente se encuentra realizando el Doctorando en Fonoaudiología. Ha trabajado con niños con discapacidad motora, especialmente niños con Parálisis Cerebral.

ENCUESTAS

A continuación se muestra el contenido de las encuestas realizadas a los expertos:

1. ¿Conoce la herramienta JClíc? Marque con una cruz la respuesta que corresponda.

Si:

No:

En caso de haber contestado que Si, responda a las siguientes preguntas:

2. ¿Considera que JClíc es una herramienta utilizada en el ámbito educativo? ¿Por qué?
3. ¿Cree que puede ser utilizada además en el contexto de la educación especial? ¿Conoce situaciones de uso? Mencione alguna/s.
4. ¿Qué dificultades encuentra en la utilización de JClíc, en particular, para los chicos con Discapacidad motora?

RECONOCIMIENTO DE VOZ

1. ¿Cree que es una buena alternativa utilizar comandos por voz para personas con discapacidad motora que no presenten dificultades o dificultades mínimas en el habla? ¿Por qué?

2. ¿Conoce el motor Sphinx? Marque con una cruz la respuesta que corresponda.

Si:

No:

3. En caso de haber respondido SI en la pregunta anterior, responda la siguiente ¿Considera acertada la elección de Sphinx como motor de reconocimiento por voz? ¿Por qué?
4. ¿Qué otros motores de reconocimiento de voz conoce? En caso de tener experiencia de uso o conocimiento sobre ellos, mencione las que cree son sus principales características y ventajas.

INTEGRACIÓN DE COMANDOS POR VOZ Y JCLIC

1. ¿Qué grupos cree que se verán más favorecidos con la adaptación propuesta?

ANEXO B

DEFINICIONES

Alófonos: Un alófono es una realización posible de un fonema si pertenece a la clase de equivalencia de sonidos asociada al fonema. Por ejemplo, la palabra <casa>, por ejemplo, consta de cuatro fonemas (/k/, /a/, /s/, /a/). A esta misma palabra también corresponden en el habla, acto concreto, cuatro sonidos, a los que la fonología denominará alófonos, y estos últimos pueden variar según el sujeto que lo pronuncie. La distinción fundamental de los conceptos fonema y alófono, está en que el primero es una huella psíquica de la neutralización de los segundos que se efectúan en el habla.

Aplicación X: Una aplicación X es básicamente una colección de widgets, que están unidos entre sí de diversas maneras. Para ejecutar aplicaciones X en Linux y UNIX, se necesita un sistema X Windows. Se puede encontrar más información en <http://publib.boulder.ibm.com/infocenter/idm/v2r1/index.jsp?topic=/com.ibm.db2tools.fpeic.doc.ug/fpeic209a.html>

BSD Licencia: La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva. La licencia BSD permite el uso del código fuente en software no libre.

Cepstrum: El cepstrum de una señal es el resultado de calcular la transformada de Fourier del espectro de la señal estudiada en escala logarítmica (dB). El nombre cepstrum deriva de invertir las cuatro primeras letras de spectrum.

Decibelio: es la unidad relativa empleada en acústica y telecomunicaciones para expresar la relación entre dos magnitudes, acústicas o eléctricas, o entre la magnitud que se estudia y una magnitud de referencia.

El decibelio, cuyo símbolo es dB, es una unidad logarítmica. Es un submúltiplo del belio, de símbolo B, que es el logaritmo de la relación entre la magnitud de interés y la de referencia, pero no se utiliza por ser demasiado grande en la práctica, y por eso se utiliza el decibelio, la décima parte de un belio. El belio recibió este nombre en honor de Alexander Graham Bell.

Un belio equivale a 10 decibelios y representa un aumento de potencia de 10 veces sobre la magnitud de referencia. Cero belios es el valor de la magnitud de referencia. Así, dos belios representan un aumento de cien veces en la potencia, 3 belios equivalen a un aumento de mil veces y así sucesivamente. Particularmente 20 db indica que el nivel de intensidad del sonido corresponde al de "Biblioteca".

Fricación: Se denomina fricación a la articulación de un sonido fricativo: en la palabra "faz" se producen dos fricaciones, la de la "f" y la de la "z". Se mencionan las letras fricativas en el capítulo 4.

GCC: GNU Compiler Collection. Es un conjunto de compiladores creados por el proyecto GNU - <http://gcc.gnu.org/>

HTK (Hidden Markov Model Toolkit): The Hidden Markov Model Toolkit (HTK) es un conjunto de herramientas portables para la construcción y manipulación de modelos ocultos de Markov. HTK se utiliza principalmente para la investigación de reconocimiento de voz, aunque se ha utilizado en numerosas aplicaciones, como la investigación en síntesis de voz, reconocimiento de caracteres y la secuenciación de ADN. Consta de un conjunto de bibliotecas y herramientas disponibles en forma de código fuente, programada en C. Las herramientas proporcionan instalaciones sofisticadas para el análisis de voz, entrenamiento HMM, pruebas (test) y análisis de resultados. El software soporta HMM y se puede utilizar para construir sistemas de HMM complejo. La versión liberada de HTK contiene una amplia documentación y ejemplos.

Licencia Pública General de GNU (GPL): es una licencia creada por la Free Software Foundation, en 1989, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre, y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

La licencia GPL, al ser un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se la denomina contrato de licencia o acuerdo de licencia.

RPM Package Manager: es una herramienta de administración de paquetes pensada básicamente para Linux, capaz de instalar, actualizar, desinstalar, verificar y solicitar programas.

Shareware: modalidad de distribución de software, en la que el usuario puede evaluar de forma gratuita el producto, pero con limitaciones en el tiempo de uso o en algunas de las formas de uso o con restricciones en las capacidades finales.

Símbolos Pictográficos para la Comunicación (SPC): se componen principalmente de dibujos simples. La palabra que simboliza cada dibujo está impresa encima del mismo. Algunas palabras no están dibujadas a causa de su significado abstracto y por lo tanto están simplemente impresas. Igualmente se incluyen el alfabeto, los números y espacios para colores.

Simple DirectMedia Layer (SDL): es un conjunto de bibliotecas desarrolladas con el lenguaje C que proporcionan funciones para realizar operaciones de dibujado 2D, gestión de efectos de sonido y música, y carga y gestión de imágenes. (<http://www.libsdl.org/>)

Sistema Bliss: el método de Charles Bliss, es un sistema gráfico. Los símbolos Bliss son de una gran simplicidad, y no es necesario saber leer para usarlos. En la pre-lectura, el Bliss puede usarse para identificar objetos sencillos y expresar ideas y sentimientos.

SMIL: es el acrónimo de Synchronized Multimedia Integration Language (lenguaje de integración multimedia sincronizada) y es un estándar del World Wide Web Consortium (W3C) para presentaciones multimedia. El lenguaje SMIL permite integrar audio, video, imágenes, texto o cualquier otro contenido multimedia. (<http://es.wikipedia.org/wiki/SMIL>)

W3C (World Wide Web Consortium): es un consorcio internacional que produce recomendaciones para la World Wide Web. Está dirigida por Tim Berners-Lee. También participa en la educación, desarrollo de software y funciona como foro abierto para debatir sobre la Web.

REFERENCIAS BIBLIOGRÁFICAS

Access Total – 2010 - <http://www.accesstotal.com.ar/definicion-accesibilidad.asp>

Adaptabilidad de interfaces de usuario por reflexión – 2010 - <http://www.info-ab.uclm.es/personal/AntonioFdez/download/papers/conference/DOLMEN2002-reflexion.pdf>

Aenor - 2010 - <http://www.cettico.fi.upm.es/aenor/software.htm>

Ardora – 2010 - http://webardora.net/index_cas.htm

Arquer – 2010 – Fiabilidad humana: métodos de cuantificación, juicio de expertos
http://www.insht.es/InshtWeb/Contenidos/Documentacion/FichasTecnicas/NTP/Ficheros/401a500/ntp_401.pdf

Ayudas Técnicas, habilitación y diversidad – 2010 -
http://www.care.org.ar/archivos/Tecnologia_de_ayuda.pdf

Bernal Bermúdez, Bobadilla Sancho, Gómez Vilda - Reconocimiento de voz y fonética acústica. México, Alfaomega grupo Editor, 2000

Capítulo 3, Reconocedores de voz – 2010 -
http://catarina.udlap.mx/u_dl_a/tales//documentos/lis/marquez_a_bm/capitulo_3.html

Castellano, Sacco, Zurueta – 2009 - Paper sobre discapacidad de información general:
<http://www.niee.ufrgs.br/eventos/CIIEE/2003/bloque2/comunicaciones/La%20utilizacion%20de%20Software%20de%20uso%20general%20y%20Aplicaciones%20esp.doc>

Construcción de Switches e interfaces - 2010 -
http://www.antoniosacco.com.ar/docu/construccion_de_switches_e_interfaces.pdf

Coparticipando en el conocimiento – 2010 - <http://dialnet.unirioja.es/servlet/oaiart?codigo=126248>

Deporte Adaptado 2010 -
http://www.deportes.uba.ar/deporte_social/dep_adapt/pdf/discapacidad_motriz.pdf

Discapacidad -2010 - <http://www.artrosisisalud.com.ar/>

Discapacidad Motriz - 2010 -
http://www.seq.gob.mx/portal/niveles_educativos/basica/educ_especial_motriz.php

Dominio de la frecuencia – 2010 - http://es.wikipedia.org/wiki/Dominio_de_la_frecuencia

Educacion Especial – 2009 - Descarga de diversos programas para educación especial
<http://software.computadora-discapacidad.org/>

EmuClic – 2010 - <http://www.antoniosacco.com.ar/emuclit.htm>

Entrenamiento usando SphinxTrain – 2010 - <http://www.bakuzen.com/?p=16>

Experto en tics y discapacidad - Material entregado a alumnos en el curso experto en Tics y discapacidad. Montevideo, 2009

Hot Potatoes – 2010 - <http://www.cict.co.uk/software/hotpotatoes.net/>

Introducción a las Pautas de Accesibilidad al Contenido en la Web – 2010 -
<http://www.w3c.es/Traducciones/es/WAI/intro/wcag>

Introducción a las Pautas de Accesibilidad para Agentes de Usuario – 2010 -
<http://www.w3c.es/Traducciones/es/WAI/intro/uaag>

Introducción a las Pautas de Accesibilidad para Herramientas de Autor – 2010 -
<http://www.w3c.es/Traducciones/es/WAI/intro/atag>

Java Speech API – 2010 - <http://java.sun.com/products/java-media/speech/>

JClic, sitio oficial – 2010 - <http://clic.xtec.cat/es/jclic/index.htm>

Kanghoooru – 2009 - http://espanol.softpicks.net/software/Kanghoooru_es-30052.htm

Kirriemuir, Speech Recognition Technologies – 2010 - http://www.jisc.ac.uk/uploaded_documents/tsw_03-03.pdf

Language Model – 2010 - http://sphinx.subwiki.com/sphinx/index.php/Language_model

Lim – 2010 - <http://www.educalim.com/cinico.htm>

Listado de aplicaciones basadas en Sphinx4 – 2010 - <http://cmusphinx.sourceforge.net/wordpress/applications/>

Loquendo, sitio oficial – 2010 - <http://www.loquendo.com/es/technology/speechsuite.htm>

Luque Parra, Rodríguez - <http://www.rieoei.org/2806.htm> - *Tecnología de la Información y Comunicación aplicada al alumnado con discapacidad: un acercamiento docente, Málaga, 2009*

Luque, Rodríguez y Romero, 2005 - *Intervención psicosocial: Revista sobre igualdad y calidad de vida, ISSN 1132-0559, Vol. 14, Nº 2, 2005, Págs. 209-222*

Markin – 2010 - <http://www.cict.co.uk/software/markin/index.htm>

Martínez Bernaldo de Quirós, Fundamentos básicos del reconocimiento de voz – 2010 - <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=complementosr>

Martínez, Portale, Klein, Olmos, Reconocimiento de voz – 2009 - http://www.secyt.frba.utn.edu.ar/gia/IA1_IntroReconocimientoVoz.pdf

Mouse por Barrido – 2010 - <http://www.antoniosacco.com.ar/mpb.htm>

NICO Toolkit, sitio oficial – 2010 - <http://nico.nikkostrom.com/>

Normas de Accesibilidad a la Informática – 2010 - <http://www.cettico.fi.upm.es/aenor/internet.htm>

Nuevas Tecnologías y educación de personas con dificultades – 2010 - <http://www.tecnoneet.org/docs/2002/2-82002.pdf>

Plaphoons – 2010 - <http://www.tecnoneet.org/docs/2002/4-22002.pdf>

Propuestas educativas ante el alumno con discapacidad motora - 2010 - http://www.csi-csif.es/andalucia/modules/mod_ense/revista/pdf/Numero_14/MARIA%20JOSE_HURTADO_2.pdf

Pulsadores, conmutadores e interruptores. Sistemas de Acceso para el discapacitado motórico – 2010 - <http://www.tecnoneet.org/docs/2002/4-102002.pdf>

Ratón Facial – 2010 - <http://www.crea-si.com/esp/rfacial.php>

Reconocimiento de voz y Educación – 2010 - http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/aguas_g_n/capitulo1.pdf

Sánchez Montoya – Ordenador y Discapacidad. Guía práctica a las personas con necesidades educativas especiales. Madrid, 2002

Sancho - Para una Tecnología Educativa. “La tecnología: un modo de transformar el mundo cargado de ambivalencias”, Barcelona: HORSORI, 1994

Sancho - Para una Tecnología Educativa. “La tecnología: un modo de transformar el mundo cargado de ambivalencias”, Barcelona: HORSORI, 1998

Speech Recognition HOWTO – 2010 - <http://www.tldp.org/HOWTO/Speech-Recognition-HOWTO/introduction.html#BASICS>

Speech Synthesis & Speech Recognition Using SAPI 5.1 – 2010 - <http://www.blong.com/Conferences/DCon2002/Speech/SAPI51/SAPI51.htm>

Sphinx, modelos acústicos – 2010 - http://cmusphinx.sourceforge.net/sphinx4/#acoustic_models

Sphinx, sitio oficial - 2010 - <http://cmusphinx.sourceforge.net/wordpress/>

Tecnología de la Información y Comunicación aplicada al alumnado con discapacidad – 2010 - <http://www.rieoei.org/deloslectores/2806Parrav2.pdf>

Tecnologías Adaptadas – 2010 - <http://tecnoadaptada.blogspot.com/>

Tecnologías para el acceso al ordenador – 2010 - <http://ocw.um.es/cc.-sociales/tecnologias-de-apoyo-y-atencion-a-la-diversidad/material-de-clase-1/tema2.pdf>

TextToys – 2010 - <http://www.cict.co.uk/software/textoys/index.htm>

Un juego de gravedad con reconocimiento de voz para niños con problemas de leguaje – 2010 - http://www.clihc.org/2007/papers/JuegoGravedad_ID3_shortpaper.pdf

Utilización del programa Plaphoons – 2010 - <http://tecnologiayne.wikispaces.com/tema8>

Wikipedia Accesibilidad - 2010 - <http://es.wikipedia.org/wiki/Accesibilidad>

Wikipedia Accesibilidad Web- 2010 - http://es.wikipedia.org/wiki/Accesibilidad_web

Wikipedia Acoustic model – 2010 - http://en.wikipedia.org/wiki/Acoustic_Model

Wikipedia Ayuda Técnica – 2010 - http://es.wikipedia.org/wiki/Ayuda_técnica

Wikipedia Reconocimiento del habla – 2010 - http://es.wikipedia.org/wiki/Reconocimiento_del_habla

Wink – 2010 - <http://www.debugmode.com/wink/>

W3C Accesibilidad Web - 2010 - <http://www.w3c.es/Traducciones/es/WAI/intro/accessibility>

XVoice, sitio oficial – 2010 - <http://xvoice.sourceforge.net/>