

DISEÑO DE UN SOFTWARE PARA EL RECONOCIMIENTO DE SÍMBOLOS  
MATEMÁTICOS EN LATEX MEDIANTE SÍNTESIS DE VOZ PARA PERSONAS CON  
DISCAPACIDAD VISUAL

ÁNGELA MARÍA ANDRADE MORALES

1088290864

UNIVERSIDAD TECNOLÓGICA DE PEREIRA FACULTAD DE INGENIERÍAS  
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA 2013

DISEÑO DE UN SOFTWARE PARA EL RECONOCIMIENTO DE SÍMBOLOS  
MATEMÁTICOS EN LATEX MEDIANTE SÍNTESIS DE VOZ PARA PERSONAS CON  
DISCAPACIDAD VISUAL

ÁNGELA MARÍA ANDRADE MORALES

1088290864

Trabajo de grado

Como requisito para la obtención del título

De ingeniera de Sistemas y Computación

Asesor

Ingeniero Eléctrico

Saulo Torres Rengifo

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA 2013

Nota de Aceptación:

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

*Este trabajo está dedicado a mis padres,  
Gracias a su esfuerzo y apoyo incondicional  
Hoy puedo hacer de este sueño, una realidad.*

*Está dedicado al ing. Ángel Andrés Andrade.  
Hoy tienes 24 años, mañana tendrás más y siempre.  
Siempre serás mi pequeño "hermanito".*

*... Y a ti mi bella mamá,  
Más que mi razón,  
Eres todas mis razones...*

## **Agradecimientos**

Agradezco a todos los profesores, amigos y familiares por su apoyo, porque gracias a ellos hoy puedo terminar este proyecto, y he logrado una de mis metas.

Agradezco especialmente al profesor Saulo Torres, quien me oriento en todo el desarrollo. Gracias a sus consejos este proyecto adquirió forma y se pudo realizar.

En última instancia quiero agradecer a todas las personas que de una u otra forma influenciaron las ideas plasmadas en este proyecto.

## **PRIMERA PARTE: INTRODUCCIÓN A LA INVESTIGACIÓN**

<b>CAPITULO 1</b>	<b>1</b>
1.1 INTRODUCCIÓN	1
1.2 JUSTIFICACIÓN	2
1.3 OBJETIVOS	3
1.4 MARCO REFERENCIAL	4
1.4.1 MARCO CONCEPTUAL	4
1.4.2 MARCO DE ANTECEDENTES	5
1.4.3 MARCO TEÓRICO	6
1.5 DISEÑO METODOLÓGICO	9
1.6 METODOLOGÍA UTILIZADA	10

## **SEGUNDA PARTE: ESTADO DEL ARTE**

<b>CAPITULO 2</b>	<b>13</b>
2.1 ESTADO DE ARTE	13
2.2 HERRAMIENTAS DE SÍNTESIS DE VOZ PARA DOCUMENTOS CON ALTO CONTENIDO MATEMÁTICO.	13
2.2.1 SISTEMA DE ACCESO AUDIBLE A EXPRESIONES MATEMÁTICAS DIGITALES	14
2.2.2 SISTEMA DE AYUDA A LA ESCRITURA DE TEXTOS ESTRUCTURADOS PARA ESTUDIANTES CON DÉFICIT VISUAL	21
2.2.3 ACTEMAT (ACCESIBILIDAD A TEXTOS MATEMÁTICOS)	26
2.2.4 MAB (NAVEGADOR MATEMÁTICO DE AUDIO)	29
2.2.5 ASTER (SISTEMA DE AUDIO PARA LECTURAS TÉCNICAS)	31
2.2.6 PROSODY	34
<b>CAPITULO 3</b>	<b>37</b>
3.1 LECTORES DE PANTALLA (LSR)	37
3.1.1 ORCA	37
3.1.2 NVDA	38

## **TERCERA PARTE: DESARROLLO DE LA INVESTIGACIÓN**

### **CAPITULO 4 41**

---

4.1 L <sup>A</sup> T <sub>E</sub> X EN LA WEB	41
4.2 RECONOCIMIENTO DE SÍMBOLOS MATEMÁTICOS EN L <sup>A</sup> T <sub>E</sub> X	43
4.3 MODELO DE SOFTWARE	48

## **CUARTA PARTE: CONCLUSIONES**

### **CAPITULO 5 57**

---

5.1 CONCLUSIONES	57
5.2 BIBLIOGRAFIA	58

## **QUINTA PARTE: BIBLIOGRAFIAS Y ANEXOS**

### **CAPITULO 6 63**

---

6.1 ANEXOS	63
6.1.1 SPEECHSTREAM TOOLBAR	63
6.1.2 AUDIOMATH	63
6.1.3 WINTRIANGLE	64
6.1.4 MATHTRAX	64
6.1.5 EQNEDITOR	65
6.1.6 JAVA SPEECH MARKUP LANGUAGE (JSML)	65
6.1.7 SHAREL <sup>A</sup> T <sub>E</sub> X	66

## RESUMEN

El contenido de este proyecto hace referencia al diseño de un software para el reconocimiento de símbolos matemáticos en LaTeX, mediante el cual se pretenden establecer los modelos necesarios para la posterior implementación del software.

La primera parte del proyecto muestra el planteamiento general, en esta fase se incluye información como la introducción, los objetivos propuestos, y el diseño metodológico para la construcción del mismo.

La segunda parte del proyecto tiene como objetivo brindar al lector la información sobre el estado del arte de las herramientas existentes y en proceso de desarrollo para realizar la síntesis de voz de documentos con alto contenido matemático, para ello se ha recopilado una serie de información de diversas fuentes, y se ha elaborado un documento con la información más relevante.

En la tercera parte se realizó el desarrollo de la investigación, para ello se elaboró un documento bajo el estándar de la IEEE 830 para la especificación de requerimientos, en el cuál se establece el ámbito del sistema, la descripción detallada del proyecto, los requisitos funcionales, y otra información necesaria para alcanzar los objetivos del proyecto. Además de un modelo que integra los componentes principales para implementar el software.

En la cuarta parte se encuentran las conclusiones del proyecto, con las cuáles se pretende dar a conocer el resultado de la investigación y el cumplimiento de los objetivos.

La quinta parte contiene la bibliografía y los anexos del proyecto.



## CAPITULO 1

### 1.1 INTRODUCCIÓN

*Este proyecto ha sido diseñado de tal forma que el lector pueda tomarlo como base para desarrollar una aplicación que permita reconocer la estructura del lenguaje LaTeX, y de esta forma generar un metalenguaje que pueda ser leído por un sintetizador.*

Actualmente los sintetizadores de voz son tecnologías que permiten la interacción con herramientas de software a los discapacitados visuales y el acceso a documentos digitales. Sin embargo existen tareas aún poco cubiertas, como es el acceso a documentos con contenido matemático. Esto se debe principalmente a los siguientes factores:

Es un aspecto concreto de la accesibilidad que sólo afecta a parte de los discapacitados visuales que pudieran ser usuarios de ordenadores (sólo es relevante para aquellos que vayan a trabajar con texto matemático).

- En Colombia se ha descuidado este tema de tanta importancia, pues la ceguera afecta sobre todo a las personas con menos recursos económicos.
- Existen muchas limitantes para las personas con discapacidad visual en cuanto al uso de sintetizadores de voz, puesto que hay herramientas muy completas como el JAWS pero las licencias son muy costosas.
- Este es un problema social, puesto que hace que solo cierto grupo de personas puedan acceder al uso de las mismas.
- La lectura de la notación matemática es un proceso complejo, puesto que su traducción sonora no tiene siempre el mismo orden que la traducción gráfica, además que la interpretación de algunos de sus símbolos está relacionada con el contexto en el que esté siendo utilizada.

El objetivo de este proyecto aún en fase inicial, es obtener los modelos que representen la estructura del software, de manera tal que la implementación de este, se realice de manera óptima, y se facilite el desarrollo para el programador.

## 1.2 JUSTIFICACIÓN

El proyecto que se va a realizar, puede ayudar a todas aquellas personas que se encuentren en una situación de discapacidad visual, es decir que no solo sería de ayuda para los invidentes, puesto que mediante el estudio del estado del arte se determinará el estado actual de las herramientas disponibles para la síntesis de voz, y de los editores de texto para textos con notación matemática; el diseño del modelo puede servir a cualquier persona interesada en implementar el software.

El Ingeniero Saulo Torres (invidente), docente de la universidad tecnológica de Pereira nos manifestó la necesidad de hacer un proyecto de investigación y diseño de un software que integre el LaTeX y un sintetizador de voz que puede ser de ayuda para aquellas personas que deseen desarrollar una aplicación relacionada con el tema.

El proyecto se va enfocar en encontrar herramientas libres de síntesis de voz, y proponer un modelo que integre estas herramientas con en el LaTeX, de esta forma si alguna persona lo llegara a desarrollar, este proyecto podría ser útil para cualquier persona, y podría distribuirse sin costo alguno.

## **1.3 OBJETIVOS**

### **OBJETIVO GENERAL**

Realizar el diseño de un software que permita reconocer símbolos matemáticos en LaTeX mediante la síntesis de voz para personas con discapacidad visual.

### **OBJETIVOS ESPECÍFICOS**

- Desarrollar una metodología que permita alcanzar el objetivo principal, es decir la realización del modelo.
- Analizar el funcionamiento y las características, de los sintetizadores de voz existentes.
- Evaluar modelos similares actuales, para obtener un modelo solución.
- Realizar el estado del arte de las herramientas existentes que involucren la lectura de fórmulas matemáticas mediante la síntesis de voz.
- Realizar el levantamiento de requerimientos, bajo el estándar de la IEEE 830.
- Elaborar y evaluar el diseño del sistema.

## **1.4 MARCO REFERENCIAL**

En este apartado se encuentra la información sobre las principales definiciones y teorías sobre las cuales se fundamenta el proyecto.

### **1.4.1 MARCO CONCEPTUAL**

#### **UML**

Es un lenguaje que proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Este lenguaje nos indica cómo crear y leer los modelos

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

#### **Sintetizador de voz**

La síntesis de voz es la producción artificial de habla humana. Recibe también el nombre Text-To-Speech (TTS) en referencia a su capacidad de convertir texto escrito en hablado.

La tecnología de habla se implementa como Software, Hardware o una combinación de ambos. Cuando se implementan mediante Software, las aplicaciones no interactúan directamente con los componentes de audio de una computadora sino que son abstraídas del Hardware subyacente a través del motor de habla. Esto asegura que

las aplicaciones no estén atadas a implementaciones específicas de Hardware y sean portables a otras plataformas[6].

## **LaTeX**

LaTeX es un sistema de preparación de documentos. Con él puedes preparar manuscritos, artículos de revista, cartas, tesis, presentaciones y cualquier tipo de documento que quisieras imprimir en papel o mostrar en pantalla.<sup>1</sup>

## **Multiplataforma**

Cross-platform en inglés. Que tiene la capacidad de soportar múltiples plataformas.

Esto significa que el hardware o software que es multiplataforma tiene la característica de funcionar de forma similar en distintas plataformas (distintos sistemas operativos por ejemplo).

## **Metodología**

Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software.<sup>2</sup>

### **1.4.2 MARCO DE ANTECEDENTES**

Es importante resaltar que no existe un modelo, ni una aplicación que integre el LaTeX con algún sintetizador de voz.

Para la lectura de textos con gran contenido de fórmulas matemáticas, existen variados programas y software disponible entre ellos:

- MAVIS es un programa que incluye un proyecto de un lector de pantalla/navegador para matemáticas. El proyecto se encuentra detenido desde hace años.
- MathTalk es una herramienta capaz de reconocer y leer texto matemático a través de una interfaz sonora. Tanto el reconocimiento como la lectura se

---

<sup>1</sup>Guía rápida de LaTeX. Disponible desde: <http://navarroj.com/latex/guia.html>

<sup>2</sup>Software, Historia, Crisis, Metodología de Desarrollo. Disponible desde: <http://goo.gl/hunFOy>

efectúan basándose en la presentación y símbolo a símbolo. En ningún caso interpreta estructuras globales sino que lee símbolos individuales [5].

- JAWS, MultiWeb, MicroTalk ASAW, NVDA y otros lectores genéricos que no procesan por sí mismos el lenguaje matemático, dependen del sintetizador de voz sobre el que se apoyen. Estos normalmente tampoco leen adecuadamente el lenguaje matemático, si acaso se limitan a aplicar ciertas heurísticas para pronunciar expresiones aritméticas básicas [7].
- ACTEMAT permite el acceso sonoro a textos científicos empleando un lenguaje muy próximo a cómo se leería naturalmente. Hasta ahora sólo se podía acceder de forma deficiente, o bien escuchando directamente la codificación en lenguajes de representación de contenido (TeX, SGMLs, XMLs) no concebidos para ser leídos directamente por un humano [5].

### **1.4.3 MARCO TEÓRICO**

#### **Sintetizadores de voz**

La síntesis de voz es la producción artificial de habla humana. Recibe también el nombre Text-To-Speech (TTS) en referencia a su capacidad de convertir texto escrito en hablado.

La tecnología de habla se implementa como Software, Hardware o una combinación de ambos. Cuando se implementan mediante Software, las aplicaciones no interactúan directamente con los componentes de audio de una computadora sino que son abstraídas del Hardware subyacente a través del motor de habla. Esto asegura que las aplicaciones no estén atadas a implementaciones específicas de Hardware y sean portables a otras plataformas.

#### **Historia**

La ciencia ficción se está acercando a la realidad hoy en día, las tecnologías del habla se encuentran en pleno desarrollo e introducción en nuestro entorno cotidiano. El habla, medio de comunicación por excelencia entre seres humanos, comienza a ser utilizada como medio de comunicación con la tecnología desarrollada por el ser humano. La

capacidad de interactuar con las máquinas mediante el habla ha sido una constante en el desarrollo tecnológico de la humanidad.

A partir de finales del siglo XVIII, cuando Wolfgang Von Kempelen [KEM91], en 1791, construyó un rudimentario sintetizador de voz capaz de pronunciar frases cortas mediante un ingenio mecánico. En el siglo XIX se comenzaron a desarrollar las bases matemáticas en las que se basa el análisis de la señal de voz, como el análisis de Fourier. Durante este siglo se realizaron una serie de inventos que potenciaron la utilización del habla como método de comunicación. El teléfono permitió utilizar la voz para comunicaciones instantáneas y a larga distancia; el fonógrafo permitió grabar registros de voz, siendo lo que podríamos denominar el primer sistema de respuesta oral. Ya en el siglo XX, en el año 1939 se presentó una versión electrónica del sintetizador de Kempelen. El sistema denominado VODER [DUD39] consistía en un teclado muy elaborado que permitía controlar la articulación y generación de sonidos vocálicos y consonánticos. Este sistema pionero sembró las bases de los modernos sintetizadores de voz, basados actualmente en tecnología digital, pero compartiendo la misma teoría sobre el modelo de producción del habla. Ya desde los años 30 podemos decir que comenzó la investigación sobre la señal de voz con el desarrollo de sistemas prácticos de transmisión digital mediante modulación PCM. Con el advenimiento de las comunicaciones digitales y los sistemas de audio digital, se planteó el problema de la codificación eficiente de la señal de voz para reducir al máximo la velocidad de transmisión y el número de bits necesarios para la codificación binaria de la señal de voz. Por el contrario, hasta bien entrado el siglo XX no se tuvo un conocimiento suficiente sobre el comportamiento del sistema de percepción auditiva del ser humano. Sobre 1940 se desarrollaron los primeros experimentos para estudiar el comportamiento de la membrana basilar.

Un hito muy importante dentro de la investigación y desarrollo de las tecnologías del habla fue el invento del espectrógrafo de voz en la década de 1940. El espectrógrafo de voz permitía obtener un registro gráfico de la evolución de la energía de la señal de voz en diferentes bandas frecuenciales. Esta visión tiempo-frecuencia de la señal de voz conjuntamente con los estudios sobre el sistema auditivo humano abrió el campo a la investigación en una de las áreas más fascinantes de las tecnologías del habla: el reconocimiento automático del habla.

De hecho, en 1952 se presentó un sistema electrónico de reconocimiento de voz en los laboratorios Bell que era capaz de reconocer, dentro de un margen aceptable de error, los dígitos pronunciados en inglés.

La gran revolución en las tecnologías del habla se comenzó a gestar con el advenimiento de las computadoras digitales y la informática. Desde mediados de los años 60 se comenzaron a desarrollar algoritmos de procesamiento de señal, a desarrollar las bases de la inteligencia artificial y comenzaron a aparecer los primeros desarrollos e

investigaciones en sistemas de reconocimiento automático del habla utilizando computadoras digitales. Pero es en la década de los 70 cuando se da el impulso definitivo con la financiación de grandes proyectos de reconocimiento del habla por parte de países como EE.UU, Japón, Francia, Alemania, etc. En Noviembre de 1971 la Oficina de tecnología para el procesado de la información de la ARPA (Advanced Research Projects Agency of the USA Department of Defense) iniciaba un ambicioso proyecto de investigación con el objetivo de desarrollar varios sistemas de comprensión automática del habla continua para varios locutores cooperantes que hablasen "General American dialect" [LEA80]. Cinco años más tarde y una vez finalizado el proyecto inicial, el resultado final fue en cierta forma decepcionante ya que los objetivos no llegaron a alcanzarse pero si que sirvieron de toma de contacto con la realidad del problema y sentaron las bases para posteriores investigaciones. Una de las conclusiones finales, en las que se trataba de evaluar la capacidad computacional necesaria para abordar con garantías el funcionamiento en tiempo real de estos sistemas, situaba la capacidad mínima necesaria alrededor de los 100 MIPS (1MIP=1 Millón de Instrucciones por Segundo) [KLA77].

El requisito de disponer de una capacidad computacional de este orden con sistemas informáticos de bajo coste (Ordenadores Personales, Estaciones de Trabajo...) no se ha alcanzado hasta la actualidad, y aún de una forma parcial, gracias a la aparición en el mercado de forma masiva de la tercera generación de microprocesadores especializados en el procesado digital de señal (DSP-Digital Signal Processors) [LEE88]. Los avances en la tecnología de integración de circuitos integrados (VLSI), la aportación de nuevos conceptos en arquitectura de computadores (Arquitecturas RISC y Harward) junto a las decisiones de diseño para operar en un entorno de tiempo real incentivaron la aparición de DSPs de tercera generación.

Esta revolución tecnológica de la década de los 80 y principios de los 90, conjuntamente con la investigación básica realizada durante esta década por multitud de grupos de investigación en todo el mundo en el campo de las tecnologías del habla y de la inteligencia artificial han construido los cimientos del desarrollo actual en sistemas de comunicación oral hombre-máquina, encontrándonos en las puertas de una gran revolución en el desarrollo de sistemas basados en tecnologías del habla [5].

## **Historia de LaTeX**

Desilusionado con las pruebas de imprenta de uno de sus libros, el profesor Donald Knuth, de la universidad de Standford, decidió crear él mismo un sistema de composición de textos científicos de alta calidad. En este sistema, que llamaría TEX se puso a trabajar en mayo de 1977, alentado por la American Mathematical Society.



A comienzos de los años ochenta, Leslie Lamport comenzó a desarrollar un sistema de preparación de documentos basado en TEX. Se trataba de añadir un cierto nivel de abstracción a los comandos de TEX que permitiera a los autores concentrarse en la estructura del documento más que en los detalles de formato. Estos detalles pasaron a dejarse en manos de los diseñadores que proporcionaban los denominados ficheros de estilo. Este sistema recibió el nombre de LATEX (que puede escribirse también LaTeX).

Las funcionalidades de LATEX se vieron pronto completadas con unos cuantos programas auxiliares para la generación de índices, bibliografías, referencias cruzadas, tablas de contenidos e inclusión de gráficos, características que faltaban en TEX. A esto hay que añadir los cientos de paquetes de macros que usuarios de todo el mundo han ido creando en estos años y que proporcionan nuevas características y una gran flexibilidad para adaptarse a las más diversas necesidades.

El equipo LATEX3 dirigido por Frank Mittelbach produjo una nueva versión llamada LATEX2e, que ha sustituido al LATEX2.09 que apareció hace algunos años. En la actualidad se sigue mejorando para producir lo que será la versión 3.3

## **1.5 DISEÑO METODOLÓGICO**

### **HIPÓTESIS**

¿Es posible que realizando una investigación sobre el estado del arte de las herramientas que existen en estos momentos para el reconocimiento de símbolos matemáticos, mediante síntesis de voz, se pueda diseñar un modelo que integre las herramientas LaTeX y un sintetizador de voz?

### **VARIABLES**

Para la definición de las variables, se tuvieron en cuenta los criterios necesarios para la elaboración de los cuestionarios debido a que las encuestas deben estar enfocadas a medir la confiabilidad y la validez de los elementos del modelo, por ello se deben realizar preguntas que sean precisas, cuantificables e independientes. De las preguntas se desprenden las variables del proyecto.

---

<sup>3</sup>Breve historia de TEX y LATEX. [En línea] Disponible desde: <http://goo.gl/vSrC0D>

## ENCUESTA

Califique de 1 a 5, siendo 5 la calificación más alta y 1 la más baja, los siguientes criterios:

1. ¿El modelo integra con claridad y consistencia los elementos indispensables para la definición de los procesos y las relaciones entre ellos?

**Variable:** Nivel de Integración de los elementos del modelo

2. ¿El modelo es concordante con los requerimientos definidos para el desarrollo del mismo?

**Variable:** Nivel de relación entre los requerimientos especificados y el diseño del modelo.

3. ¿El modelo implementado es eficiente?

**Variable:** Nivel de Eficiencia.

4. ¿El modelo diseñado es de fácil comprensión para personas con conocimientos afines?

**Variable:** Nivel de Comprensión

Con los resultados obtenidos de la encuesta se procede, a realizar análisis estadísticos que permitan expresar de forma cuantitativa la confiabilidad y validez del modelo.

## 1.6 METODOLOGÍA UTILIZADA

El proyecto se ha dividido en cinco fases para su consecución, a continuación se describe el procedimiento llevado en cada fase:

1. En la primera fase se procede a realizar una recopilación de todo el material que pueda ser útil para construir el estado del arte.
2. En la segunda fase se lee y analiza detalladamente, la información previamente seleccionada, y se extrae el contenido más relevante para la investigación.

3. En la tercera fase se construye el estado de arte con la información de la fase anterior.
4. En la cuarta fase habiendo finalizado el estado del arte, se procede a levantar los requerimientos y a construir los modelos propuestos.
5. En la quinta fase, se da por finalizado el proyecto, y se establece si cumplió con lo propuesto en los objetivos específicos. Se elaboran las conclusiones.



## CAPITULO 2

### 2.1 ESTADO DE ARTE

Es necesario para el desarrollo del proyecto realizar un estudio del estado del arte, puesto que esto permite obtener información de los diferentes proyectos ya existentes que cumplen el mismo objetivo, de esta forma podemos usar esa información en pro de nuestro proyecto asegurándonos de no reinventar la rueda.

En el estado del arte se pretende dar una noción general de las herramientas ya existentes o en proceso de construcción que dan solución al problema planteado, en este caso se busca encontrar todas las herramientas que sean útiles en la síntesis de voz de textos con alto contenido matemático, para ello se divide el proyecto en dos fases:

- “Fase heurística: se procede a la búsqueda y recopilación de las fuentes de información, que pueden ser de muchas características y diferente naturaleza.
- Fase Hermenéutica: Durante esta fase cada una de las fuentes investigadas se leerá, se analizará, se interpretará y se clasificará de acuerdo con su importancia dentro del trabajo de investigación”.<sup>4</sup>

Posterior a la fase hermenéutica, se elaborará el documento, con la información, más relevante de las fuentes encontradas.

### 2.2 HERRAMIENTAS DE SÍNTESIS DE VOZ PARA DOCUMENTOS CON ALTO CONTENIDO MATEMÁTICO.

Es importante resaltar que hasta el momento no se ha encontrado una herramienta que permita la síntesis de voz de los documentos realizados en LaTeX, esto se debe a que la representación de símbolos matemáticos (integrales, derivadas, sumatorias...) en este lenguaje es una combinación de caracteres numéricos y alfanuméricos, que al ser leídos por un sintetizador de voz no se interpretan como la combinación de un conjunto de caracteres, sino que son leídos como caracteres individuales.

Tomando en cuenta lo anterior, no existe una herramienta que permita realizar la síntesis de voz, específicamente para documentos hechos en LaTeX, pero hay

---

<sup>4</sup>Seminario Estudios de Usuario. Universidad de Antioquia. Disponible desde: <http://goo.gl/Z49iFD>

herramientas que facilitan la lectura de textos con alto contenido matemático; a continuación se realizará una descripción detallada de esas herramientas.

## **2.2.1 SISTEMA DE ACCESO AUDIBLE A EXPRESIONES MATEMÁTICAS DIGITALES**

### **2.2.1.1 Introducción**

Como mencionan Carlos Arriola-Arciniega y FranciscoJ<sup>5</sup>, Aceves este software fue desarrollado con el fin de que los estudiantes de ingeniería tuvieran una herramienta que les permitiera interactuar de forma audible, con documentos con alto contenido matemático.

Dado que la red no se ha diseñado para ser accesible a personas en situación de discapacidad visual hay una gran cantidad de info-excluidos, esto genera una problemática que apenas comienza a afrontarse, con la inclusión de nuevas herramientas.

La aplicación se realizó usando un enfoque sistémico, de esta forma se consiguió desarrollar una aplicación que hace accesibles expresiones matemáticas complejas, como ecuaciones de segundo grado, integrales, gradientes.

### **2.2.1.2 Metodología de Desarrollo**

La metodología usada fue la de Sistemas Suaves de Checkland (1990), esta metodología permite plantear una solución a problemas donde hay un alto componente social, político, y humano. Al hacer uso de esta metodología se pueden analizar los agentes que participan en el proceso de desarrollo, de la siguiente forma:

---

<sup>5</sup>Carlos Arriola-Arciniega, Francisco J. Aceves. Herramienta auditiva para acceder expresiones matemáticas digitales. Sección de Estudios de posgrado e investigación, Escuela Superior de Ingeniería Mecánica y Eléctrica Zacatenco, Instituto Politécnico Nacional. México ,2010.

CATWOE	
<i>C</i>	= Clientes (beneficiarios, víctimas, discapacitados)
<i>A</i>	= Actores (los que participan en el sistema)
<i>T</i>	= Transformación (proceso básico que se realiza)
<i>W</i>	= <i>Weltanschauing</i> (perspectiva adoptada)
<i>O</i>	= Owners (propietarios del sistema)
<i>E</i>	= Entorno (constricciones del sistema)

*Figura 1.2.1 Agentes participantes en el proceso, representados mediante la metodología de sistemas suaves.*

También se hizo uso del modelo de complejidad de tres niveles de Sáez Vacas para representar los niveles de comunicación en el ciberespacio.

Al aplicar la metodología definida, se establecen unos criterios para la construcción de los modelos, a continuación se describen detalladamente:

### **Tipos y niveles de Comunicación en el ciberespacio**

Mediante el modelo de complejidad de tres niveles de Sáez Vacas para representar los niveles de comunicación en el ciberespacio, teniendo en cuenta que en el ciberespacio quienes se comunican directamente son las computadoras y estas a su vez sirven como mediadoras para la comunicación interpersonal, se definieron tres niveles de comunicación:

- a) Las relaciones de intercambio de información entre computadoras.
- b) Las relaciones de intercambio de información entre hombres y computadoras.
- c) Las relaciones de intercambio de información entre seres humanos a través de las computadoras.

Estos tres tipos de comunicaciones se dan simultáneamente, formando parte de un proceso: (a) para establecer contacto con otros seres humanos que están en otros puntos de la red, (b) nosotros nos comunicamos con las computadoras, (c) que a su vez se comunican entre sí.

### **Comunicación humano-computadora**

Mediante el uso de dispositivos de I/O (entrada-salida), las personas pueden establecer

una comunicación con los computadores. Para llevar a cabo esta comunicación debe haber un lenguaje que permita traducir el lenguaje humano a instrucciones comprensibles para una máquina.

Partiendo de la idea anterior, se puede decir que la comunicación con la computadora se limita a las instrucciones que la máquina pueda procesar, y que pueda llevar a una salida lógica o mecánica.

### **Formalización del modelo**

Después de realizar el modelo, se debe comprobar que cumple con unos criterios de validez, estos requisitos pueden ser comprobados, mediante una validación real, es decir cuando se aplique al ciberespacio, y responda a las necesidades planteadas.

### **Humanización / Convivialidad**

“Un sistema de diseño participativo debe comprender herramientas para poder acceder al ciberespacio, para poder realizar las actividades y proporcionar soporte a las actividades en grupo”.<sup>6</sup>

Teniendo en cuanto que el nivel de accesibilidad depende del estado del arte de la tecnología, las aplicaciones deben estar diseñadas de tal forma que sean de fácil uso para el hombre. Además se debe considerar que la tecnología avanza de una forma muy rápida, por lo cual ahora más que nunca se debe afrontar el problema de la accesibilidad.

### **Modelos**



<sup>6</sup> Carlos Arriola A, Francisco J. Aceves H. Innovación Tecnológica para acceder expresiones matemáticas digitales, pág.5.



Figura 1.2.2 Modelo de tres Niveles de Complejidad del ciberespacio.

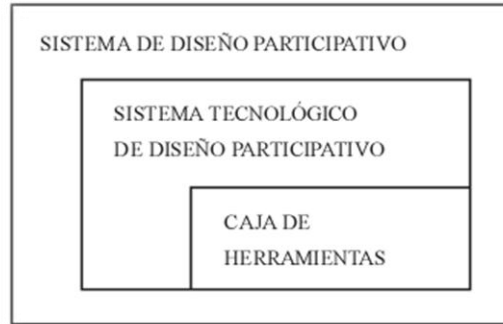
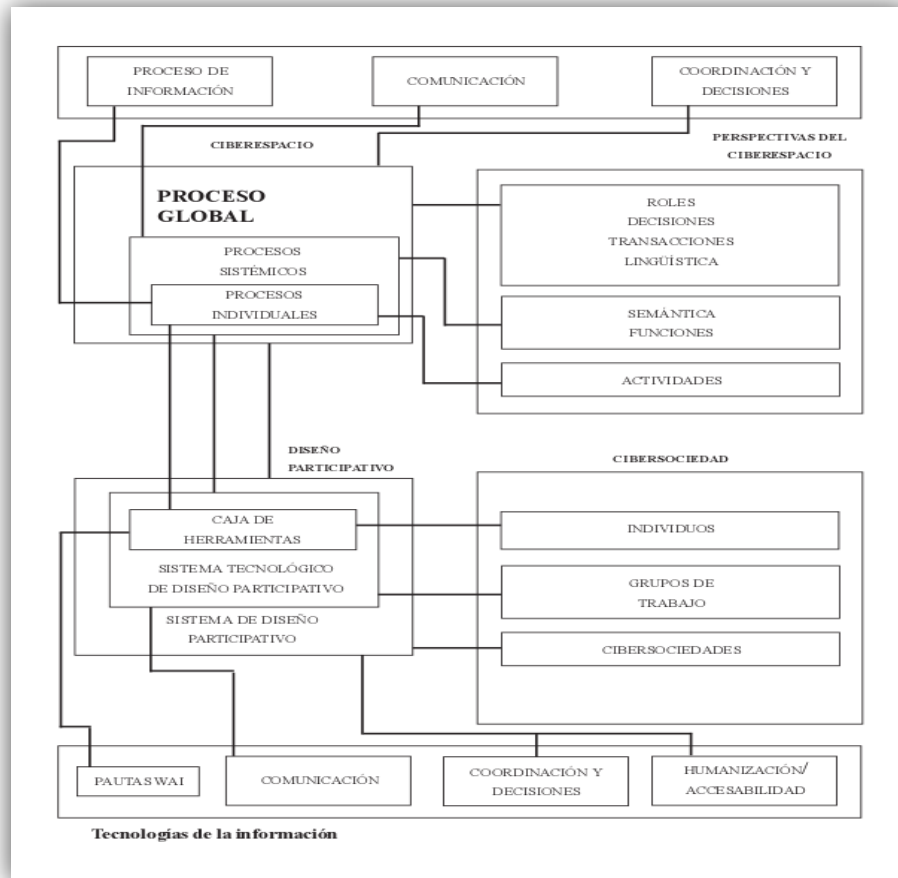


Figura 1.2.3 Modelo de tres Niveles de Complejidad del diseño participativo.

Al combinar los modelos de tres Niveles de Complejidad del Ciberespacio y del Diseño Participativo, así como el Modelo de Funciones, se obtiene el Modelo Completo del Ciberespacio.



*Figura 1.2.4 Modelo Completo del Ciberespacio y Diseño Participativo*

### **2.2.1.3 Sistema de acceso audible a expresiones matemáticas digitales**

A continuación se describen las herramientas utilizadas, para la implementación del sistema que permite la lectura de expresiones matemáticas para MathML, en la web.

#### **Herramientas Utilizadas**

##### **MathML**

Es un lenguaje de marcas que permite incluir lenguaje matemático en la web, a diferencia de HTML, se puede realizar edición de las expresiones, lo que facilita la reutilización de las mismas en otros documentos, además el acceso mediante medios no visuales.

MathML es una aplicación XML, y existen diversos editores como Amaya, MathType que permiten implementar código MathML, sin que el usuario tenga que recurrir a digitar directamente el lenguaje XML.

“MathML incluye dos formas de marcado que deben producir representaciones iguales, una de ellas es llamada marcación por presentación y la otra marcación por contenido. Están relacionadas con las dos formas en que puede pensarse o leerse cualquier expresión matemática”<sup>7</sup>.

##### **MathType**

Es una aplicación que permite editar ecuaciones matemáticas en Word, esta aplicación fue desarrollada por “Design Science”, entre sus funcionalidades se encuentra el exportar las expresiones matemáticas a MathML, debido a que genera el código XML.

##### **MathPlayer**

Es un plugin gratuito, para el navegador Internet Explorer, este plugin permite representar en una página web, las expresiones desarrolladas en MathML, además de otras funcionalidades que permiten agrandar el tamaño de la expresión y pronunciar esta, mediante el sintetizador de voz de Microsoft Reader.

---

<sup>7</sup>Carlos Arriola A, Francisco J. Aceves H. Innovación Tecnológica para acceder expresiones matemáticas digitales, pág.8

## **Jaws**

Es un lector de pantalla compatible con MathML y MathPlayer.

## **Método**

El procedimiento para la lectura de las expresiones se realiza de la siguiente forma:

1. Instalar MathType en el procesador de textos Word.
2. Instalar el plugin MathPlayer para el navegador Internet Explorer (versión 6.0 o mayor).
3. Editar las expresiones matemáticas en Word con MathType.
4. Exportar el código de las expresiones matemáticas con (Export MathPage) editadas en Word con MathType, de esta forma se tendrán las expresiones en formato html, y ya serán accesibles por el lector de pantalla Jaws 5.0, como se presenta a continuación.

## ***Lectura de datos para MathML:***

*Fórmula para solución de ecuaciones de segundo grado*

$$x = \left[ \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \right]$$

El lector de pantalla leerá: x igual a paréntesis izquierdo inicia fracción; menos b más menos inicia raíz cuadrada de b cuadrada menos 4 a c fin de raíz sobre 2 a, fin de fracción paréntesis derecho.

*Integral*

$$\int_{-\infty}^{\infty} 4x^3 dx$$

El lector de pantalla leerá: integral de menos infinito a más infinito de cuatro x cúbica dx.

## *Gradiente*

El lector de pantalla leerá:

$$\nabla (x_1, x_2, x_3) = x_1^3 + 2x_2 + x_3^2 + 1$$

gradiente del vector paréntesis izquierdo x1 coma x2 coma x3 paréntesis derecho igual a x1 cúbica más 2 x2 más x3 cuadrada más 1.

## **2.2.2 SISTEMA DE AYUDA A LA ESCRITURA DE TEXTOS ESTRUCTURADOS PARA ESTUDIANTES CON DÉFICIT VISUAL**

### **2.2.2.1 Resumen**

Como mencionan Esther Durá y Juan Domingo<sup>8</sup> la escritura de documentos grandes que contienen expresiones matemáticas es un proceso complejo para los estudiantes en situación de discapacidad visual, tomando en cuenta que los editores clásicos dan muy poco soporte para la edición de estos documentos.

El proyecto que se pretende realizar es una aplicación que facilite la escritura de documentos en LaTeX, el sistema permitirá que mediante instrucciones simples de voz, se pueda construir la estructura del texto, además que haciendo uso de la síntesis de voz la aplicación informe sobre errores, mensajes de estado, etc.

Algunas ventajas que proporciona el uso de lenguaje de marcas, para personas con déficit visual, son: la facilidad de conversión a otro formato (HTML, XML...), la posibilidad de incluir expresiones que se pueden traducir a MathML o a herramientas similares, la posibilidad de generalizar la aplicación a lenguajes de programación, la facilidad para trabajar de forma colaborativa.

### **2.2.2.3 Soluciones disponibles**

Hay una gran cantidad de aplicaciones libres para realizar la síntesis de voz, como el proyecto Orca, o el de KDE2, esto permite que cualquier persona pueda hacer uso de estas herramientas. También hay otras de código abierto como el NVDA.

Cabe resaltar que existen otras aplicaciones comerciales, de uso restringido, debido a sus licencias copyright, y a que están diseñadas para ser funcionales bajo una plataforma específica, tal es el caso de “SuperNova Access Suite5” o “GWMicro6”.

Algunas herramientas solo funcionan como lectores de pantalla, es decir que no toman en cuenta la estructura del texto, a diferencia de otros como el “SimplyWeb” o el lector web incluido en la distribución “LinuxSpeaks8”, que facilitan la accesibilidad a personas con déficit visual.

En cuanto a la escritura las herramientas son mucho más escasas, puesto que se

---

<sup>8</sup>Esther Durá y Juan Domingo. Un sistema de ayuda a la escritura de textos estructurados en LaTeX para estudiantes con déficit visual. Instituto de Robótica Universidad de Valencia, 2011.

considera que el uso de un teclado Braille resuelve el problema, pero esta no es una solución accesible, debido a que por ejemplo los no videntes desconocen la estructura global del texto.

#### **2.2.2.4 Lenguajes de marcas para NEE: déficit visual**

El uso de lenguajes de marcas, proporciona una gran cantidad de ventajas en la escritura, lectura, y edición de textos para las personas en situación de discapacidad visual e invidentes, puesto que la forma como se codifica el documento contiene marcas que definen la estructura interna, dando información detallada de su diseño.

Algunas de las principales ventajas son:

- Estructura clara: Cada una de las etiquetas de marcado en LaTeX, posee un significado semántico, lo cual facilita la comprensión de un documento, puesto que este ya está estructurado en capítulos, secciones, subsecciones, etc. Para las personas con déficit visual esto es una gran ventaja porque ellos pueden imaginar cómo es la estructura interna del documento.
- Facilidad de construcción de macros y órdenes: Los usuarios pueden cambiar la forma del documento fácilmente, por ejemplo la definición de márgenes, o el sistema de numeración, pueden adaptarse fácilmente mediante una orden que modifique el inicio del documento, también si una expresión aparece paulatinamente en el documento, puede ser reemplazada mediante una orden que redefina todo el bloque.
- Presentación automática: LaTeX permite fácilmente dar formato a un documento, obteniendo artículos de alta calidad, lo cual no permite discernir si este fue escrito por un no vidente.
- LaTeX facilita la escritura de:
  - Ecuaciones matemáticas: Las expresiones matemáticas se escriben en dos dimensiones, en LaTeX, se puede definir la estructura haciendo uso de las marcas, que permiten identificar la posición de los símbolos en la expresión.
  - Tablas: Las tablas se definen de manera similar a las ecuaciones, LaTeX proporciona una gramática en la que se establecen las dimensiones y la posición de cada elemento que la compone.
  - Conversión a formatos diferentes: LaTeX se puede exportar a otros formatos como HTML o XML, básicamente el código se puede convertir a otro lenguaje de marcas.

- Trabajos en grupo: LaTeX posee características para que los documentos puedan ser compartidos con otras personas no videntes, esto permite a un grupo de estudiantes tener una retroalimentación en los trabajos en clase, por ejemplo.

Debido a lo descrito anteriormente se eligió LaTeX, como el lenguaje de marcas base para la implementación de una aplicación que permita editar, escribir y leer documentos estructurados.

### **2.2.2.5 Desarrollo de un prototipo**

El prototipo de software debe cumplir con una serie de requisitos que verifiquen que la solución propuesta es coherente con los objetivos, para levantar estos requisitos se tendrán en cuenta, las herramientas a utilizar y la metodología de desarrollo.

#### **1.2.2.6 Requisitos**

- El sistema debe permitir la creación de la estructura vacía de un texto complejo, y navegar por ella mediante órdenes de voz, o teclas especiales (en el caso de usar un teclado braille), también debe emitir por voz la estructura del documento. Esta lectura debe tener en cuenta la lista de secciones, subsecciones, tablas, expresiones, y todo lo realizado hasta ese momento.
- El sistema debe ser de fácil aprendizaje y manejo, aun si el usuario conoce el funcionamiento de LaTeX, además la interfaz solo debería tener unas cuantas órdenes simples, para así evitar las confusiones entre ellas, y responder eficientemente.
- El sistema debe emitir mensajes de error, u otro tipo de mensajes mediante la síntesis de voz, además debe incluir mecanismos para la corrección de errores mediante órdenes de voz, tanto al momento de compilar el documento, como durante el tiempo de ejecución.
- El sistema preferiblemente debe estar diseñado para el uso de herramientas libres, que faciliten la distribución entre los usuarios, y puedan ser modificados por la comunidad de programadores.

### 2.2.2.7 Implementación

El sistema estará compuesto por un conjunto de bloques que se conectarán entre si usando pipes, fifos, u otras herramientas usadas por el sistemas tipo UNIX, para la administración de procesos.

Los componentes principales serán:

- Un reconocedor de voz restringido, y acoplado para que reconozcan un pequeño set de instrucciones.
- Un editor de texto ASCII, que permita visualizar en pantalla lo hecho, esto con el fin de que el sistema pueda ser usado por usuarios videntes o con déficit visual.
- Un sintetizador de voz para emitir los mensajes.
- Una distribución de LaTeX para compilar la aplicación.
- Un módulo que permita analizar los errores, y otro que secuencie las acciones de los componentes que se conectan entre sí.

La siguiente figura contiene los componentes principales del sistema:

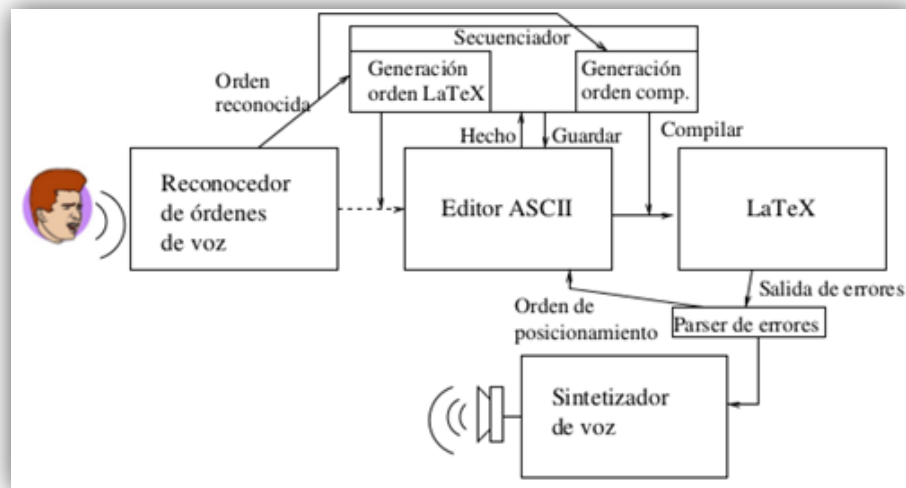


Figura 1.2.5 Esquema general del proyecto



Se podría hacer uso de la herramienta basada en CMU Sphinx10 para el reconocedor de voz. Para el editor podría usarse el emacs o vi, referente al sintetizador de voz se podría utilizar Festival 11.

En cuanto al analizador se podría implementar un script en Perl, y para el modulo que permite la interconexión de los componentes bastaría con desarrollar una aplicación en C o C++ que maneje los hilos, y procesos.

### **2.2.2.8 Conclusiones**

La conclusión más relevante es que usar lenguajes estructurados de marcas, en especial LaTeX, es una gran ventaja para los estudiantes con déficit visual o invidentes, gracias a que su estructura facilita la creación, edición y navegación sencilla de un documento.

## 2.2.3 ACTEMAT (ACCESIBILIDAD A TEXTOS MATEMÁTICOS)

### 2.2.3.1 Resumen

Como lo mencionan Yod Samuel Martín García y Javier López Moratalla<sup>9</sup>, ACTEMAT es una aplicación que permite a los usuarios con deficiencias visuales acceder a contenidos matemáticos mediante una representación sonora. Esto se logra mediante la traducción del lenguaje matemático al lenguaje natural.

La aplicación está diseñada para procesar el lenguaje estructurado de marcas MathML, pero usando el filtro adecuado se pueden traducir otro tipo de lenguajes.

ACTEMAT es mejor que otras aplicaciones, puesto que combina la semántica del lenguaje matemático con el lenguaje natural, además permite definir una serie de reglas para facilitar la lectura de las expresiones matemáticas.

### 2.2.3.3 Descripción

ACTEMAT permite la lectura de documentos en formato MathML, esto lo realiza generando un texto, con frases que son la representación sonora de las expresiones matemáticas, estas frases posteriormente pueden ser leídas por una gran cantidad de sintetizadores de voz.

Haciendo uso de la herramienta JSML (Java Speech Markup Language) se puede adicionar información a la salida de audio del sintetizador, así lograr que el audio suene de forma más natural.<sup>10</sup>

La aplicación puede interpretar los documentos hechos en el lenguaje de marcas MathML del World Wide Web Consortium (W3C). Teniendo en cuenta que muchos usuarios utilizan el formato TeX, a continuación se presentan las razones de haber escogido MathML y no TeX:

MathML es capaz de soportar la semántica de los elementos matemáticos a la vez que reglas de presentación, mientras que en TeX se tratan habitualmente de la misma manera elementos con distinta semántica pero igual representación gráfica.

MathML se puede emplear como salida de herramientas comerciales

---

<sup>9</sup>Accesibilidad a textos matemáticos. Yod Samuel Martín García y Javier López Moratalla. UPM Madrid.

<sup>10</sup>Java Speech Markup Language. [En línea] Disponible desde: <http://goo.gl/Ao2LKD>

(Maple, Mathematical), manteniendo toda la semántica que permiten dichas herramientas.

MathML es más regular que TeX y por tanto, más fácil de procesar. Sin embargo, admite tanta extensibilidad como TeX, gracias al elemento semantics y al atributo definition URL.

MathML es XML, lo que facilita la utilización de tecnologías existentes (parsers de XML, XSLT) para su procesado, así como su integración en documentos HTML.

MathML está concebido, según se indica en su especificación, para incluir a las interfaces audibles, si bien no logra este objetivo de la forma que sería deseable, por lo que sigue haciendo falta una gran cantidad de procesamiento para transformarlo a lenguaje hablado.<sup>11</sup>

La aplicación funciona usando unos ficheros de contexto, en los que el usuario define las reglas de traducción, de esta forma se puede traducir de MathML a JSML, evitando modificar directamente el XSL.

ACTEMAT posee un contexto básico, en el que están definidos los principales tipos de expresiones que usualmente se encuentran en un documento matemático, por ello el usuario solo debe definir un set de reglas para instanciar la plantilla, también puede crear y redefinir una ya existente.

#### **2.2.3.4 Estrategias de traducción**

Como se mencionó anteriormente la aplicación está diseñada de tal forma que el usuario pueda configurar la traducción de las expresiones matemáticas definiendo adecuadamente cada elemento del contexto.

También se definió que existe un contexto básico que cubre las principales expresiones matemáticas este contexto sirve para:

Traducir textos con expresiones comúnmente utilizadas, y permitir una traducción por defecto para otro tipo de expresiones.

- Ampliar la información del contexto, puesto que se pueden definir reglas de traducción para nuevos operadores, siempre y cuando estos estén definidos en el contexto básico.

---

<sup>11</sup>Accesibilidad a textos matemáticos. Yod Samuel Martín García y Javier LópezMoratalla. UPM Madrid. Pág.6

- Poseer un repositorio de plantillas previamente definidas que puedan ser de uso para ampliar el contexto con nuevas plantillas para nuevos tipos de expresiones.

### **2.2.3.5 Innovaciones y Mejores**

Utilizar un lenguaje estándar para la entrada y salida del traductor (MathML y JSML o SABLE respectivamente), es una gran ventaja ya que mejora la funcionalidad, puesto que permite utilizar el lenguaje con una gran cantidad de sistemas externos. Por ejemplo MathML puede usarse con otras herramientas comerciales para matemáticas como Maple, y al usar JSML, el proyecto ACTEMAT puede aceptar una gran cantidad de sintetizadores de voz para producir el audio.

Otra ventaja es que las personas con déficit visual o invidentes, pueden utilizar ACTEMAT con el TTS habitual, esto facilita la usabilidad del software.

Aparte del uso de lenguajes estándares para la entrada y salida, se ha hecho uso de lenguajes estándares para la comunicación interna entre los componentes del traductor (XML y XSL), esto hace el sistema sea versátil, debido a que se pueden usar los módulos por separado.

Es importante resaltar que es la primera solución implementada para conseguir la traducción al español, sin que ello implique limitar la posibilidad de utilizar el sistema con otros idiomas.

## 2.2.4 MAB (NAVEGADOR MATEMÁTICO DE AUDIO)

Hay muchos obstáculos para utilizar notación matemática en la web, por eso algunas personas recurren a soluciones como usar una imagen (GIF) para representar ecuaciones matemáticas, pero esto hace inaccesible a la web para persona e situación de discapacidad visual.

La notación matemática no solo es bastante compleja, también está en constante evolución. Continuamente los matemáticos inventan nuevas notaciones para representar conceptos e ideas innovadoras, pero actualmente las herramientas de accesibilidad (tales como los lectores de pantalla), no son capaces de reproducir esos símbolos. Para complicar más las cosas, el significado de los símbolos matemáticos puede cambiar de acuerdo al contexto en que estos son usados.

El descubrimiento de formas de mostrar la notación matemática tradicional dentro de un navegador presenta muchos desafíos y ni siquiera aborda la manera de comunicar de manera efectiva la información a las personas con discapacidad visual.

La Universidad de Toronto “AdaptiveTechnologyResource Centre” (ATRC) está desarrollando, un navegador matemático de audio (MAB), diseñado para hacer la notación matemática completamente accesible a individuos con discapacidad visual. Este esfuerzo forma parte de “ATRC's Network for Inclusive DistanceEducation” (NIDE), financiado por el programa de redes “Ontario'sTelecommunications Access Partnerships (TAP)”.

La aplicación MAB puede ser capaz de producir salida de audio para toda las páginas webs, que incluyan expresiones matemáticas creadas con MathML, MathML es una iniciativa del World Wide Web (W3C) capaz de mostrar virtualmente cualquier notación matemática. (Actualmente las aplicaciones para lectura de pantalla no son capaces de interpretar este tipo de información).

Debido a la complejidad de las expresiones matemáticas, el MAB agregará otra dimensión a la salida de audio, mediante la inclusión de inflexión al sintetizador de voz a través del tono de voz, la velocidad del habla, etc.

Adicionalmente, los usuarios del MAB podrán clarificar o aclarar el audio, aplicando reglas específicas de matemáticas a las ecuaciones dentro de una página web. Si por ejemplo, los usuarios realizan un documento que contiene aritmética, un cliente podría cargar un conjunto de directrices para asegurar que los símbolos están siendo interpretados por las expresiones matemáticas.

Cuando los videntes examinan una ecuación matemática, ellos son capaces de separar

el contenido visualmente y dividirlo de tal forma que sea comprensible. Aquellos que dependen de los sintetizadores de voz, no tienen este tipo de control, porque escuchar es en esencia un proceso pasivo, y conlleva a que se dificulte extraer información de una gran cantidad de datos.

El MAB permitirá a los usuarios controlar el ritmo y la velocidad de lectura de las expresiones matemáticas. La aplicación incluirá un mecanismo para dividir el contenido en bloques de información pequeños, ordenados lógicamente, que pueden ser más fácilmente procesados por los oyentes.

Respecto al futuro, las siguientes versiones del MAB incluirán formas más sofisticadas de analizar ecuaciones matemáticas y los usuarios tendrán conjuntos más amplios de reglas disponibles para aplicar a los documentos de la web.<sup>12</sup>

---

<sup>12</sup>MathML Accessibility Project [En línea] Disponible desde: <http://goo.gl/fAeduL>

## 2.2.5 ASTER (SISTEMA DE AUDIO PARA LECTURAS TÉCNICAS)

### 2.2.5.1 Introducción

AsTeR es una aplicación que acepta la notación TeX como entrada y produce una representación como salida, que es más audible que visual.

### 2.2.5.2 Interprete Matemático

AsTeR tiene tres componentes principales. Un reconocedor que interpreta la notación LaTeX y crea una representación interna que es fácil de manipular para el programa. Un lenguaje de audio llamado AFL, que representa el texto interpretado usando voz y nonspeech sonidos. El tercer componente es un sistema para la navegación de audio.

El reconocedor extrae la estructura e incluye lo más significativo del TeX codificado. Cuando las expresiones matemáticas son recibidas, se interpreta la entrada completamente antes de que comience la representación de audio. El texto generado, incluye algunas cosas simples, que el sistema de síntesis de voz debe hacer, por ejemplo, un símbolo de pregunta al final de la oración puede alterar la entonación del comienzo, pero los sistemas de síntesis de voz nunca reorganizan las palabras dichas.

Un ejemplo simple es el formato de audio de la expresión  $\log_{10}x$ , que un oyente debería escuchar como “logaritmo de x en base 10”, en la creación de esta representación en voz, AsTeR, no puede simplemente procesar los símbolos en su secuencia original. Las integrales presentan un cambio similar, porque los oyentes necesitan conocer cuáles son las variables de la integral tan pronto como sea posible.

La integral:

$$\lim_{x \rightarrow \infty} \int_0^x e^{-y^2} dy = \frac{\sqrt{\pi}}{2}$$

Podría ser leída como “la integral con respecto a x, de cero a infinito, de e a la menos x, dx” LaTeX codifica esto como:

$$\int_0^{\infty} e^{-x} dx$$

AsTeR requiere buscar la  $\int dx$  al final de la expresión antes de que pueda ser representada  $\int$  del comienzo. Para generar las representaciones de textos como ese, AsTeR debe descomponer la expresión en las piezas que lo componen y entonces re-

ensamblarlas en diferente orden.

Las expresiones matemáticas están estructuradas como árboles. La ecuación  $y=x+2$  pueden ser entendida como un árbol que tiene el operador '=' como raíz que tiene adicionalmente dos ramas 'x' y '2'. El árbol puede ser representado con la notación en prefijo como  $(= (+ x 2))$ . AsTeR emplea una notación similar internamente.

La representación matemática en TeX, es altamente ambigua; una expresión tal como 'f(x+y)' puede significar "el producto de 'f' y 'x+y' ", o también puede significar "la función 'f' aplicada a 'x' + 'y'".

Los superscripts son otras construcciones que pueden tener múltiples significados: ' $x^{-1}$ ' significa '1/x' pero ' $\sin^{-1}$ ' se refiere al inverso de la función seno;  $A^T$  es probablemente la transpuesta de una matriz y  $D^2$  puede ser la segunda derivada. El reconocedor de AsTeR es capaz de resolver muchas de esas ambigüedades, pero los usuarios pueden especificar como una notación en particular va a ser interpretada.

### **2.2.5.3 Representando las expresiones matemáticas mediante voz**

Cuando el reconocedor ha hecho su trabajo, el segundo componente de AsTeR se hace cargo de representar lo interpretado mediante sonidos; esto lo hace aplicando las reglas en AFL, el lenguaje de formato de audio. Las reglas determinan que palabras serán pronunciadas y como serán pronunciadas, es decir controlan el tono, la velocidad de la voz y otra gran variedad de cualidades; las reglas también invocan las señales de nonspeech.

Las reglas estándares en AsTeR para la reproducción de fracciones leen una simple expresión tal como  $a/b$  como "a sobre b", pero una instancia más complicada como ' $(x + y)/(x - y)$ ' es tomada como "la fracción con numerado 'x+y' y denominador 'x-y'". Son pocos los casos especiales, por ejemplo ' $1/2$ ' puede ser reproducido como "un medio", esto es mejor que "uno sobre dos". Todas las reglas del AFL están sujetas a la modificación.

Cuando se reproducen superscripts y subscripts cambian las características de voz, por ejemplo los superscripts son leídos en tonos altos y los subscripts en tonos bajos. Tales señales de voz ayudan a resolver ambigüedades en la reproducción de audio. Por ejemplo, ' $X^n + 1$ ' se distingue rápidamente de ' $X^{n+1}$ ', incluso sin una referencia que indique donde terminan los exponentes.

Los sonidos nonspeech proveen una forma concisa y discreta de transmitir características al texto. En una lista de viñetas, un tono breve puede anunciar cada vez un nuevo ítem, esto es mejor que repetir la palabra "viñeta". Los sonidos pronunciados



continuamente en el fondo mientras continua la voz pueden servir para enfatizar o resaltar una parte del texto, así se provee una un audio equivalente de letra cursiva y negrita.

#### **2.2.5.4 Escuchando de forma Activa**

AsTeR da control al oyente para navegar por el flujo de voz, la clave de esto es posiblemente la representación interna en forma de árbol del texto, que describe los detalles de la expresión matemática y la arquitectura a gran escala de todo el documento.

Los comandos de teclado permiten la navegación rápida a través del árbol, con mayor flexibilidad que los controles de avance rápido y rebobinado de un casete.

## **2.2.6 PROSODY**

### **2.2.6.1 Introducción**

MathML provee una rica solución para visualizar símbolos especiales y en dos dimensiones que normalmente son requeridos en la notación matemática. Adicionalmente este software soporta la representación matemática en la Web.

Sin embargo la lectura de expresiones matemáticas representadas por lenguajes de marcas como MathML, plantean desafíos en la producción automática de síntesis de voz, para las aplicaciones TTS y los lectores de pantalla.

En este artículo se pretende demostrar y explicar el diseño de un lector de código abierto para la lectura de documentos electrónicos mediante síntesis de voz.

Este lector de documentos está orientado a los estudiantes de colegio con discapacidades para la lectura. El proyecto está diseñado para hacer uso de Prosody e inferencias de contexto, que permiten hablar las expresiones matemáticas de una forma natural, imitando la forma como un humano leería la expresión.

El lector de documentos trabaja como un complemento del navegador web Firefox. Se escogió el navegador de voz Firefox, por las ventajas de no costo, amplia disponibilidad y capacidades integradas para la representación visual de MathML. Adicionalmente, Charles Chen ha producido, un conjunto de librerías y componentes para los sistemas TTS, que se usan para la producción de texto a voz.

### **Área del Problema**

La producción automática de síntesis de voz no es un problema nuevo, desde hace una década, sistemas de software especializado han intentado leer expresiones matemáticas complejas en pro de los discapacitados visuales. Un enfoque adoptado por los desarrolladores de software es añadir componentes léxicos, como signos de puntuación al proceso de habla matemática. Los componentes léxicos pueden ser frases como “abro paréntesis”... “cierro paréntesis”, o “comienza fracción”... “finaliza fracción”.

### **2.2.6.2 Prosody**

Un enfoque alternativo al proceso de habla de las matemáticas, es insertar componentes prosódicos en la lectura de las expresiones, esto se refiere a pausas, énfasis, acentos y tonos, como lo suelen hacer los humanos.

Los motores TTS integrados en Windows y en algunos computadores personales de Apple son capaces de recibir instrucciones de marcas prosódicas, y generar sonidos comprensibles. Desafortunadamente las instrucciones de marcas prosódicas no funcionan bien en gran cantidad de plataformas, pero el proyecto está orientado a dar soporte para los sintetizadores de voz de la W3C, SAPITTS de Microsoft, y comandos de voz embebidos en los sistemas de Apple.

### **2.2.6.3 Inferencias contextuales**

Para complicar el problema de producción de voz para las expresiones matemáticas, se debe tener en cuenta que los símbolos pueden ser representados de diferentes formas, esto depende del contexto.

Basados en los trabajos de David McKain, el lector de documentos usa reglas heurísticas para inferir precedencias entre los operadores, que normalmente están implícitas en multiplicaciones y funciones dentro de las expresiones matemáticas.



## **CAPITULO 3**

### **3.1 LECTORES DE PANTALLA (LSR)**

Los lectores de pantalla son aplicaciones de software que tratan de identificar e interpretar aquello que se muestra en pantalla, esta interpretación se representa mediante síntesis de voz, siendo útil para las personas con discapacidad visual.<sup>13</sup>

#### **3.1.1 ORCA**

##### **3.1.1.1 Introducción**

Orca es una ayuda técnica libre/código abierto, flexible, extensible y potente para las personas ciegas y deficientes visuales que utilizan el sistema operativo Linux. Usando varias combinaciones de voz, braille y magnificación Orca ayuda a proporcionar accesibilidad a muchas aplicaciones.

Orca proporciona las siguientes funcionalidades:

- Lector de pantalla: proporciona acceso no visual a las aplicaciones estándar de Oracle Solaris Desktop mediante la voz y la salida braille.
- Magnificador: permite el registro del enfoque automático y la magnificación a pantalla completa a fin de ayudar a los usuarios con baja visión.

##### **3.1.1.2 Voces disponibles**

Orca proporciona interfaces para los servicios de voz gnome-speech y emacspeak. Las voces para Orca de la actualidad se restringen mediante los motores de voz admitidos por los servicios de voz que hay disponibles.

A continuación, se mencionan algunos de los motores de voz que se encuentran disponibles:

Motores libres:

- ESpeak

---

<sup>13</sup><http://accesibilidadweb.dlsi.ua.es/?menu=lectores>

- Festival
- FreeTTS

Motores comerciales:

- Fonix DECtalk
- Loquendo
- Eloquence
- Cepstral

### **3.1.1.3 Magnificación**

Actualmente, Orca utiliza el servicio de magnificación de gnome-mag. El servicio gnome-mag ahora admite la magnificación a pantalla completa con más suavizado, que se basa en extensiones más recientes del servidor del sistema de ventanas X. Estas extensiones no siempre funcionan bien en todas las plataformas. Por lo tanto, es posible que la magnificación a pantalla completa con suavizado no siempre funcione.<sup>14</sup>

## **3.1.2 NVDA**

### **3.1.2.1 Introducción**

NVDA es un lector de pantallas para Microsoft Windows gratuito. Es un proyecto de software libre, por lo que también está disponible el código fuente del programa de forma gratuita.<sup>15</sup>

Este lector de pantalla proporciona soporte para:

- Navegar por la web.
- Leer y escribir documentos con programas tales como el Wordpad y el Microsoft Word.
- Enviar y recibir email con el Outlook Express.
- Usar programas de líneas de comandos en Dos de Windows.
- Producir hojas de cálculo básicas en Microsoft Excel.

---

<sup>14</sup>Orca Screen Reader and Magnifier. Disponible desde: <http://goo.gl/YdftNG>

<sup>15</sup>Universidad de Alicante. NVDA. Disponible desde: <http://accesibilidadweb.dlsi.ua.es/?menu=nvda>

- Manejo general del ordenador a través de Mi PC / Explorador de Windows, Panel de Control y otras tareas genéricas de Windows.
- Posibilidad de instalarlo y usarlo desde un dispositivo USB.
- Sintetizador de voz integrado que soporta más de 20 idiomas

### **3.1.2.2 Sintetizadores de voz soportados**

NVDA permite al usuario leer el contenido de cualquier lenguaje, siempre y cuando exista un sintetizador de voz que soporte el lenguaje. El NVDA usa el eSpeak como sintetizador de voz por defecto.

### **3.1.2.3 Implementación y diseño**

NVDA está escrito básicamente en Python, esto hace que tenga una cantidad de ventajas. El código que debe ser inyectado dentro de otros procesos está escrito en C++ para alto rendimiento. Su diseño abstracto permite al NVDA ser extensible; además está construido bajo el paradigma Orientado a Objetos.<sup>16</sup>

### **3.1.2.4 Licencia**

NVDA está cubierto por la GNU General Public License (Versión 2). Esto implica que hay libertad para compartir o modificar este programa de cualquier modo, distribuyendo la licencia junto con el programa, y hacer todo el código fuente disponible a quien lo quiera. Esto se aplica al original y a las copias modificadas del programa, más cualquier software que utilice código tomado desde este programa.

NVDA es copyright 2006-2011 por el colectivo de colaboradores de NVDA.

---

<sup>16</sup>Nvaaccess. Home of the free. Disponible desde: <http://goo.gl/l3LgVD>





## CAPITULO 4

### Desarrollo de la Investigación

La construcción del software es una posible solución, pero hay otro método para lograr reproducir mediante la síntesis de voz, un texto hecho en LaTeX. Este método se pudo deducir gracias a la previa investigación.

El método consiste en traducir las expresiones matemáticas hechas en LaTeX, a otro lenguaje matemático, que si pueda ser reconocido por un sintetizador. Para ello se puede exportar el código en LaTeX al lenguaje de marcas MathML.

#### 4.1 LaTeX EN LA WEB

##### LaTeX2MathML

Esta es una librería implementada en PHP5 que permite convertir fórmulas matemáticas en LaTeX a XML+MathML. Esta librería da soporte en línea, y puede ser instalada en el equipo mediante un script.

Se puede utilizar el demo, que se encuentra en la página: <http://latex2mathml.freewebmaster.fr/demo.php>, para convertir las expresiones matemáticas.<sup>17</sup>

##### MathJax

Es una aplicación implementada en Javascript, de código abierto que permite visualizar ecuaciones matemáticas que trabajan en todos los navegadores.

##### Características:

- MathJax usa hojas de estilos (CSS), y fuentes, para representar las ecuaciones en lugar de imágenes, que no pueden ser leídas por un lector de pantalla.
- MathJax permite incluir expresiones matemáticas en wikis, blogs, páginas web.

---

<sup>17</sup>LaTeX2MathML : from LaTeX to MathML. Disponible desde: <http://latex2mathml.freewebmaster.fr/index.php>

- MathJax permite cortar y pegar las ecuaciones matemáticas de su página web, dentro de documentos en Word, y LaTeX, además de otras aplicaciones.
- MathJax es compatible con los lectores de pantalla usados por las personas con discapacidad visual, además tiene características de Zoom, para permitir a todos los lectores leer pequeños detalles.
- MathJax permite convertir TeX a MathML.

MathJax puede instalarse fácilmente descargando el paquete correspondiente a su SO, desde la página de MathJax. También se encuentra en los repositorios de Ubuntu.

MathJax trabaja con el MathPlayer, este puede detectar cuando el Plug-in esta instalado en IE, y reproducir las expresiones matemáticas, siempre y cuando se use MathML como la representación matemática de las ecuaciones [2].

### **TeX4ht**

Es una aplicación para convertir documentos escritos en TeX, LaTeX, ConTeX al lenguaje de etiquetas HTML, a XML, Braille entre otros, haciendo uso de MathML.

Usando TeX4ht se puede producir un formato XHTML y MathML, que puede ser usado con un sistema TTS, y lectores de pantalla.

### **PlasTeX**

PlasTeX es un framework para el procesamiento de documentos en LaTeX, está escrito en Python. Esta aplicación permite generar el formato HTML y MathJax, que se puede visualizar en muchos navegadores web, entre ellos Firefox, Chrome, Safari.

### **MathType**

MathType es un editor de ecuaciones que trabaja con Microsoft Word, esta aplicación permite escribir las expresiones directamente en el editor de Word, y exportarlas al formato XHTML y MathML, que pueden ser leídos por lectores de pantalla y sintetizadores de voz.

Usando los lenguajes para transformar el documento en LaTeX, en otro lenguaje, se puede lograr que sea accesible para las personas con discapacidad visual. Para ello podemos usar:

## **XHTML+MathML: IE, MathPlayer y voz**

El MathPlayer es requerido para poder visualizar y reproducir mediante síntesis de voz las expresiones matemáticas. Este Plug-in es libre y puede descargarse de la página. Usando el Jaws como lector de pantalla se pueden reproducir los textos, en Internet Explorer (IE).

## **4.2 RECONOCIMIENTO DE SÍMBOLOS MATEMÁTICOS EN LaTeX**

Partiendo del estado del arte, en el transcurso de la investigación se pudo refutar la hipótesis que planteaba, que no existía una herramienta que permitiera realizar la síntesis de voz para documentos con alto contenido matemático. Esta herramienta conocida como AsTeR no solo reconoce las expresiones matemáticas sino también, en general todo lo concerniente a LaTeX, es decir que reconoce párrafos, macros, y en su mayoría todas las funcionalidades de este lenguaje de marcas.

Estudiando la herramienta AsTeR, se pudo encontrar que aunque es una herramienta muy potente, existen algunos problemas de accesibilidad, entre estos:

- La herramienta lee las expresiones en inglés, es decir que no es accesible para las personas de habla hispana
- La herramienta solo funciona bajo el entorno Emacs, puesto que el software corre como un subproceso de esa aplicación.
- La herramienta está diseñada en LISP, esto no implica que sea obsoleta, pero cabe resaltar que en este momento existen lenguajes de mucho mas nivel, como Python, que también trabaja con la librería Lex y Yacc.
- La herramienta es poco accesible teniendo en cuenta que trabaja en Emacs, y que realmente este es un entorno muy poco amigable con los usuarios.
- La herramienta en el diseño tiene su propio sintetizador, puesto que AsTeR, trabaja con dos módulos el AFL para producir la salida de audio, y el navegador. Esto hace que el usuario no pueda interactuar con su sintetizador de voz genérico, como lo permiten otras herramientas como el ACTEMAT.

Tomando en cuenta lo anterior, entrar a modificar el código fuente de AsTeR, es una solución poco eficiente, por lo tanto se plantea la construcción de una nueva aplicación que se desarrolle usando el lenguaje Python, que funcione con un sintetizador de voz

genérico, y de preferencia de código abierto.

También se debe tener en cuenta que la herramienta AsTeR, puede otorgar una base para la construcción del nuevo software, partiendo de unas definiciones básicas, como la gramática de LaTeX, que definió el señor T.V, Raman.

## **4.2.1 Especificación de Requerimientos**

### **Introducción**

Documento de requerimientos para el proyecto de diseño de un software para el reconocimiento de símbolos matemáticos para LaTeX.

### **Propósito**

Con el fin de realizar una aplicación que permita reconocer mediante un sintetizador de voz, textos con alto contenido matemático realizados en LaTeX, se hará una lista de requerimientos, en los que se establecerán los criterios necesarios para facilitar la construcción del software.

### **Ámbito del Sistema**

- El sistema debe reconocer las expresiones matemáticas en LaTeX y generar la representación de cada expresión matemática mediante una gramática definida por el desarrollador.
- El sistema debe generar un documento externo donde se encuentre la estructura de la expresión matemática, expresada en algo similar al lenguaje natural.
- El sistema debe permitir la interacción con un sintetizador de voz, que pueda navegar por el documento externo, y leer la estructura general del mismo.
- El sistema debe estar implementando usando herramientas libres, para que sea accesible a cualquier usuario, y para que el código fuente pueda ser modificado por la comunidad de desarrolladores.

### **Referencias**

- El proyecto será realizados bajo la metodología de trabajo Espiral, que permite realizar entregas incrementales de la aplicación.
- La especificación de requerimientos fue realizada según el estándar de la IEEE 830.

## **Visión General del documento**

El documento de especificación de requerimientos debe permitir establecer las necesidades para la implementación del aplicativo teniendo en cuenta factores como:

- La usabilidad de la aplicación tomando en cuenta la comunidad de usuarios a quien va dirigido.
- Los recursos de software y hardware necesarios para el correcto funcionamiento del aplicativo.
- Los componentes necesarios para construir el software.

## **Descripción General**

### **Perspectiva del producto**

La aplicación para el reconocimiento de símbolos matemáticos que se diseñará, se realizará tomando en cuenta los requerimientos establecidos. El principal objetivo del software, es poder brindar a las personas con discapacidad visual la posibilidad de acceder a documentos escritos en LaTeX, ya que como se ha encontrado en el estudio del estado del arte, solo existe la herramienta conocida como ASTER, que no es accesible para la comunidad hispana, además de que cuenta con otro tipo de falencias debido a su complejidad de uso.

### **Funciones del producto**

- El software permitirá leer el contenido de un documento hecho en LaTeX y reconocer las expresiones matemáticas allí definidas, como integrales, sumatorias, límites, entre otras.
- El software permitirá generar la representación de cada expresión matemática reconocida,

- El software deberá traducir la representación de las expresiones, a un lenguaje cercano al lenguaje natural.
- El software permitirá la creación de un documento que contenga la estructura general (traducciones) del documento en LaTeX.
- El software permitirá que un sistema TTS pueda navegar por el documento creado, e interpretar esas traducciones mediante el uso de voz.

### **Características de los Usuarios**

Se asume que los usuarios del software deben tener conocimientos básicos sobre el manejo del computador. Además que han trabajado con la herramienta LaTeX.

### **Restricciones**

- El sistema solo funciona para el lenguaje de marcas LaTeX.
- El software se implementará usando el lenguaje Python 2.7.3.
- Se utilizara el sintetizador de pantalla NVDA.
- Para la construcción de los modelos se usará el lenguaje unificado de modelado UML 2.0.

### **Suposiciones y Dependencias**

- El software solo corre bajo versiones superiores a Python 2.7.3.
- Las herramientas utilizadas serán libres, y las proporcionará el equipo de trabajo.
- El rendimiento de la aplicación estará limitado por las capacidades físicas de la maquina donde se utilice.

### **Requisitos Futuros**

- La aplicación se puede extender a la lectura de todo el contenido de un

documento hecho con LaTeX.

- El sistema podría utilizar el formato de texto JSML para lograr una salida de audio más natural, y comprensible para el oyente.
- La aplicación podría ser más interactiva con el usuario, permitiendo que él pueda definir reglas de traducción teniendo en cuenta que la simbología matemática es muy ambigua.

### **Requisitos Específicos**

- Construir la gramática regular BNF, definición de símbolos terminales, no terminales y reglas de producción.
- Implementar las expresiones regulares en Python, para expresiones matemáticas básicas como: exponentes, integrales, sumatorias, límites, así como de sistemas de ecuaciones complejos.
- Implementar las reglas para la interpretación de los lexemas, o expresiones regulares.
- Implementar funciones para el manejo de errores, de símbolos no reconocidos por el analizador en la trama de entrada.
- Definir tipo de análisis sintáctico.
- Implementar de las reglas de derivación de la gramática formal, y la tabla de precedencias para los operadores.
- Implementar el generador de AST (árbol de sintaxis abstracto), que permita analizar la semántica de la gramática BNF definida.
- Definir e implementar las reglas de traducción de los lexemas.
- Implementar generador del metalenguaje para LaTeX.

## 4.3 MODELO DE SOFTWARE

### 4.3.1 Esquema General

El modelo del proyecto integra tres componentes de software, con los cuales se pretende dar cumplimiento a los requisitos y métricas definidas en la etapa de análisis y levantamiento de requerimientos.

Con el fin de lograr los objetivos, se ha propuesto la creación de los siguientes componentes:

- Reconocimiento de símbolos
- Procesamiento de símbolos.
- Lector de símbolos.

El componente de reconocimiento de símbolos, integra a su vez los siguientes tres módulos:

- Modulo para el análisis léxico.
- Módulo para el análisis sintáctico.
- Módulo para el manejo de errores.

El componente para el procesamiento de símbolos se encarga de representar en el lenguaje natural los símbolos terminales de la gramática.

El lector de símbolos es un componente externo que se encarga de la traducción sonora de las expresiones matemáticas, para ello se puede hacer uso de un sintetizador de voz.



### 4.3.2 Diagrama de Componentes

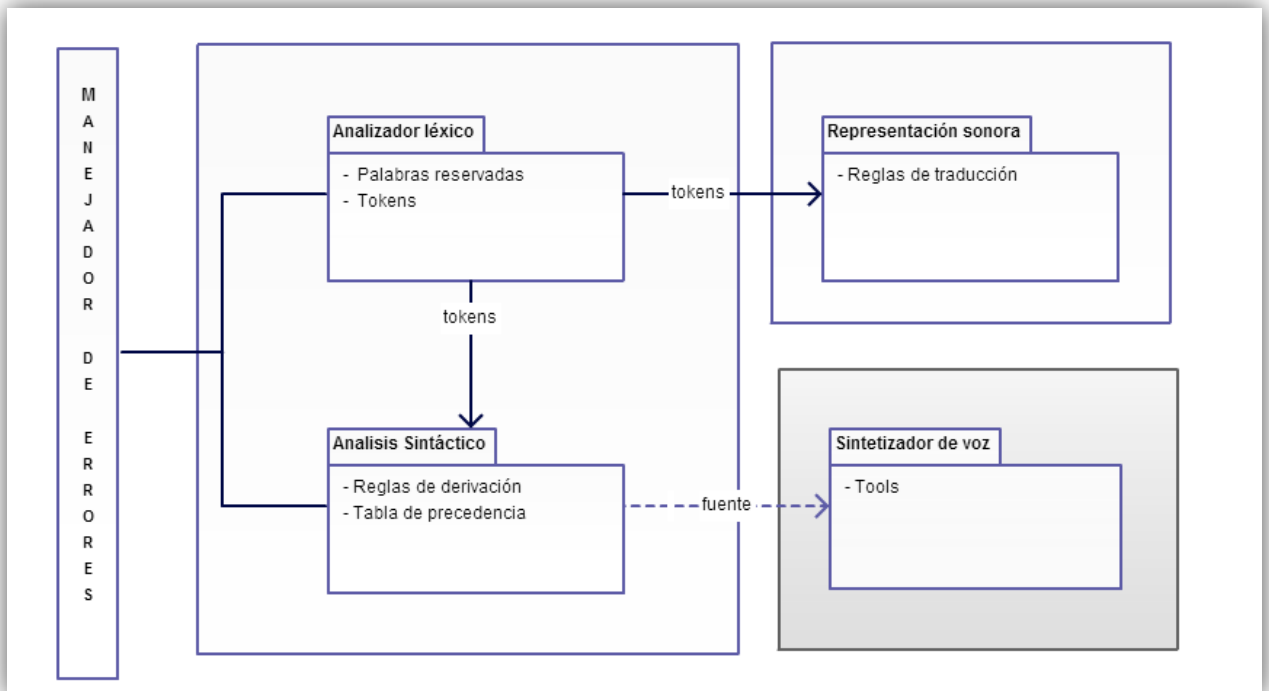


Figura 4.1.1 Arquitectura del modelo propuesto.

#### Modulo para el análisis léxico

En el analizador léxico se definen los símbolos, y las palabras reservadas, del lenguaje, el analizador agrupa los caracteres de la entrada o programa fuente, y verifica que estos si estén definidos en el lenguaje de LaTeX.

Para la construcción del analizador léxico se debe tener en cuenta que el lenguaje LaTeX cuenta con una vasta cantidad de símbolos matemáticos, entre ellos:

- Letras Griegas
- Símbolos de tamaño variable.
- Operadores binarios.

Lo que se pretende con el análisis léxico, es agrupar todos los símbolos de tal forma que se puedan construir componentes léxicos, que posteriormente serán utilizados

como entrada del módulo para el análisis sintáctico.

En la construcción del analizador léxico, se debe tener en cuenta que no todos los símbolos de la gramática que compone a LaTeX, serán reconocidos por el analizador, la idea es que inicialmente este solo reconozca los símbolos matemáticos, y además omita expresiones como comentarios, o símbolos como espacios en blanco y saltos de línea.

Basados en el desarrollo realizado por el señor Raman en su tesis de maestría, para la presente investigación y la construcción del analizador léxico, se propuso la clasificación de los símbolos matemáticos en las siguientes categorías:

- **Operadores ordinarios:** Letras, números y otros símbolos.
- **Operadores grandes:** Operadores grandes, normalmente unarios, como los símbolos para las integrales y sumatorias.
- **Operadores binarios:** Operadores binarios, como la unión, la intersección.
- **Operadores relacionales:** Operadores lógicos, como el símbolo de mayor, menor, menor igual...
- **Delimitadores:** Operadores que sirven para el cierre, apertura o la agrupación de las expresiones tales como, los paréntesis, las llaves, los corchete.

Partiendo de la definición y la clasificación de los símbolos se puede hacer uso de la librería para Python PLY, que provee herramientas para el procesamiento de símbolos, y el manejo de errores, como Lex y Yacc.

### **Modulo para el análisis sintáctico**

El análisis sintáctico, agrupa los lexemas del programa fuente en frases gramaticales esto se puede realizar gracias a las reglas de producción. Esta representación se realiza mediante un árbol de sintaxis abstracto, que muestra la jerarquía entre lo componentes léxicos.

Para el análisis sintáctico se debe definir la precedencia de los operadores, para ellos se construye una tabla, y se le asigna un peso (que puede ser numérico) a cada operador.

Para la construcción del AST, se deben definir las reglas de derivación, para ello se hace uso de la gramática, y de la herramienta para el análisis sintáctico Yacc. El analizador léxico, arroja los componentes léxicos del archivo tex leído, estos componentes son procesados por el analizador sintáctico quien construye la representación de las expresiones, tratando a cada elemento como un hijo del árbol.

Es importante realizar el análisis sintáctico dado que se construye la estructura del flujo de datos, lo que permite acoplar esta estructura de acuerdo a las necesidades de otros componentes como el de procesamiento de símbolos.

Un ejemplo más claro de la representación interna de los objetos matemáticos se puede visualizar para la expresión  $x-y$ , que para el compilador es una expresión de la forma: raíz +, e hijos  $x$  y  $y$ . Representando las expresiones matemáticas mediante nodos, se puede construir estructuras cada vez más complejas.

### **Modulo para la representación sonora**

En este módulo se definen las reglas de traducción de las expresiones matemáticas, puesto que los símbolos ya están definidos en el analizador léxico, lo que se define es la representación sonora de cada símbolo según el lenguaje natural.

Haciendo una analogía entre el proyecto AsTeR, y este proyecto se puede decir que ambos poseen dos módulos comunes, el analizador léxico, y sintáctico, pero el AFL y el navegador de código, que caracterizan a la aplicación de Raman, no se utilizaron para este proyecto, puesto que desde mi perspectiva, desarrollar un sistema para la traducción de audio es redundante y acota la accesibilidad del software. Es por ello que partiendo de la gran cantidad de herramientas libres existentes, se puede hacer uso de cualquier lector de pantalla o sintetizador libre, para hacer genérica la aplicación.

Es difícil lograr una correcta traducción de los símbolos dada la ambigüedad de las expresiones, un caso común se presenta en la traducción de los exponentes, por ejemplo la expresión  $A$  elevada a la  $T$ , se podría interpretar como un exponente cualquiera, pero normalmente es usada para representar la matriz inversa.

Además de los problemas de ambigüedad, hay otro problema que debe ser tomado en cuenta en el presente desarrollo y es que la lectura de algunas expresiones se hace muy confusa cuando no se hacen las pausas necesarias, para que el oyente pueda entender de la manera correcta la expresión.

### 4.3.3 Diagrama de Casos de Uso

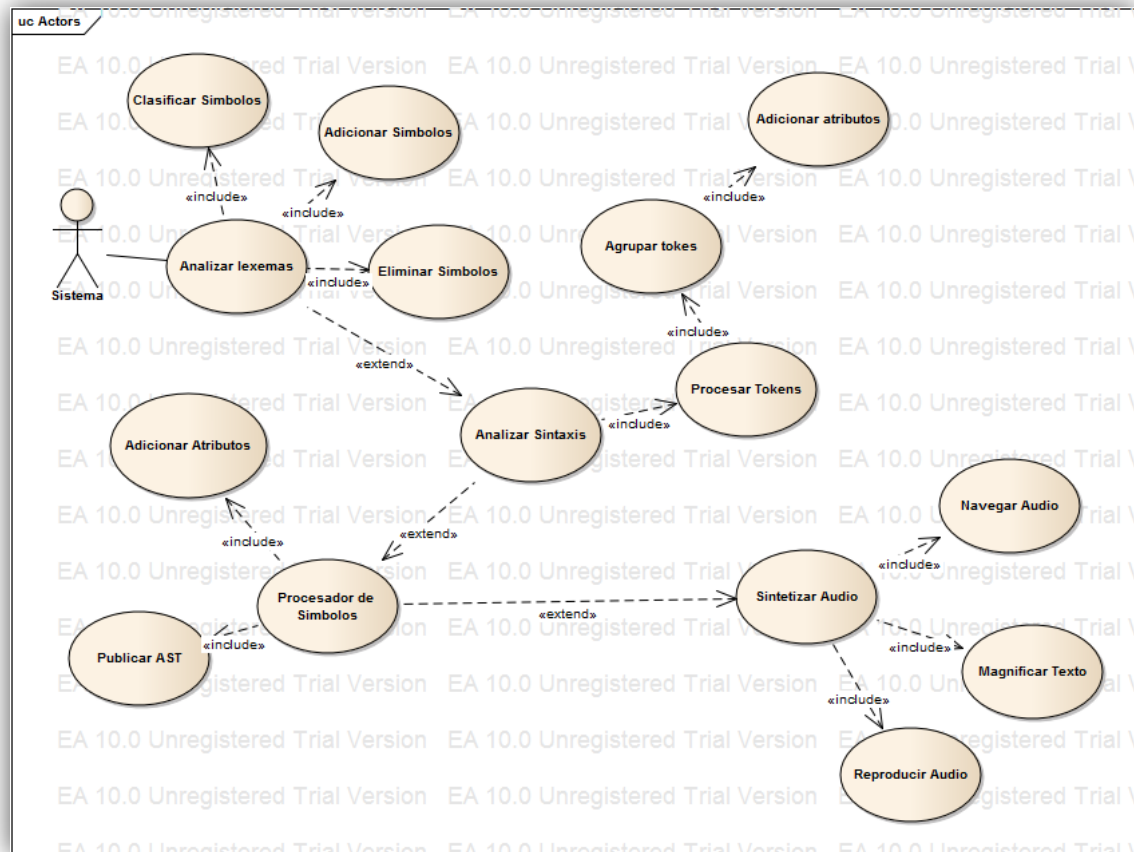


Figura 4.1.2 Casos de uso del modelo propuesto.

El diagrama de casos de uso desarrollado para el modelo propuesto, se base en las actividades realizadas por los diferentes módulos que actúan, con el fin de implementar las funciones básicas que describen la interacción entre el los diferentes módulos del software.

Las actividades que deben llevar a cabo van desde el análisis léxico con su respectiva clasificación de símbolos en tokens y eliminación de símbolos (espacios, tabulaciones y otros caracteres especiales no admitidos), análisis sintáctico, procesamiento de símbolos y sintetizador de voz.

En el modelo propuesto se pueden observar las relaciones de inclusión y generalización entre los diferentes módulos, esto facilita la comprensión del proyecto ya que de manera general el esquema presenta las funcionalidades que debería poseer cada módulo, y como esas funcionalidades interactúan entre sí.

De acuerdo a la anterior se puede visualizar por ejemplo, que al procesador de símbolos se le delega la tarea de adicionar al diccionario de datos la cadena de pronunciación para cada lexema, además de retornar cuando el analizador sintáctico lo requiera la información necesaria para la pronunciación del mismo.

### 4.3.4 Diagrama de Secuencia

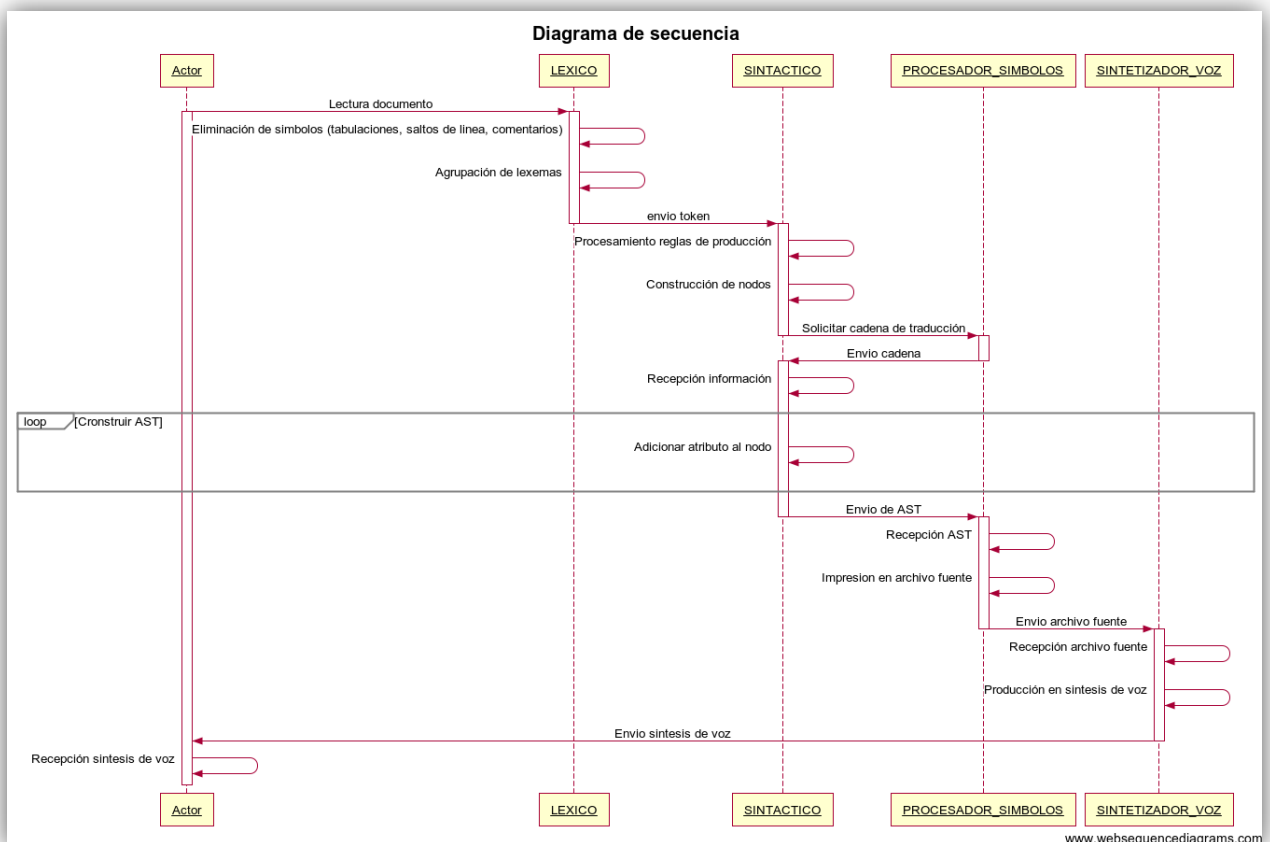


Figura 4.1.3 Diagrama de actividades del modelo propuesto.

El diagrama de secuencia desarrollado para el modelo propuesto, pretende modelar la interacción entre los componentes del software, detallando las diferentes actividades que ocurren al transcurrir el tiempo durante todo el proceso de compilación.

Para lograr una correcta sincronización entre los objetos encontrados en el diseño se realiza la especificación de los eventos necesarios para cumplir cada tarea, de esta forma mediante la línea de tiempo se va generando la interconexión entre los módulos para obtener la salida esperada.

Como se puede visualizar en el documento la fase inicial del proceso comienza cuando el componente **léxico** recibe el documento fuente (TeX) a analizar a través de un actor externo; este componente se encarga de la eliminación de símbolos (espacios, tabulaciones y otros caracteres especiales no admitidos) y la agrupación de los lexemas en tokens. Esta es una tarea importante del analizador léxico, debido a que se llevan a cabo una serie de eventos que procesan las expresiones, y las reestructuran de tal forma que pueda ser utilizada por los otros objetos, esto con el fin de adicionar a cada lexema la información relevante para continuar con las otras rutinas, en otras palabras en esta etapa del proceso lo que se lleva a cabo es un filtro del documento (TeX) con el fin de optimizar su uso en las actividades posteriores.

Al finalizar el proceso de análisis léxico el flujo obtenido, es la trama de tokens. Esta trama es recibida por el componente **sintáctico**, la cual es fundamental para continuar con el resto del proceso ya que mediante los tokens, y las reglas de producción se logra armar un sistema de nodos pertenecientes al árbol de sintaxis abstracto (AST). En una etapa intermedia de este proceso, el componente sintáctico solicita al componente de **procesamiento de símbolos**, cadenas de traducción para los diferentes nodos construidos, las cuales una vez recibidas se agregan a los nodos correspondientes en el árbol, finalizando así la construcción del AST.

Posteriormente el componente de **procesamiento de símbolos**, recibe el AST generado y lo imprime en un archivo fuente, el cual alimenta al componente **sintetizador de voz**, para su respectiva síntesis de voz, que entrega como salida esperada para el actor que inició el proceso.

#### 4.3.5 Prototipo

Para la presente investigación se realizó un prototipo del analizador léxico que reconoce expresiones de la forma:

- $$(a+b)^3=a^3+3a^2b+3ab^2+b^3$$

- $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
- $x^{2y} \neq (x^2)^y$
- $\int \frac{dx}{x} = \log x$
- $\frac{1 + \sqrt{5}}{2} = \phi$

El analizador léxico recibe como entrada esas expresiones y retorna cada símbolo con información adicional, como ejemplo se encuentra la siguiente expresión:

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

La salida del analizador para esta expresión es:

```

delimiter,'$$',1
lparen,'(',1
id,'a',1
plus,'+',1
id,'b',1
rparen,')',1
superscr,'^',1
inumber,'3',1
assign,'=',1
id,'a',1
superscr,'^',1
inumber,'3',1
plus,'+',1
inumber,'3',1
id,'a',1
superscr,'^',1
inumber,'2',1
id,'b',1
plus,'+',1

```

inumber,'3',1  
id,'ab',1  
superscr,'^',1  
inumber,'2',1  
plus,'+',1  
id,'b',1  
superscr,'^',1  
inumber,'3',1  
delimiter,'\$\$',1

El código para la creación del analizador se compone de:

1. **Palabras reservadas:** un conjunto de palabras reservadas y un diccionario de las mismas que contiene palabras propias del lenguaje como: over, neq, frac, int, sqrt.
2. **Tokens:** Símbolos más palabras reservadas como:
  - Operadores ordinarios: números (inumber, fnumber), identificadores (id).
  - Operadores relacionales: Comparadores (less, greater).
  - Delimitadores: Paréntesis (lparen, rpren), llaves (lbrace, rbrace).
3. **Definición de los Tokens:** Mediante expresiones regulares, utilizando la librería re de Python se definen todos los símbolos.
4. **Función principal:** En la función principal se realiza un llamado a otra función que lee las expresiones, y las envía como entrada al lexer.



## CAPITULO 5

### 5.1 CONCLUSIONES

- El estado de arte, permitió dar por hecho una de las hipótesis de este proyecto en la cual se planteaba, no existía una herramienta capaz de procesar expresiones construidas mediante el lenguaje de marcas LaTeX.
- El estado de arte permitió dar constancia de la gran cantidad de herramientas existentes para la síntesis de voz de expresiones matemáticas, y de las herramientas en proceso de desarrollo.
- El enfoque del proyecto AsTeR de Raman, permitió sentar las bases para la elaboración de este proyecto, también aplicaciones como ACTEMAT tuvieron incidencia en el diseño y en la implementación futura de nuevas mejoras para la herramienta.
- El levantamiento de los requerimientos permitió centralizar la información de los requisitos y componentes de software necesarios para la construcción del aplicativo.
- Gracias al levantamiento de requerimientos, se pudieron identificar las principales necesidades para realizar el diseño de la aplicación, además se presentó una base para una futura implementación tomando en cuenta los objetivos propuestos en la construcción de este proyecto.

## 5.2 BIBLIOGRAFIA

[1] Jacek Polewczak. LaTeX, MathML, Tex4th: Tools For Creating Accesible Documents. [En Línea]. Disponible desde: <http://goo.gl/F6z8bU>.

[2]. MathJax Consortium, American Mathematical Society (AMS) and the Society for Industrial and Applied Mathematics (SIAM). MathJax Beatifull Math in all Browsers. [En Línea]. Disponible desde: <http://goo.gl/5irUXZ>.

[3] MAA Mathematical Associative of America. Accesibility Project. [En Línea]. Disponible desde: <http://goo.gl/Dk0rge.sicu>

[4] JACEK POLEWCZAK LATEX, MATHML, AND TEX4HT: Tools For Creating Accesible Documents (A Brief Tutorial) [En Línea] Disponible desde: <http://goo.gl/IJQ0ou>

[5] Yod Samuel Martín García. ACTEMAT. Accesibilidad A Textos Matemáticos. [En Línea] Disponible desde: <http://web.dit.upm.es/~samuelm/ACTEMAT.pdf>

- [6] SICUMA. Sistemas de Información Cooperativos. Universidad de Málaga. Síntesis y Reconocimiento. [En Línea] Disponible desde: <http://goo.gl/vIAAnx>
- [7] Instituto Politécnico Nacional. Escuela Superior de Ingeniería Mecánica y Eléctrica. Accesibilidad a Expresiones Matemáticas Digitales. [En Línea] Disponible desde: <http://goo.gl/SbrBdc>
- [8] Albeiro Cuesta Meza. Modelo de Industrialización del Software en el Triángulo del Café (2010).
- [9] Helder Ferreira, Diamantino Freitas. Faculty of Engineering University of Porto. Audio Rendering of Mathematical Formula e using MathML and AudioMath.
- [10] Alvin Miguel Flores Bran. Alicia Ivette Salinas Benítez. Laura Yanira Salazar Calderón. Universidad Don Bosco. Facultad de Ingeniería. Desarrollo de una Herramienta de Accesibilidad del tipo Lector de Pantalla (2002).
- [11] Sandra B. Lucero. Universidad Nacional de Cuyo. Sistema Integrado de Documentación, SID Servicios Especiales para Personas con Discapacidad.
- [12] Mireia Ribera Turró. ONCE. Organización Nacional de Ciegos Españoles. Publicaciones científicas accesibles para personas ciegas y deficientes visuales (2010).
- [13] Jorge Bernal "Koke". Curso de LATEX. Fórmulas matemáticas (2003).
- [14] Offensive Security. Penetration Test Report fo Internal Lab and Exam v.1.0
- [15] Juan Carlos Yelmo García, Yod Samuel Martín García, José María del Álamo Ramiro. Casos de estudio sobre la aplicación de tecnologías de la ingeniería telemática al e-learning: federación de identidades y accesibilidad.
- [16] Carlos Arriola-Arciniega. Francisco J. Aceves. Instituto Politécnico Nacional MÉXICO. Herramienta auditiva para acceder expresiones matemáticas digitales. Científica, Vol.14 Núm. 3, pp. 137-144.
- [17] Carlos Arriola A., Francisco J. Aceves H. Instituto Politécnico Nacional Innovación Tecnológica para Acceder Expresiones. Matemáticas Digitales (2009).

[18] Ing. Jorge Plano. Facultad Regional Buenos Aires, Universidad Tecnológica Nacional. Por fin: matemáticas accesibles (2003).

[19] BaKoMa TeX is an innovative TeX/LaTeX implementation with *BaKoMa TeX Word - a Visual (True WYSIWYG) LaTeX Editor*. [En Línea] Disponible desde: <http://www.bakoma-tex.com/> (visitada febrero de 2013).

[20] Ed Gellenbeck Justyn Bell Ian Campbell Andreas Stefik. Speaking MathML: Using Prosody and Context-Sensitive Inferences to Produce Synthesized Speech.

[21] Instituto de Robótica, Universitat de Valencia. Esther Durá. Instituto de Robótica, Universitat de Valencia. Juan Domingo. Un sistema de ayuda a la escritura de textos estructurados en LaTeX para estudiantes con déficit visual (2011).

[22] MathJax: Beautiful math in all browsers. [En Línea] Disponible desde: <http://goo.gl/Zm45X3> (visitada febrero de 2013).

[23] Online LaTeX Editor. Quickly start using LaTeX and work together in real-time. [En Línea] Disponible desde: <https://www.sharelatex.com/>

[24] Kevin D. Smith. plasTeX - A Python Framework for Processing LaTeX Documents. [En Línea] Disponible desde: <http://goo.gl/Z4IY1w> (visitada junio de 2013).

[25] textHelp: SpeechStream Toolbal. [En Línea] Disponible desde: <http://goo.gl/nUwL1B> (visitada marzo de 2013).

[26] Gary Eason. Digital textbooks open a new chapter. [En Línea] Disponible desde: <http://goo.gl/7F1unW>.

[27] AWA Herramientas de utilidad en accesibilidad web. [En Línea] Disponible desde: <http://labda.inf.uc3m.es/awa/es/node/125> (visitada marzo de 2013).

[28] Marco's accessibility blog. [En Línea] Disponible desde: <http://goo.gl/kyMb>

[29] TeX/LaTeX to MathML Online Translator. [En Línea] Disponible desde: <http://goo.gl/Qa9I7>

- [30] EBU. The voice of blind and partially sighted people in Europe. [En Línea] Disponible desde: <http://goo.gl/N7bHvy>
- [31] David M. Beazley. PLY (Python Lex-Yacc) [En Línea] Disponible desde: <http://www.dabeaz.com/ply/ply.html>(visitada agosto de 2013)
- [32] AudioMath: Speaking Mathematics with MathML - PowerPoint PPT Presentation. [En Línea] Disponible desde: <http://goo.gl/COz0vw>
- [33] T. V. Raman. Audio System For Technical Readings. May 1994. [En Línea] Disponible desde: <http://goo.gl/KWR3Ea>
- [34] MIT Adaptive Technology Information Center (ATIC). [En Línea] Disponible desde: <http://goo.gl/OLAA1o>
- [35] Math Trax. National Aeronautics and Space Administration. [En Línea] Disponible desde: <http://goo.gl/DdFyaV>
- [36] Information Technology and Disabilities Journal. AsTeR: Audio System for Technical Readings. [En Línea] Disponible desde: <http://goo.gl/W3sYnM>
- [37] Sistemas de Información Cooperativos. Universidad de Málaga Java Speech Markup Language (JSML). [En Línea] Disponible desde: <http://goo.gl/Ao2LKD>(visitada mayo de 2013)
- [38] [AJO] AHO, SETHI, ULLMAN: Compiladores: Principios, técnicas y herramientas: Addison-Wesley Iberoamericana, 1990
- [39] [GARR] A. Garrido, J. Iñesta, F. Moreno y J. Pérez. 2002. Diseño de compiladores. Universidad de Alicante.
- [40] RasBodik, ThibaudHottelier, James Ide. Hack Your Language! Toward a Compiler Compiler: A recognizer. [En Línea] Disponible desde: <http://goo.gl/o670Jd>
- [41] FBU. Orca, el lector de pantallas libre. [En Línea] Disponible desde: <http://goo.gl/NYt6HX>.

[42] Universidad de Alicante. Sergio Luján Mora. [En Línea] Disponible desde:  
<http://goo.gl/uz4eR3>

[43] Andrew Roberts. Getting to Grips with LaTeX. [En Línea] Disponible desde:  
<http://goo.gl/ZN1FY>(visitada agosto de 2013)

[44] Kile. Texmaker. The universal LaTeX editor. (2003) [En Línea] Disponible desde:  
<http://www.xm1math.net/texmaker/> (visitada agosto de 2013)

[45] Universidad de Antioquia. Escuela Interamericana de Bibliotecología. Organización y Elaboración de proyectos de estudio de usuarios. [En Línea] Disponible desde:  
<http://goo.gl/Z49iFD>

## CAPITULO 6

### 6.1 ANEXOS

#### 6.1.1 SpeechStream Toolbar



Figura 6.1.1 Barra de herramientas para la lectura de contenidos HTML.

SpeechStream Toolbar es una barra de herramientas para la lectura de contenidos HTML, esta Toolbar permite leer, hacer pausas, y controlar la velocidad del contenido que está siendo transmitido. Además el SpeechStream Toolbar permite editar y crear reglas de pronunciación, y leer contenidos en diferentes idiomas.

Este aplicativo permite la lectura de documentos MathML usando el xhtml formato. Se puede realizar la integración con MathML haciendo uso del navegador Firefox, o en Internet Explorer instalado el plugin MathPlayer.<sup>18</sup>

#### 6.1.2 AudioMath

Esta herramienta es una de las soluciones más accesibles cuando se trata de leer contenido matemático, debido a que permite la visualización de contenido matemático en la web.

Las siguientes características influyeron en la implementación del aplicativo.

- Los textos científicos poseen mucho contenido matemático.
- La aparición de los sistemas TeX resolvían la mayoría de los problemas para la escritura de expresiones matemáticas.

---

<sup>18</sup> SpeechStream Toolbar. Disponible desde: <http://speechtoolbar.texthelp.com/DemoSpeechToolbar/default.htm>

- La aparición de editores WYSIWYG, y la aparición de los lenguajes de etiquetas como HTML.
- HTML no permite la inclusión directa de lenguaje matemático.

Bajo las anteriores circunstancias fue implementado AudioMath que permite la lectura del lenguaje MathML; de la misma forma, mediante la integración con otras herramientas para la conversión entre LaTeX / Tex y MathML puede leer el lenguaje LaTeX.

### 6.1.3 WinTriangle

Esta aplicación es un procesador de texto para invidentes, desarrollado por John Gardner en la universidad del estado de Oregon. Este programa permite a los invidentes o discapacitados visuales, leer y escribir expresiones matemáticas complejas.

Los usuarios de WinTriangle pueden convertir los archivos LaTeX a WinTriangle haciendo uso del plugin LaTeX2Tri. Después de haber realizado la conversión el resultado puede ser leído desde Word, siempre y cuando se instale en el Word las fuentes Triangle.ttf y MTEExtra.ttf.

WinTriangle es un software de código abierto, y se puede descargar del sitio oficial dirigido por el programa de acceso a la tecnología de la universidad estatal de Oregon.

### 6.1.4 MathTrax

Esta aplicación desarrollada por el National Aeronautics And Space Administration (NASA) provee herramientas para la descripción y reproducción de gráficos y simulaciones.

MathTrax es una herramienta gráfica para la escuela media y estudiantes de secundaria para graficar ecuaciones y simulaciones físicas. Los gráficos tienen descripciones y sonidos que ellos pueden escuchar y leer. Los invidentes y usuarios con discapacidad visual pueden acceder al gráfico o experimentar con ecuaciones y conjuntos de datos.<sup>19</sup>

---

<sup>19</sup> National Aeronautics and Space Administration. MathTrax. Disponible desde: <http://goo.gl/DdFyaV>



## 6.1.5 Eqneditor

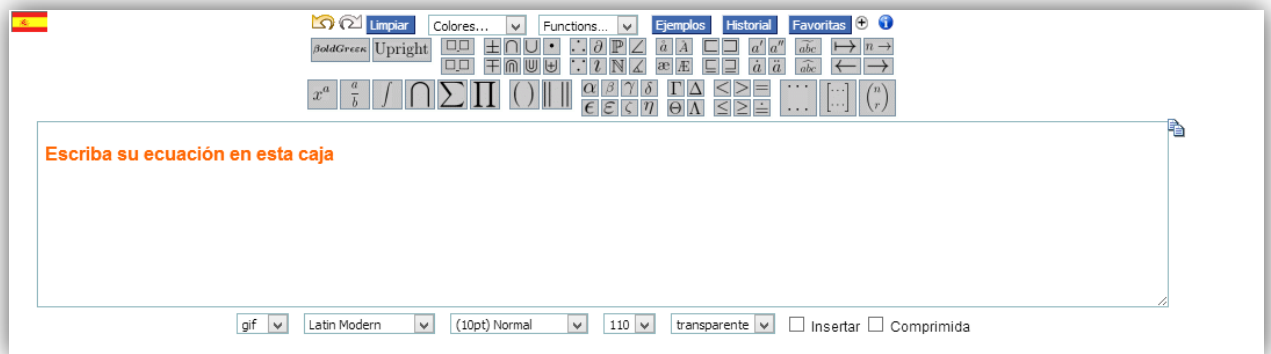


Figura 6.1.2 Editor online de fórmulas matemáticas.

Es un editor online, que permite escribir expresiones matemáticas, este editor da soporte para varios lenguajes entre ellos LaTeX, y permite exportar el código a MathML.<sup>20</sup>

## 6.1.6 Java Speech Markup Language (JSML)

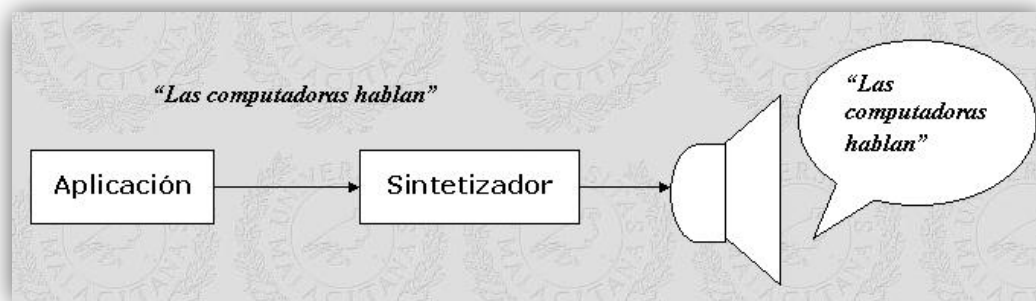


Figura 6.1.3 Diagrama de integración entre el lenguaje JSML y el sintetizador de voz.

JSML es un lenguaje de etiquetas que permite etiquetar las expresiones que ingresan al sintetizador de voz, de tal forma que suenen más similares a la voz humana. Las etiquetas agregadas son texto adicional que mejora la calidad y naturalidad de la salida

<sup>20</sup>Editor online de fórmulas matemáticas. Disponible desde: <http://www.codecogs.com/latex/eqneditor.php>

sintetizador.

El aplicativo funciona bajo XML, y posee un conjunto de elementos para etiquetar el texto, estos elementos son:

- Elementos estructurales para etiquetar párrafos y oraciones
- Elementos para controlar la producción del habla sintetizada, por ejemplo la pronunciación, el énfasis de las palabras, pausas, el control de tono.
- Elementos que representan elementos embebidos en el texto para habilitar controles específicos del sintetizador.<sup>21</sup>

Con todos estos elementos se puede reorganizar un texto, para hacer que la virtualización de la salida de audio del sintetizador, suene más como la voz humana.

### 6.1.7 ShareLaTeX

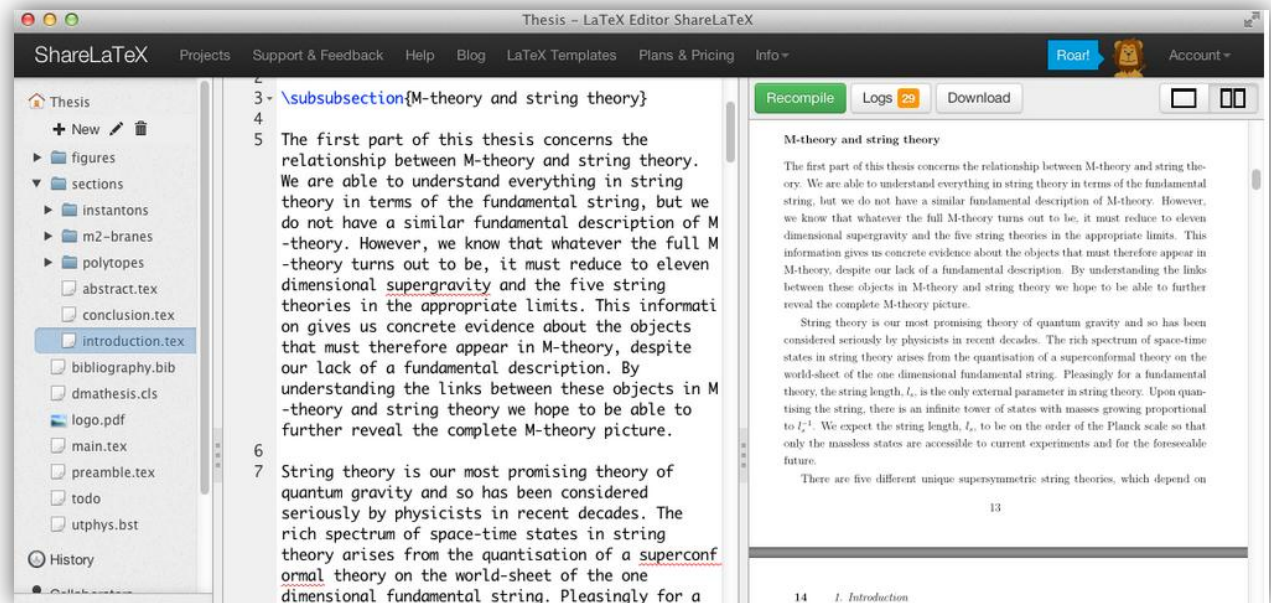


Figura 6.1.4 Editor online de fórmulas matemáticas.

<sup>21</sup>Java Speech Markup Language (JSML). Disponible desde: <http://goo.gl/Ao2LKD>

Esta aplicación es un editor de contenido matemático online para LaTeX, algunos de sus principales componentes son:

- LaTeX Wikibook: Es una herramienta para aprender LaTeX de forma rápida, con muchos ejemplos que se pueden reutilizar.
- WebEquation: Esta herramienta permite realizar dibujos de los símbolos matemáticos en la pantalla y el muestra el equivalente matemático.
- HowTeX: Es una colección de manuales y tutoriales para aprender LaTeX.<sup>22</sup>

---

<sup>22</sup>LaTeX Resources. Disponible desde: <https://www.sharelatex.com/resources>