



ESPE

**UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA**

**DEPARTAMENTO DE ELÉCTRICA Y
ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**TEMA: “DISEÑO E IMPLEMENTACIÓN DE UN PROGRAMA
INFORMÁTICO ORIENTADO A PERSONAS CON
DISCAPACIDAD VISUAL PARA EL APRENDIZAJE DE
ÁLGEBRA ELEMENTAL”**

AUTOR: SALCEDO GARCÉS, JAIME RICARDO

DIRECTOR: TIPÁN CONDOLO, EDGAR FERNANDO

SANGOLQUÍ

2016



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**DISEÑO E IMPLEMENTACIÓN DE UN PROGRAMA INFORMÁTICO ORIENTADO A PERSONAS CON DISCAPACIDAD VISUAL PARA EL APRENDIZAJE DE ÁLGEBRA ELEMENTAL**” realizado por el señor SALCEDO GARCÉS JAIME RICARDO, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor SALCEDO GARCÉS JAIME RICARDO para que lo sustente públicamente.

Sangolquí, 14 de marzo 2016

Una firma manuscrita en tinta azul que parece decir 'Edgar Fernando Tipán Condo'.

EDGAR FERNANDO TIPÁN CONDOLO
DIRECTOR



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Yo, **SALCEDO GARCÉS JAIME RICARDO**, con cédula de identidad N° 1718550864, declaro que este trabajo de titulación “**DISEÑO E IMPLEMENTACIÓN DE UN PROGRAMA INFORMÁTICO ORIENTADO A PERSONAS CON DISCAPACIDAD VISUAL PARA EL APRENDIZAJE DE ÁLGEBRA ELEMENTAL**” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 14 de marzo 2016

*Jv
Garcés*

SALCEDO GARCÉS JAIME RICARDO
C.C. 1718550864



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo, **SALCEDO GARCÉS JAIME RICARDO**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación “**DISEÑO E IMPLEMENTACIÓN DE UN PROGRAMA INFORMÁTICO ORIENTADO A PERSONAS CON DISCAPACIDAD VISUAL PARA EL APRENDIZAJE DE ÁLGEBRA ELEMENTAL**” cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 14 de marzo 2016

Jv the Znpdnu

SALCEDO GARCÉS JAIME RICARDO
C.C. 1718550864

Dedicatoria

A mis padres, por su amor y ejemplo en todo momento.

A mis hermanos, por su cariño y amistad.

Agradecimientos

Ing. Edgar Tipán, por dedicar tiempo valioso a transmitir sus conocimientos y ser la persona clave para el desarrollo de esta serie de proyectos sociales.

Ing. Julio Larco, por su visión acertada de las necesidades de las personas no videntes en el desarrollo del proyecto.

Lic. Lilia Tipán, por su dedicado esfuerzo en la búsqueda de una educación con accesibilidad para las personas no videntes y su hijo.

Lic. Wladimir Ganchala y Dr. José Benavides, al dar su tiempo y conocimientos para probar este proyecto.

Índice general

1. Generalidades	1
1.1. Antecedentes	1
1.2. Justificación e Importancia	3
1.3. Alcance	4
1.4. Objetivos	5
1.4.1. General	5
1.4.2. Específicos	5
2. Marco Teórico	6
2.1. Discapacidades Visuales	6
2.2. El Sistema Braille	7
2.3. No videntes y las matemáticas	8
2.4. Representaciones	9
2.4.1. Estructuras de árboles	9
2.4.2. Visual	12
2.4.3. Verbal	14
2.4.4. Táctil	17
3. Diseño del Software	18
3.1. Necesidades del Usuario	18
3.2. Interfaz Gráfica de Usuario	19
3.2.1. Arquitectura y Navegación	20
3.2.2. Distribución	21
3.2.3. Tipografía	27
3.2.4. Color	29
3.2.5. Contenidos	29
3.2.6. Presentación de la información	31
3.2.7. Accesibilidad	47
3.3. Algoritmos	53
3.3.1. Diagramas de flujo	53
3.3.2. Diagramas UML	63

4. Implementación	68
4.1. Programación de la interfaz de usuario	68
4.2. Programación de los algoritmos	71
5. Análisis de Resultados	84
5.1. Pruebas	84
5.1.1. Pruebas de la interfaz	84
5.1.2. Pruebas con ejemplos	86
5.2. Resultados	87
6. Conclusiones y Recomendaciones	91
6.1. Conclusiones	91
6.2. Recomendaciones	92
6.3. Trabajos futuros	93

Índice de tablas

1. Abecedario en braille de 6 puntos	8
2. Números en Braille	8
3. Representaciones	9
4. Ejemplo con XML	12
5. Sintetizadores de voz comerciales	14
6. Componentes de la ventana de presentación	22
7. Barra de menús de la ventana del editor	22
8. Barra de menús de la ventana del editor	23
9. Barra de herramientas de la ventana del editor	25
10. Barra de Información de la ventana del editor	25
11. Componentes de las ventanas abrir y guardar	26
12. Componentes de la ventana imprimir	26
13. Características del monitor	27
14. Tipografía 1	28
15. Tipografía 2	28
16. Tipografía 3	28
17. Paleta de colores	29
18. Operaciones y Funciones	29
19. Símbolos	30
20. Estructuras clasificadas por número de parámetros	31
21. Etiquetas MathML de las estructuras matemáticas	33
22. Etiquetas MathML para los parámetros de las estructuras matemáticas	34
23. Eco para cada tecla imprimible	37
24. Eco para cada tecla de función	42
25. Eco para caracteres braille	43
26. Salida braille modelo de las estructuras matemáticas	45
27. Salida audiotexto modelo de las estructuras matemáticas	46
28. Funciones del panel de entrada de texto	47
29. Funciones del panel braille	48
30. Funciones del panel de entrada de expresiones algebraicas	49
31. Funciones del panel de audiotexto	50
32. Funciones del panel de renderizado	51
33. Comandos rápidos por teclado por defecto	52

34.	Comandos rápidos por teclado nuevos	52
35.	Herramientas de Software	68
36.	Ejercicios	86
37.	Duración, número de operaciones y velocidad	87
38.	Complejidad ejercicios 1-4	88
39.	Complejidad ejercicios 5-8	88

Índice de figuras

1.	Clasificación de las discapacidades visuales	6
2.	Árbol semántico	11
3.	Funcionamiento general del programa	19
4.	Arquitectura del programa	21
5.	Diagrama de la navegación entre ventanas	21
6.	Bosquejo de la ventana de presentación	22
7.	Bosquejo de la ventana del editor	23
8.	Bosquejo de la ventana de abrir/guardar	26
9.	Bosquejo de la ventana de imprimir	27
10.	Plantilla de una operación y una función	32
11.	Renderización de una operación	34
12.	Renderización de una función	34
13.	Renderización de una operación anidada	35
14.	Renderización de una fracción	35
15.	Renderización de un logaritmo	35
16.	Renderización de una potencia	36
17.	Renderización de una raíz	36
18.	Funcionamiento general del proceso de renderización	54
19.	Proceso de exportación	54
20.	Proceso de linealización	55
21.	Proceso de reordenamiento entre-orden	56
22.	Proceso de reordenamiento pre-orden	57
23.	Creación y configuración del archivo de salida	57
24.	Preparación de los mensajes de voz	58
25.	Funcionamiento general de conversión a audiotexto	59
26.	Funcionamiento general de conversión a braille	59
27.	Proceso de adaptación y traducción	60
28.	Proceso de traducción N1	60
29.	Proceso de traducción N2	61
30.	Proceso de traducción N3	61
31.	Proceso de traducción N4	62
32.	Proceso de traducción N5	62
33.	Proceso de traducción N6	63

34.	Diagrama de la clase Estructura	64
35.	Diagrama de la clase Elemento	64
36.	Diagrama de la clase Caracter	64
37.	Diagrama de la clase Abrir	65
38.	Diagrama de la clase Guardar	65
39.	Diagrama de la clase Imprimir	65
40.	Diagrama de la clase Usuario	66
41.	Diagrama de la clase Audio	67
42.	Diagrama de la clase Principal	67
43.	Ventana del editor implementada	69
44.	Ventana abrir implementada	69
45.	Ventana guardar implementada	70
46.	Ventana imprimir implementada	70
47.	Gráfico Parámetros por Ejercicio	89

RESUMEN

El programa informático de aprendizaje de álgebra elemental está orientado a personas con discapacidad visual total o parcial que se encuentren estudiando en secundaria o bachillerato; permitiendo a estos usuarios ingresar, editar y modificar expresiones algebraicas usando un computador a través de estructuras de árboles de datos, comandos por teclado, mensajes de voz y un menú con una lista de las funciones u operaciones matemáticas soportadas, es decir es una herramienta para la edición de expresiones algebraicas de manera accesible y no realiza ningún cálculo matemático. El programa fue desarrollado en Visual Studio 2010 Express y funciona en los sistemas operativos desde Windows 7 o posteriores. El mismo que se caracteriza por convertir las fórmulas matemáticas ingresadas a representaciones de forma renderizada, braille y verbalizada usando el lenguaje de marcas MathML, el código matemático unificado con el sistema braille de seis puntos y un sintetizador de voz que soporta el idioma español respectivamente. El programa beneficia a los no videntes primero al reducir la carga mental y la limitación de ubicarse dentro de las expresiones matemáticas en dos dimensiones, segundo al ganar independencia de una persona que supervise contantemente su interacción con el computador o sus actividades escolares y tercero al poder intercambiar documentos que contengan fórmulas matemáticas con personas que tengan poco o ningún conocimiento del sistema braille.

Palabras clave:

- **MENSAJES DE VOZ**
- **RENDERIZADA**
- **MATHML**
- **BRAILLE**
- **SINTETIZADOR DE VOZ**

ABSTRACT

The program of learning elementary algebra is aimed at people with total or partial visual impairment who are studying in school and high school; it allows to these users to enter, edit and modify algebraic expressions using a computer through tree data structures, keyboard shortcuts, voice messages and a menu with a list of mathematical supported operations or functions, in summary this is a software tool for editing algebraic expressions in an accessible way and it does not perform any mathematical calculation. It was developed on Visual Studio 2010 Express and it runs on operating systems since Windows 7 or later. The same one is characterized by converting mathematical formulas entered to rendering, braille and text to speech representations using the marks language MathML, mathematical code unified with six points braille and a speech synthesizer that supports the Spanish language respectively. The program is beneficial for the blind people; firstly to reduce the mental load and limiting to be within mathematical expressions in two dimensions, secondly to gain independence from a person who is steady supervise their interaction with the computer or their school activities and thirdly to be able to exchange documents containing mathematical formulas with people who have low or any knowledge of braille.

Keywords:

- **VOICE MESSAGE**
- **RENDER**
- **MATHML**
- **BRAILLE**
- **SPEECH SYNTHESIZER**

Capítulo 1

Generalidades

1.1 Antecedentes

La tiflotecnología es una tecnología de apoyo aplicada a personas ciegas o con baja visión, cuyo objetivo es dar un aprovechamiento práctico de las herramientas y técnicas de la tecnología con el objetivo de mejorar su calidad de vida.

Una muestra de dicho aprovechamiento es el sistema braille, con el cual se puede representar cualquier carácter de un alfabeto, símbolo o número a fin de transmitir sus ideas de manera textual. Este es un sistema donde cada carácter codificado utiliza un arreglo de seis puntos ordenados en tres filas y dos columnas, estos puntos pueden estar o no en relieve. Por medio del sentido del tacto específicamente usando las yemas de los dedos de las manos la persona no vidente es capaz de reconocer la presencia de puntos en alto relieve sobre una superficie plana para determinar de qué carácter se trata.

Así mismo en la Universidad de las Fuerzas Armadas - ESPE se han realizado proyectos sociales en este campo, como en el año 2014 el proyecto de fin de carrera titulado: “Diseño e implementación de un sistema electrónico con interface a PC para automatizar una máquina de escribir de braille” [1], mediante el cual se desarrolló una impresora braille en tres etapas: La primera etapa del proyecto consistía en adaptar actuadores eléctricos a una máquina de escribir de braille (máquina de Perkins, similar a una máquina de escribir común) conjuntamente con un controlador electrónico para los mismos. La segunda etapa consistió en realizar un software que permitiera traducir los caracteres del alfabeto latino y números decimales al sistema de codificación braille. La etapa final fue juntar tanto el hardware como el software creado para que funcionara como una impresora braille comercial.

La funcionalidad de este sistema permite imprimir únicamente la simbología utilizada para el alfabeto latino, signos de puntuación, exclamación, interrogación y números decimales, sin embargo cabe destacar que el proyecto no abarca toda la simbología que el sistema braille permite, por ejemplo el sistema braille posee además un estándar para notación musical, notación química y particularmente para notación matemática. Por citar algunos ejemplos de uso de notaciones matemáticas adoptadas en distintas regiones del mundo, que según el artículo [2], se tiene los siguientes estándares:

- *Nemeth Mathematical Code N.M.C.*, usado en Estados Unidos.
- *Royal National Institute of Blind People R.N.I.B.*, usado en Reino Unido.
- *Marburg*, usado en Alemania, Austria y Polonia.
- Código Matemático Unificado C.M.U., usado en el Ecuador y Latinoamérica.

Por otro lado, algunos programas informáticos para el aprendizaje de las matemáticas orientados a personas con discapacidad visual, utilizan los estándares antes mencionados para la edición de textos con fórmulas matemáticas:

- *MAVIS* [3] y *LaBraDoor* [4]: son programas que traducen ecuaciones al braille, proveen salida de audio mediante tonos para objetos de dos dimensiones generados por estructuras jerarquizadas, requieren licencia pagada, no están en idioma español y no utilizan el CMU.
- *AudioMath* [5]: es un editor de ecuaciones, convierte expresiones algebraicas en notación matemática a una versión hablada de la misma pero en idioma portugués, se desconoce que si utiliza o no CMU, se encuentra en desarrollo.
- Editor Lambda [6]: es un potente editor de ecuaciones razón por la cual es adoptado por la ONCE (Organización Nacional de Ciegos Españoles), utiliza una codificación braille no estándar basada 8 puntos pero utiliza CMU, no genera estructuras jerarquizadas para despliegue de información, requiere licencia pagada.
- Existen impresoras braille comerciales [7], con su software único para el computador, con un costo desde 3600 USD (sin contar con cargos de importación), este precio es justificado debido a la velocidad que tienen al imprimir alrededor de 300 páginas por hora y por su versatilidad para imprimir tanto caracteres como imágenes en distintos grados de relieve pero no posee un editor de fórmulas matemáticas.

Las características de estos programas no cumplen con todos los requerimientos necesarios para tener acogida dentro de los sistemas educativos porque en primer lugar, es importante que se utilice el código matemático unificado (con braille de 6 puntos) debido que al usar otro estándar causaría una interpretación errónea de la información. En segundo lugar, deben ser de libre acceso para personas con discapacidad o al menos con licencias pagadas. En tercer lugar, el idioma del programa debe ser el español. Y por último, debe tener opciones de accesibilidad para permitir la interacción con el usuario no vidente.

1.2 Justificación e Importancia

La Constitución de la República del Ecuador en el título II, capítulo tercero, sección sexta, apartado 7, reconoce el derecho de las personas con discapacidad a: “Una educación que desarrolle sus potencialidades y habilidades para su integración y participación *en igualdad de condiciones*. Se garantizará su educación dentro de la educación regular. Los planteles regulares incorporarán trato diferenciado y los de atención especial la educación especializada. Los establecimientos educativos *cumplirán normas de accesibilidad para personas con discapacidad* e implementarán un sistema de becas que responda a las condiciones económicas de este grupo.”[8]

Sin embargo, un informe del Consejo de Regulación y Desarrollo de la Información y Comunicación dice: “en 2014 se presentaron 2.018 personas con discapacidad a rendir pruebas en el SNNA (Sistema Nacional de Nivelación y Admisión), superando lo conseguido en 2013 cuando 865 personas se inscribieron. Del total de inscritos en 2014, un aproximado de mil personas con discapacidad ingresarán a distintas universidades a nivel nacional” [9]. Teniendo en cuenta que el número total de personas con discapacidad visual en Ecuador es 46.435 [10], en “el país existen 27.359 personas con ceguera, de las cuales 3.200 están en edad de escolarización, es decir entre 5 y 19 años, según los datos recogidos por “Manuela Espejo” en 2009.” [11].

Lo que demuestra que existe una creciente cantidad de personas con discapacidad visual que buscan acceder a los sistemas educativos del país, los mismos que no cuentan con los recursos apropiados (o son inexistentes) en cuanto a herramientas para la educación y aprendizaje, que ocasiona inconvenientes en la educación básica, media y superior relegando esta responsabilidad a organizaciones particulares de no videntes como FENCE (Federación Nacional de Ciegos del Ecuador) una organización sin fines de lucro encargada de brindar capacitación especializada para personas no videntes, la misma que tiene interés en el desarrollo de proyectos de inclusión social.

Por otro lado, en cuanto uso del sistema braille en las matemáticas presenta ciertos inconvenientes como:

- *Unidimensionalidad*: al escribir en braille el proceso se realiza de manera plana es decir, como si cada carácter debe ubicarse estrictamente dentro de un renglón a diferencia de escribir en “tinta” donde el proceso es más versátil pudiendo escribir en cualquier lugar dentro y fuera del renglón.
- *Tamaño*: los textos en braille ocupan más espacio físico por lo que el tamaño de los textos escolares poseen más páginas, hojas, volúmenes, etc.

- *Carga cognitiva*: los dos puntos anteriores provocan que la cantidad de información que debe interpretar una persona al usar el sistema braille para matemáticas sea tal que necesite un mayores esfuerzos para retenerla.

Por lo tanto, el desarrollo de un programa informático para creación y edición de fórmulas matemáticas permitirá tener una mejor interacción entre docentes y sus alumnos con deficiencia visual de nivel medio y bachillerato al momento de elaborar, corregir y evaluar las actividades escolares en la asignatura tanto en deberes como en evaluaciones, de tal manera que los mismos estudiantes puedan realizar sus actividades escolares sin utilizar más tiempo del necesario y los docentes puedan supervisar y evaluar las actividades realizadas por los estudiantes con deficiencia visual además de crear material didáctico para matemáticas, por ejemplo permitiría al docente (que tenga un conocimiento básico del sistema braille) el plantear evaluaciones a los estudiantes no videntes.

El objetivo de dicho programa es reducir el grado de dependencia de las personas no videntes y las limitaciones que contrae la notación matemática para este grupo de la población. En braille existe todo un estándar para la representación de expresiones algebraicas elementales [12], que mediante este programa permitirá facilitar la labor del aprendizaje para los estudiantes no videntes en la asignatura de matemáticas diferenciándose de otros proyectos porque permitirá crear un software en idioma español con opción de audio para escuchar las expresiones algebraicas ingresadas por el usuario, utilizando el estándar internacional CMU.

1.3 Alcance

Este proyecto esta centrado en crear un programa informático pensado para estudiantes de nivel medio y bachillerato con discapacidad visual para ingresar, editar e imprimir expresiones algebraicas elementales, usando el sistema operativo *Windows 7* o posteriores. El procesamiento que realiza este programa para las expresiones algebraicas ingresadas consiste en las etapas que se describen a continuación:

- Ingreso: en esta etapa el usuario ingresa de manera guiada o asistida por medio del teclado del computador la información (constantes, variables y operadores) para formar expresiones algebraicas.
- Interacción: la información ingresada es convertida a representaciones visuales, audibles y táctiles (braille) para que el usuario pueda revisar, corregir y modificar según se requiera.

- Almacenamiento: el usuario almacena su información en un archivo digital dentro del computador para un posterior uso.

El programa consta de las siguientes características:

- Asistir al usuario en el momento de ingresar, editar o eliminar símbolos y estructuras matemáticas, editando estos componentes para generar expresiones algebraicas elementales.
- Presentar la información de forma audible, mediante un sintetizador de voz en idioma español. De forma impresa, mediante un prototipo de impresora braille utilizando CMU. De forma visual, mediante archivos de imágenes generados en base a un lenguaje de marcado.
- Convertir tanto textos como expresiones algebraicas al sistema braille CMU de 6 puntos.

1.4 Objetivos

1.4.1 General

Diseñar e implementar un programa informático orientado a personas con discapacidad visual usando herramientas de accesibilidad auditiva para ingresar, editar e imprimir expresiones algebraicas elementales.

1.4.2 Específicos

- Realizar el estado del arte de aplicaciones que permitan la conversión de expresiones algebraicas elementales a braille.
- Organizar las expresiones algebraicas elementales ingresadas por el usuario mediante un asistente de edición.
- Presentar la información previamente procesada de forma auditiva, visual y en braille para interactuar con el usuario.
- Evaluar el funcionamiento del programa creado mediante pruebas con personas con discapacidad visual para verificar el desempeño del programa.
- Documentar el proyecto organizando la información generada con diseños, algoritmos, búsquedas en un archivo entregable.

Capítulo 2

Marco Teórico

En este capítulo se expondrán conceptos y herramientas relacionados con tecnologías de accesibilidad que ayudan a entender el funcionamiento y desarrollo del *software*.

2.1 Discapacidades Visuales

El término “discapacidad” significa “falta o limitación de alguna facultad física o mental que imposibilita o dificulta el desarrollo normal de la actividad de una persona.” [13], en el caso de una discapacidad visual la limitación física es cuando los órganos de la visión han sido afectados.

La pérdida de la vista por más mínima afecta directa e indirectamente a todas las actividades cotidianas y su relación con el entorno, por lo cual se considera un daño “severo” aún si es temporal. Según la Organización Mundial de la Salud [14], las discapacidades visuales se clasifican como muestra en la Figura 1.

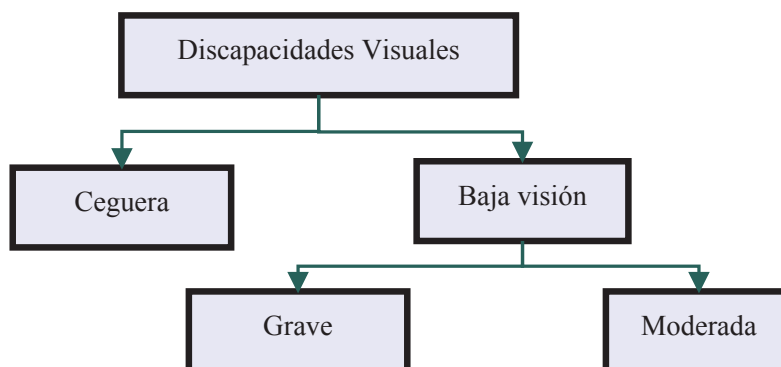


Figura 1 Clasificación de las discapacidades visuales

Detallando cada una de las mismas, se puede notar que existen diferentes grados de discapacidad visual, como:

Baja visión grave, se refiere al afectamiento del sentido de la vista que no pueda ser corregida totalmente mediante lentes, gafas, tratamiento o cirugía, y el porcentaje de recuperación de la visión con ayuda se encuentra por debajo del 50%.

Baja visión moderada, se refiere al mismo afectamiento pero con el porcentaje de recuperación por encima del 50%.

Ceguera, se refiere al afectamiento total del sentido de la vista que impide la respuesta ante estímulos de luz.

Sin importar el grado de afectación, la vista constituye un factor importante dentro del proceso de aprendizaje, razón por la cual, han llegado a crearse tecnologías pensadas en la accesibilidad, es decir, que buscan el acceso a la información para cualquier persona sin importar si tiene una discapacidad visual o no. Pero debido a factores económicos y sociales, la problemática que se presenta para las personas con ceguera y baja visión se presenta en:

- Adquirir o acceder a literatura en braille o en audio, pues la necesidad de buscar información, entretenimiento, educación o cultura, disminuye su capacidad de integración y formación personal.
- Adquirir sistemas de escritura, como las máquinas de escritura braille, impresoras braille o hasta una simple regleta y un punzón.
- Adquirir material especialmente adaptado para la interacción con personas con discapacidad visual, es decir que tengan acceso a información con herramientas de accesibilidad como señales auditivas o en sistema braille.

Debido a estos factores, la educación para una persona con discapacidad visual es un verdadero reto, lo que ha obligado en algunos casos a dejar de lado la formación académica.

2.2 El Sistema Braille

Este sistema utilizan las personas con discapacidad visual por medio el sentido del tacto usando las yemas de los dedos, tiene sus ventajas debido a su simplicidad, y es útil para interpretar y representar sus ideas a fin de interactuar en su entorno.

El braille es un sistema de caracteres simplificado, cuya característica más notable es el relieve que se percibe cuando esta impreso en una superficie (Ver Tabla 1), cada uno de estos caracteres se le conoce como “signo generador” y están conformados por un conjunto de puntos agrupados en filas y columnas. Existen dos estándares: braille de 6 puntos y braille de 8 puntos, la diferencia entre ambos consiste en la cantidad de combinaciones que se pueden obtener para representar más caracteres, 64 y 256 combinaciones respectivamente.

Tabla 1**Abecedario en braille de 6 puntos**

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p
q	r	s	t
u	v	w	x
	y	z	

2.3 No videntes y las matemáticas

Poder leer, escribir o modificar una expresión matemática usando la notación común constituye un verdadero reto para los no videntes, pues se debe comprender que la posición de cada elemento dentro de una expresión matemática puede estar ubicado a lo largo o ancho (2 dimensiones) de una superficie, por lo que daría lugar a algunas confusiones. Considerando también que cuando se trata de expresiones matemáticas más extensas se genera una mayor carga mental, es decir tener que recordar dónde y qué es el elemento de una fórmula es un verdadero dolor de cabeza.

Para tratar de solventar estos problemas se ha creado notaciones matemáticas simplificadas para no videntes (1 dimensión) usando el braille así como también las asociaciones de ciegos de América Latina en el año de 1987 en la ciudad de Montevideo en Uruguay aprobaron la utilización (para los países latinoamericanos) del Código Matemático Unificado (C.M.U.), que es un estándar para la estructura, notación y terminología de las matemáticas en braille que actualmente se utiliza en centros de educativos e imprentas (ver Tabla 2). Más adelante se hablará en detalle de la notación C.M.U.

Tabla 2**Números en Braille**

0	1	2	3	4
5	6	7	8	9

2.4 Representaciones

En matemáticas la manera de comunicar o transmitir una idea para que otra persona pueda interpretarla es un factor determinante; en el caso de las expresiones algebraicas es necesario plasmar estas ideas mediante representaciones ya sean de manera escrita, visual, auditiva o hasta de manera táctil.

Es importante recalcar que dichas representaciones de las expresiones algebraicas no se expresan de la misma manera, cada una tiene sus características peculiares pero a fin de cuentas el objeto de las distintas representaciones es llegar a la misma idea. En la Tabla 3 se muestra lo que se verá en este capítulo de manera general, así como las herramientas de software para matemáticas dedicadas para personas con discapacidad visual.

Tabla 3
Representaciones

Representación	Sistema	Ejemplo
Especial	Estructura de árbol	<pre> graph TD A[Fracción (/)] --> B[Suma (+)] A --> C[Resta (-)] B --> D[x] B --> E[1] C --> F[x] C --> G[1] </pre>
Visual	Renderización gráfica	$\frac{x+1}{x-1}$
Auditiva	Conversión audiotexto	“Fracción, el numerador es: x más 1 y el denominador es: x menos 1”
Táctil	Sistema braille	 $(x + 1)/(x - 1)$

2.4.1 Estructuras de árboles

La primera representación llamada “especial” se trata de árboles, que es una estructura de datos que toma el modelo de los árboles en la naturaleza, conformados por elementos denominados nodos (similar a las “ramas”) en donde se almacena por cada nodo un dato (similar a las “hojas”), también tiene la capacidad de extenderse indefinidamente cuando un nodo tiene uno o más nodos hijos (similar a las “ramificaciones”).

Los árboles de datos tienen peculiaridad de que únicamente poseen un nodo raíz del que pueden derivarse los demás nodos [15].

Algunas de las ventajas al utilizar estructuras tipo árboles son:

- La información se almacena de manera organizada y jerárquica.
- Ideal para expresiones matemáticas.
- Facilidad de procesamiento por computador.
- Facilidad de desplazamiento entre los nodos.

Las desventajas que se presentan en cambio son:

- Se requiere usar técnicas recursivas para su procesamiento; que en grandes tamaños podría desbordar la pila del sistema.
- No existe un método único para recorrer cada nodo del árbol.

Desde el punto de vista de los seres humanos se puede comprender las relaciones entre los elementos de una operación, función, así como la relación entre una o varias operaciones y funciones al utilizar las estructuras tipo árbol, por así decirlo es una manera ordenada por jerarquía, útil al examinar una expresión algebraica [16] pero de gran importancia al aplicar este concepto cuando se tiene una discapacidad visual. Esto ayuda sustancialmente para tener una *guía* de la estructura de una expresión algebraica en comparación a no tener ninguna referencia. Y es una solución utilizada en programas informáticos de matemáticas para no videntes como: MAVIS [3] y Audio-Math [5], en donde se permite al usuario no vidente desplazarse para examinar una expresión matemática por medio de los nodos de los árboles facilitando el ingreso y modificación de los datos de una manera estructurada.

Según una publicación relacionada [17], se muestra como identificar las expresiones matemáticas mediante reconocimiento computarizado de imágenes, se exponen dos modelos para el procesamiento de las expresiones mediante las estructuras de datos llamadas árboles:

- **Árbol de distribución:** representa la información relacionada con la distribución y la geometría de los símbolos y caracteres de una fórmula. *Por ejemplo*, con esta estructura de datos la expresión algebraica conocida como fórmula general para determinar las raíces de un polinomio de segundo grado identificando sus

variables (en color celeste), constantes (en color violeta) y operadores (en color naranja), se obtendría:

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

- **Árbol semántico:** representa la información relacionada con el contenido y relación de una fórmula. *Por ejemplo*, con esta estructura de datos la fórmula anterior (con los mismos colores) se obtendría la Figura 2.

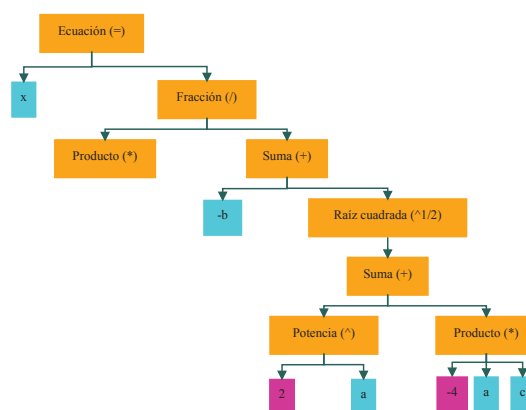


Figura 2 Árbol semántico

Al usar las estructuras de árboles se puede realizar transformaciones, es decir poder reordenar lineal y no linealmente estas estructuras para obtener otras representaciones de las expresiones algebraicas como son: renderizadas, verbalizadas y en braille. Tal ventaja es gracias a los lenguajes de marcado, que son sistemas diseñados para que los computadores puedan manipular, procesar e intercambiar información sin causar confusiones o pérdidas, con este propósito fue creado por el consorcio *World Wide Web* como un estándar. Este sistema se caracteriza por el uso de anotaciones o “marcas” para organizar información de una manera simple e interpretar de una forma mecanizada, un tipo de estos lenguajes es el “Lenguaje de Marcas Extensible - XML”. Por ejemplo se clasifica a los integrantes de una familia mediante etiquetas para reconocer a cada miembro y mediante separaciones con espacios horizontales para reconocer las relaciones entre los mismos en la Tabla 4.

Los lenguajes de marcado abstraen el concepto de las estructuras de datos tipo árboles, y su principal ventaja es no depender de una plataforma, lo que permite que los navegadores web de cualquier dispositivo procesen archivos que contengan información organizada con este sistema.

Tabla 4**Ejemplo con XML**

Código
<pre> <familia> <padre> Luis </padre> <madre> Isabel </madre> <hija> Emilia </hijo> <hijo> Carlos </hija> </familia> </pre>

2.4.2 Visual

La segunda representación “render”, es un anglicismo que se refiere al proceso de convertir las expresiones algebraicas a la notación común de las matemáticas. En la primera fila de la columna “Ejemplo” de la Tabla 3 se puede identificar cuáles son las operaciones y sus componentes, esta es la representación más utilizada debido a que la ubicación geométrica de los elementos de una expresión tienen significado y permiten entender de una manera visual las relaciones entre dichos elementos.

Un lenguaje de marcado que permite el procesamiento, renderizado y almacenamiento de expresiones matemáticas es denominado MathML, el cual se especializa en notación matemática [18], y es ampliamente utilizado en programas informáticos de cálculo matemático para procesamiento y renderización.

Ejemplo:

Una expresión algebraica bidimensional sería:

$$\frac{1-x}{1+x}$$

Representada la misma expresión en MathML se tendría mediante este lenguaje de marcas organizadamente toda la expresión algebraica anterior mediante un nodo principal que lo llama “math”, un nodo hijo llamado “mfrac” para referirse a la estructura de fracción, este a su vez utiliza dos nodos hijos denominados “mrow” el primero para referirse al numerador de la fracción y el segundo para referirse al denominador de la

fracción, los mismos que cuentan con los elementos “mn”, “mo” y “mi” para referirse por separado a las constantes, operadores y variables individualmente.

```

<math>
  <mfrac>
    <mrow>
      <mn>1</mn>
      <mo>-</mo>
      <mi>x</mi>
    </mrow>
    <mrow>
      <mn>1</mn>
      <mo>+</mo>
      <mi>x</mi>
    </mrow>
  </mfrac>
</math>

```

Paralelamente mediante tabulaciones horizontales se establece la jerarquía de las operaciones que para este caso en primer lugar se encuentra la fracción y luego las operaciones aritméticas.

Actualmente solo el navegador “Mozilla Firefox” desde la versión 31 tiene un soporte estable para *MathML* según un estudio realizado [19], al probar el resultado de la renderización de expresiones matemáticas complejas con navegadores web. Aunque también existen diversas aplicaciones y librerías ya implementadas que pueden ser integradas por otros programas para escribir y visualizar expresiones matemáticas de manera renderizada usando *MathML*, se tiene entre las más destacadas:

- *MathJax* [20]: es una biblioteca hecha en *javascript* interpreta tanto *MathML* como *LaTeX* en los navegadores web.
- *MathView* [21]: es un proyecto de código abierto desarrollado en lenguaje C#, utiliza tanto *MathML* como *LaTeX*.
- *FMath* [22]: es una librería elaborada en *Javascript* y *Flash*, interpreta tanto

MathML, *LaTeX* y un lenguaje de marcado para matemáticas soportado por *Microsoft*.

- *MathPlayer* [23]: es un complemento para el navegador *Internet Explorer* para interpretar *MathML* funciona conjuntamente con el narrador de pantalla *JAWS*.
- *Gecko* [24]: es una librería que permite incluir el navegador web *Mozilla Firefox* dentro de un programa en lenguaje C#.

En la siguiente sección se examinará el proceso de pronunciar una expresión algebraica.

2.4.3 Verbal

La tercera representación de la Tabla 3 se refiere a la expresión original pero de manera verbalizada esta es redundante en palabras para poder expresar con exactitud la expresión algebraica.

Para los no videntes existe una marcada preferencia en utilizar información en audio que en braille. Algunos de los motivos son el costo y el tiempo, por citar un ejemplo para que una persona no vidente pueda conocer las noticias de un periódico, es mucho más simple y rápido escuchar a alguien más leer el periódico que obtener esta misma información en braille.

Esta preferencia ha provocado que se impulse el desarrollo de sintetizadores de voz dentro de las tecnologías de accesibilidad conocidos como “Text-To-Speech (TTS)”, que simplemente son programas informáticos capaces de leer la información que se encuentra de manera escrita a una manera hablada o verbalizada. En la Tabla 5 se listan algunos sintetizadores de voz.

Tabla 5

Sintetizadores de voz comerciales

	JAWS [25]	IVONA [26]	Narrator [27]
Sistema Operativo (S.O.)	Windows	Multiplataforma	Windows
Voces	4	51	Win. 7: idioma inglés Win. 8 en adelante: 16

Continua \implies

	JAWS	IVONA	Narrator
Idiomas	4	23	8
Instalación audible	SI	NO	Ya instalada
Soporte MathML	SI	NO	SI
Compatible con Office	SI	SI	SI
Licencia	SI	SI	Incluida con el S.O.
Acceso a librerías nativas	NO	SI	SI

Sin embargo, hay que notar que existe mucha ambigüedad al tratar de verbalizar una expresión algebraica. Una de las razones es que gráficamente se utiliza la ubicación de los elementos de una expresión o se extiende los símbolos alargándolos, en vez de utilizar separadores como paréntesis, llaves o corchetes. Lo que provoca que para una misma expresión verbalizada tenga varias representaciones gráficas equivalentes. La ambigüedad se da especialmente en los siguientes casos:

Índices, al tratarse de exponentes en el caso de la potenciación o de la base en el caso de logaritmos por ejemplo, gráficamente se diferencia estos elementos ubicándolos al lado derecho superior o inferior de otro elemento según sea el caso.

<i>Ejemplo</i>	¿Cuál de las siguientes expresiones algebraicas representa a la expresión verbalizada “A elevado a B más C menos D”?
Opción 1	$A^B + C - D$
Opción 2	$A^{B+C} - D$
Opción 3	A^{B+C-D}
Respuesta	Cualquiera de estas opciones podría ser la correcta, en realidad no se podría saber con certeza cuáles elementos pertenecen al exponente de la potencia y cuáles no.

Fraciones y radicales, para ambos casos sucede lo mismo cuando se tiene expresiones extensas dentro de los operadores de una raíz o de una fracción, lo que se suele

hacer es extender o alargar el signo de la operación hasta abarcar todos los elementos implicados.

<i>Ejemplo</i>	¿Cuál de las siguientes expresiones algebraicas representa a la expresión verbalizada “Raíz cuadrada de X más Y menos Z”?
Opción 1	$\sqrt{X+Y-Z}$
Opción 2	$\sqrt{X+Y}-Z$
Opción 3	$\sqrt{X}+Y-Z$
Respuesta	Cualquiera de estas opciones podría ser la correcta, como se pudo apreciar en la radicación para distinguir que elementos pertenecen a la raíz se delimita los elementos implicados con el símbolo de la raíz.

Según algunos artículos [28] y [29], se ha tratado de estandarizar la manera en como se debería verbalizar una expresión algebraica. La solución que se propone es sustituir algunas expresiones complejas o extensas por otras más simples.

Por ejemplo, de la fórmula general para determinar las raíces de un polinomio de segundo grado:

$$x = \frac{-b \pm \sqrt{b^2 - 4 * a * c}}{2 * a}$$

Se puede sustituir el término: $\sqrt{b^2 - 4 * a * c}$ por la letra alfa (α), entonces la fórmula quedaría:

$$x = \frac{-b \pm \alpha}{2 * a}, \quad \text{donde : } \alpha = \sqrt{b^2 - 4 * a * c}$$

De la misma manera, de forma verbalizada la misma fórmula original sería:

”x” es igual fracción donde el numerador es: menos “b” más menos raíz cuadrada de: “ “b” elevado al cuadrado menos “4” por “a” por “c” ”; y el denominador es: “2” por “a” por “c”.

Se puede sustituir la frase “raíz cuadrada de: “ “b” elevado al cuadrado menos “4” por “a” por “c” ”, por la letra alfa (α), que daría el siguiente resultado:

“x” es igual fracción donde el numerador es: menos “b” más menos α ; y el denominador es: “2” por “a” por “c”.

Donde: α es igual a “raíz cuadrada de: “ “b” elevado al cuadrado menos “4” por “a” por “c” ” ”

Con este método de sustitución, se puede anidar expresiones algebraicas unas dentro de otras, permitiendo obtener un mayor grado de complejidad al verbalizarlas, reduciendo la carga mental y dando una mejor visión general de todas una expresiones algebraicas.

2.4.4 Táctil

La última representación que se muestra en la segunda fila de la columna “Ejemplo” de la Tabla 3 representa a la misma expresión que en la fila anterior pero con la signografía braille, esta incluye más caracteres al adaptar o traducir al braille.

El sistema braille tiene 3 grados de simplificación (del 1 al 3), donde el grado 2 y 3 tratan de simplificar y reducir caracteres al transcribir textos al braille a fin de reducir espacio, en cambio el grado 1 simplemente es una sustitución de los caracteres en tinta a braille. El grado de transcripción más empleado es el grado 1, aceptado también por la Comisión de Braille Española C.B.E., máximo organismo legal en España que establece:

- Normas para la correcta transcripción al braille de documentos en distintas materias.
- Normas para confeccionar materiales accesibles en relieve (mapas, planos, señales, pictogramas, etc.).
- Recomendaciones a distintos sectores para incluir el braille en sus productos y servicios (etiquetado de envases, marcas en braille en distintos soportes, etc.).

Por lo expuesto anteriormente se ha adoptado en este proyecto las recomendaciones de documentos técnicos oficiales como:

- “Braille y matemáticas” [12].
- “Signografía básica” [30] de las guías de la comisión braille española.
- “Signografía matemática” [31] de las guías de la comisión braille española.

Capítulo 3

Diseño del Software

En este capítulo se explicará el diseño del software de editor de expresiones algebraicas elementales para personas no videntes considerando las necesidades especiales de estos usuarios como requisitos de diseño.

El programa consiste en un editor de expresiones algebraicas -no una hoja de cálculo-, que permite a una persona no vidente armar expresiones matemáticas por medio de plantillas predefinidas de operaciones matemáticas que poseen espacios para ser rellenados con la información ingresada por el usuario, un sintetizador de audio asiste al usuario no vidente por medio de mensajes de voz para informar de lo que sucede dentro del programa. Una vez ingresada la expresión el programa puede convertir la información a representaciones braille, audibles o visuales por medio de comandos.

3.1 Necesidades del Usuario

Las necesidades de un usuario no vidente influyen directamente en el diseño del programa, por ello fue necesario conocer el manejo del computador de las personas no videntes para determinar sus necesidades concretas:

- El usuario no vidente no utiliza ni el monitor ni el ratón del computador para interactuar con el computador, consecuentemente se debe utilizar otros periféricos de entrada y salida para realizar las mismas funciones. El teclado y mensajes de audio pueden cumplir este papel.
- El usuario no vidente no sabe lo que está ocurriendo en pantalla al ejecutarse un programa, por lo tanto es necesario informarle mediante un narrador de pantalla (sintetizador de voz) acerca de los eventos y acciones que están ocurriendo.
- El usuario no vidente no puede localizar el puntero del sistema, por lo tanto es necesario utilizar el tanto foco del teclado (una señal similar al puntero del ratón pero controlada por el teclado) y mensajes de voz con la ubicación del foco como alternativa.
- El usuario no vidente preferentemente no utiliza un explorador de archivos simplemente tiene un directorio base definido donde administra sus archivos.

- El usuario no vidente tiene dificultades para ingresar, editar y modificar expresiones algebraicas usando la notación matemática común debido a la ubicación de los componentes, para ello es necesario utilizar una estructura de datos llamada árbol y un control llamado árbol de vista para navegar por la expresión usando el teclado.
- El usuario no vidente tiene dificultades al interpretar una expresión algebraica en forma audible debido a que existe ambigüedad al verbalizarla.
- El usuario no vidente desea interactuar con el computador con mayor rapidez.
- El usuario con baja visión si utiliza el monitor y el ratón del computador pero necesita de un magnificador de pantalla y de un alto contraste de colores, normalmente los sistemas operativos cuentan con estas opciones de accesibilidad.

3.2 Interfaz Gráfica de Usuario

Para el diseño se ha seguido los principios de diseño expuestos en la publicación “Diseño de pantalla” [32] y los documentos “Windows Vista and 7 User Experience Interaction Guidelines” [33] y “Windows 8 desing and coding guidelines”[34]. Motivo por el cual en las siguientes secciones se irá explicando en detalle el diseño previo a la implementación de este software. Para tener una idea general del programa, véase la figura 3.

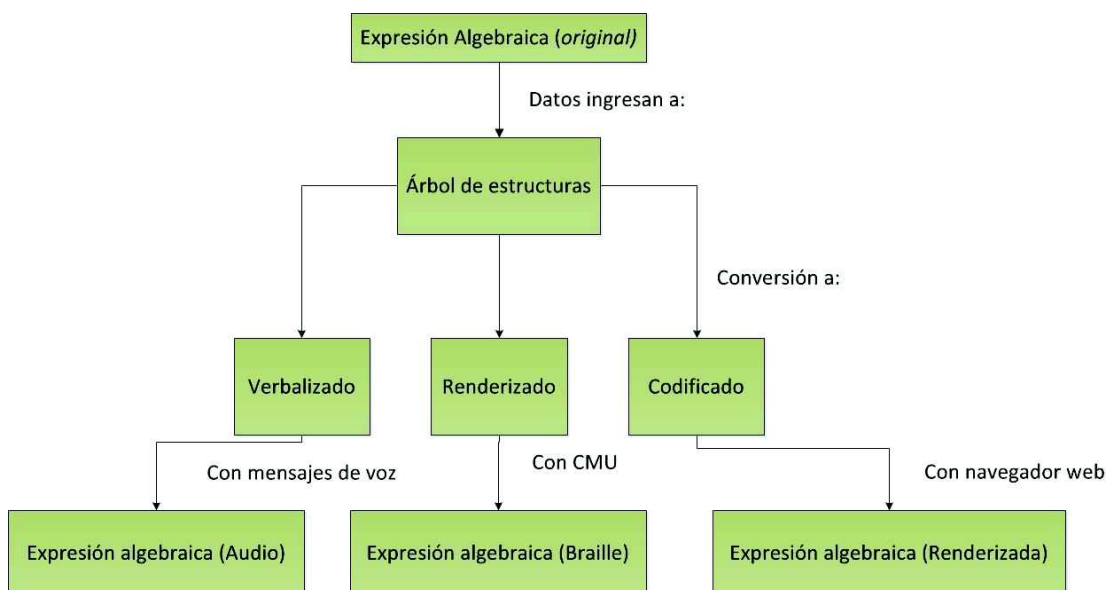


Figura 3 Funcionamiento general del programa

Algunas características del programa:

- El nombre del programa es “Editor de Expresiones Algebraicas Elementales”.
- Los archivos generados por el programa tendrán la extensión: “.xml”.
- Ventanas de diálogo simplificadas con respecto a las comunes de Windows.

3.2.1 Arquitectura y Navegación

En esta sección, se explica la estructura general del programa, el cuál se compone de las siguientes ventanas:

- Una *ventana de presentación*, con la cual inicia y da a conocer algunos detalles del programa con respecto a su desarrollo.
- Una *ventana del editor*, en la cual el usuario puede ingresar, editar o modificar expresiones algebraicas a través del teclado, revisar o escuchar.
- Una *ventana para abrir archivos*, una ventana simplificada pensada en el usuario no vidente, en la cual puede buscar un archivo “.xml” almacenado en su equipo en el directorio base para poder seguir trabajando con él, este formato que permite almacenar y organizar con estructuras de datos tipo árboles.
- Una *ventana para guardar archivos*, una ventana simplificada pensada en el usuario no vidente, en la cual puede guardar un archivo “.xml” y almacenarlo en su equipo en el directorio base, el archivo contiene la información de usuario ingresada en el árbol de estructuras.
- Una *ventana para imprimir archivos*, en la cual el usuario puede imprimir un archivo “.txt” almacenado en su equipo en el directorio predeterminado usando las impresoras instaladas en su computador, el archivo contiene las expresiones algebraicas ingresadas del árbol de estructuras traducida al sistema braille.
- Una *ventana para configuraciones de usuario*, en la cual el usuario puede personalizar ciertas características del programa según sus preferencias.

Usando una arquitectura se ha organizado las ventanas del programa mediante niveles (ver Figura 4), que se clasifican en: nivel 1 o de información, nivel 2 o de eventos y nivel 3 o de edición.

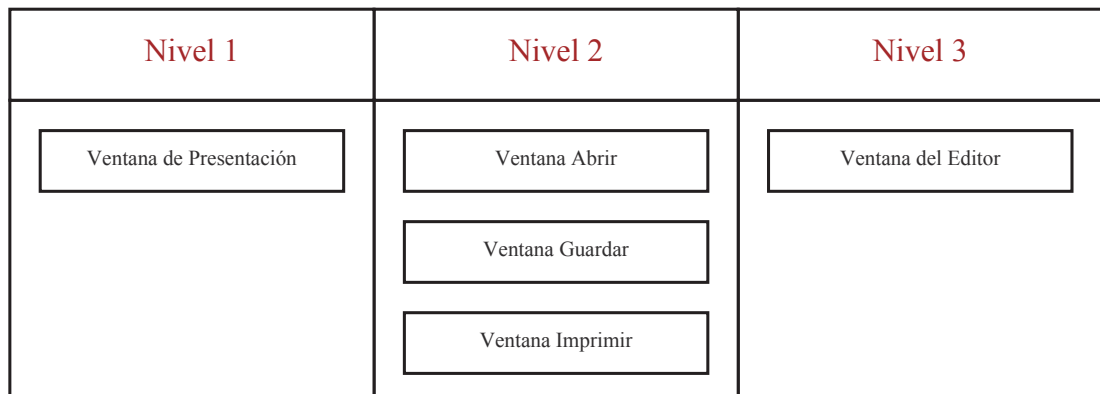


Figura 4 Arquitectura del programa

También es importante desde el punto de vista del usuario conocer cómo desplazarse entre las distintas ventanas del programa, por lo que mediante el diagrama de navegación (ver Figura 5) se establece la interacción del usuario entre las distintas ventanas como una guía para orientación.

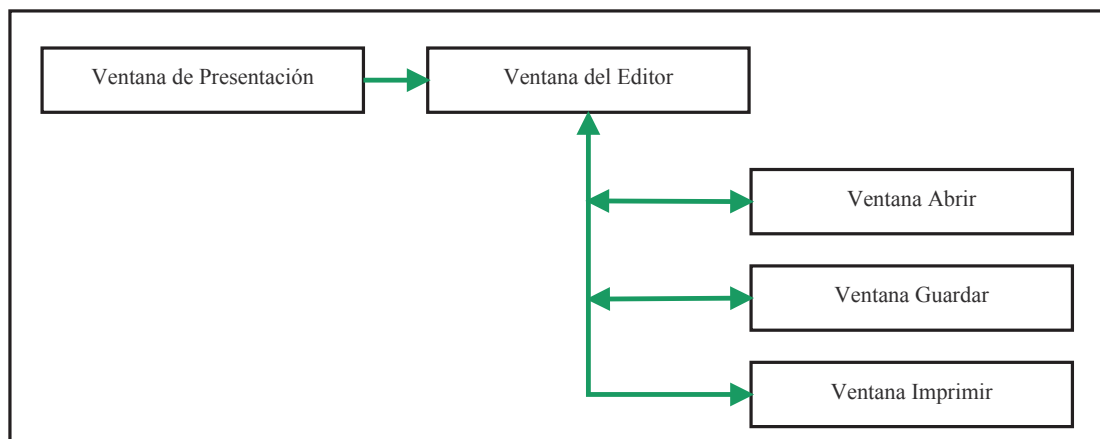


Figura 5 Diagrama de la navegación entre ventanas

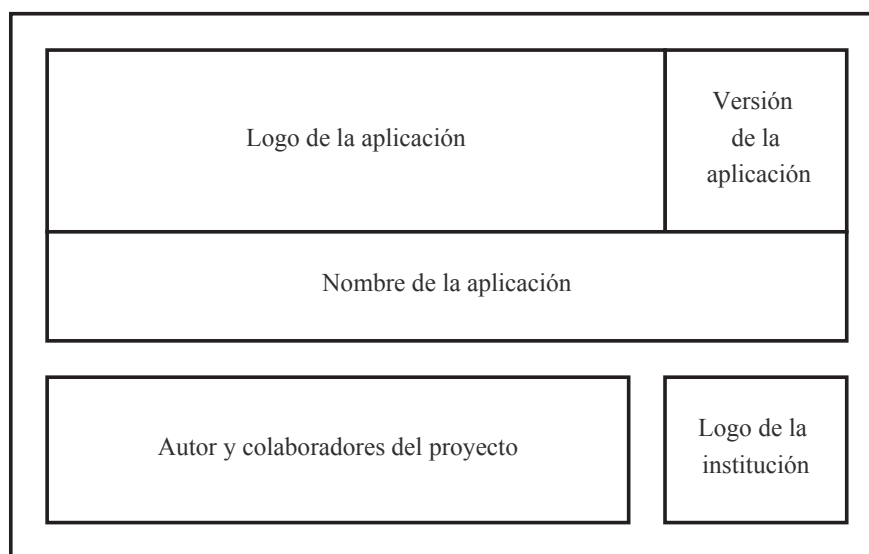
3.2.2 Distribución

En esta sección se muestra cuáles son los componentes de la interfaz gráfica así como su ubicación, por medio de dos aspectos: en primer lugar, un listado de los componentes y bosquejos gráficos; en segundo lugar las características de pantalla.

- **Ventana de Presentación:** la siguiente lista contiene los componentes de la ventana (Tabla 6) y bosquejo gráfico muestra su ubicación (Figura 6).

Tabla 6**Componentes de la ventana de presentación**

1	Nombre, logo y versión de la aplicación
2	Logo de la institución
3	Nombre del autor y colaboradores del proyecto

**Figura 6 Bosquejo de la ventana de presentación**

- Ventana del Editor: la siguiente lista contiene los componentes de la ventana (Tabla 7) y el bosquejo gráfico muestra su ubicación (Figura 7).

Tabla 7**Barra de menús de la ventana del editor**

Núm	Item
1	Nombre del programa
2	Barra de menús
3	Barra de herramientas
4	Menú de búsqueda dinámica
5	Panel de entrada e expresiones algebraicas
6	Cuadro de edición
7	Panel de renderización

Continua \Rightarrow

Núm	Item
8	Panel de entrada de texto
9	Panel braille
10	Barra de información

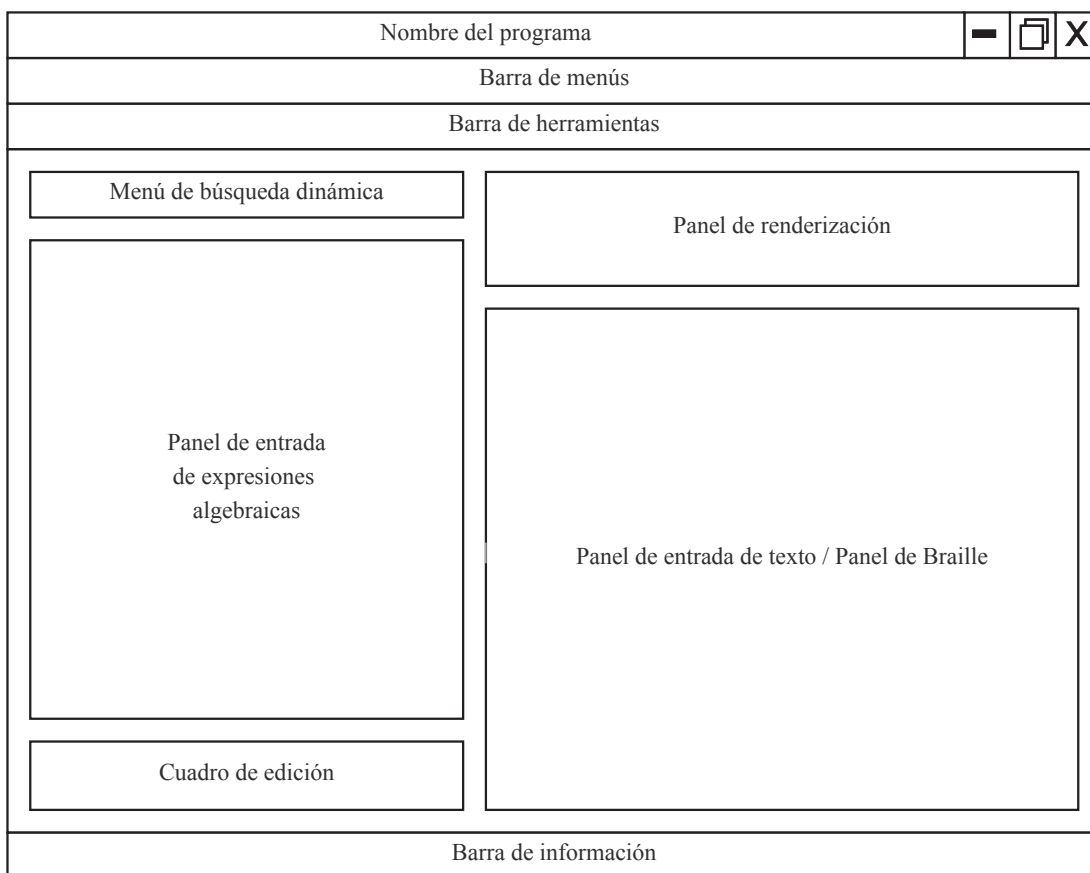


Figura 7 Bosquejo de la ventana del editor

- Barra de Menús de la ventana del editor: la siguiente lista de contiene los diferentes títulos de los menús y opciones que se podrán encontrar en esta barra (Tabla 8).

Tabla 8

Barra de menús de la ventana del editor

Menú	Opciones	Sub-opciones
Archivo	Nuevo	---
	Abrir	---

Continua ⇒

Menú	Opciones	Sub-opciones
Archivo	Cerrar	---
	Exportar	MS Word Braille 6 ptos.
	Guardar como. . .	---
	Imprimir. . .	---
	Salir	---
Editar	Insertar	---
	Modificar	---
	Eliminar	---
	Localizar	---
Ver	Panel de braille	---
	Panel tinta	---
Generar	Braille	---
	Render	---
	Audiotexto	---
Ayuda	Comandos rápidos	---
	Documentación. . .	---
	Acerca de. . .	---
Opciones	Leer en voz alta	Activar Desactivar Callar Cambiar de voz
	Modo de lectura	Ninguno Carácter Palabra Oración Párrafo Todo
	Volumen	Subir Bajar
	Velocidad	Subir Bajar
	Directorio Base	---

- Barra de Herramientas de la ventana del editor: la siguiente lista contiene distin-

tos menús y opciones que se podrán encontrar en esta barra (Tabla 9).

Tabla 9

Barra de herramientas de la ventana del editor

Sub-barras	Función	Tipo control
Estándar	Abrir...	Pulsador
	Guardar como...	Pulsador
	Imprimir...	Pulsador
Matemáticas	Eliminar	Pulsador
	Insertar	Pulsador
	Localizar	Pulsador
	Modificar	Pulsador
Audiotexto	Reproducir	Botón
	Pausar	Botón
	Detener	Pulsador
	Volumen	Barra de desplazamiento
	Voz	Lista desplegable
	Modo de lectura	Lista desplegable
Conversión	Generar Braille	Pulsador
	Generar Render	Pulsador
Otros	Panel Braille/Tinta	Selector

- Barra de Información de la ventana del editor: la siguiente lista contiene la información que se presentará con etiquetas en esta barra (Tabla 10).

Tabla 10

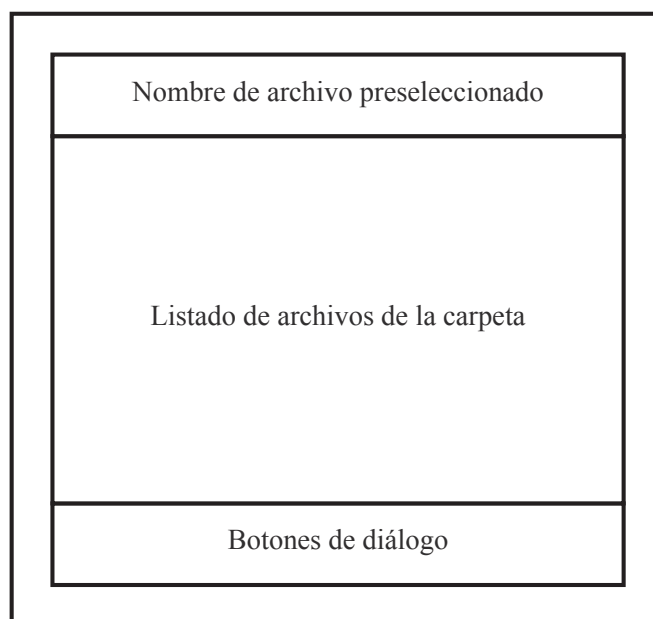
Barra de Información de la ventana del editor

Función	Tipo
Foco del sistema	Etiqueta
Coordenadas	Etiqueta
Mensaje del programa	Etiqueta
Directorio base	Etiqueta
Voces instaladas	Lista

- Ventana de Abrir/Guardar: la siguiente lista contiene los componentes de esta ventana (Tabla 11) y bosquejo gráfico muestra su ubicación (Figura 8).

Tabla 11**Componentes de las ventanas abrir y guardar**

1	Listado de archivos
2	Archivo preseleccionado
3	Botones de diálogo

**Figura 8 Bosquejo de la ventana de abrir/guardar**

- Ventana de Imprimir: la siguiente lista contiene los componentes de esta ventana (Tabla 12) y bosquejo gráfico muestra su ubicación (Figura 9).

Tabla 12**Componentes de la ventana imprimir**

Item	Nombre
1	Vista previa
2	Impresora seleccionada
3	Botones de diálogo

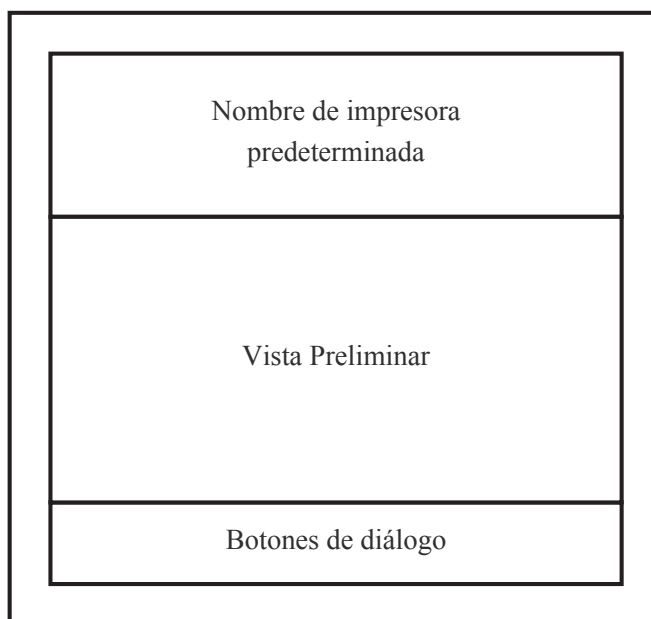


Figura 9 Bosquejo de la ventana de imprimir

En segundo lugar, tomando en cuenta que cada monitor tiene características dependiendo del tamaño, resolución y orientación de la pantalla (ver Tabla 13).

Tabla 13

Características del monitor

Tamaño mínimo:	1600 x 900
Resolución mínima:	1024 x 600
Orientación:	Horizontal

Se debe tener en cuenta estos parámetros mínimos para que las ventanas aparezcan en pantalla y la proporción de los componentes tenga una proporción estética y visual que permita diferenciar los componentes dentro de la ventana del programa.

3.2.3 Tipografía

El uso de caracteres adicionales al alfabeto latino como signos braille, letras griegas en el programa provoca que deba utilizarse distintas tipografías para abarcar todos estos símbolos necesarios en las fórmulas matemáticas en braille o en notación matemática común, en las Tablas 14, 15 y 16 se encuentra definidas las tipografías utilizadas

con el propósito de dar unicidad y coherencia al programa; definiendo el estilo, tamaño, color y uso de las tipografías en los títulos, menús, recuadros, cuadros de texto, etiquetas, listas, entre otros componentes de las ventanas gráficas.

Tabla 14**Tipografía 1**

Tipo de letra	“Microsoft Sans Serif”
Tamaño	Mínimo 9 puntos y máximo 16 Puntos
Estilo	Normal
Color	Negro RGB: (255, 255, 255)
Uso	Texto de elementos comunes en ventanas: títulos, menús, cuadros de texto y etiquetas

Tabla 15**Tipografía 2**

Tipo de letra	“Segoe UI Symbol”
Tamaño	Mínimo 20 Puntos
Estilo	Normal
Color	RGB: (255, 255, 255)
Uso	Cuadros de texto para visualización en braille.

Tabla 16**Tipografía 3**





Tipo de letra	“Segoe UI Symbol”
Tamaño	Mínimo 20 Puntos
Estilo	Normal
Color	RGB: (255, 255, 255)
Uso	Cuadros de texto para visualización de caracteres del alfabeto griego.

3.2.4 Color

Para el uso de colores de fondo se ha establecido una paleta de cuatro colores para todo el proyecto (ver Tabla 17).

Tabla 17

Paleta de colores

#	Color	Código
1		HEX: 58D379 RGB: 88–211–121
2		HEX: 52A9BD RGB: 82–169–189
3		HEX: FFB46B RGB: 255–180–107
4		HEX: FF7C6B RGB: 255–124–107

3.2.5 Contenidos

Basándose en el alcance del proyecto y en el artículo [35] que muestra cuales son las operaciones más utilizadas en los textos escolares, se ha optado por limitar la cantidad de operaciones, funciones, símbolos matemáticos, etc. El soporte que brinda el editor se expone en las tablas de operaciones, funciones y símbolos (ver tablas 18 y 19).

Tabla 18

Operaciones y Funciones

Tipo	Estructura	Ejemplo
Básicas	Adición	$A + B$
	Sustracción	$A - B$
	Producto	$A * B$
	Cociente	A / B
Relación	Mayor que	$A > B$
	Menor que	$A < B$

Continua \implies

Tipo	Estructura	Ejemplo
Relación	Igual	$A = B$
	Mayor o igual que	$A \geq B$
	Menor o igual que	$A \leq B$
Booleanas	OR	$V \vee F$
	AND	$V \& F$
Otras	Potenciación	A^b
	Radicación	$\sqrt[b]{A}$
	Logaritmos	$\log_b(A)$
	Fracciones	$\frac{A}{B}$
Trigonómicas Básicas	Seno	$\sin(\angle\theta)$
	Coseno	$\cos(\angle\theta)$
	Tangente	$\tan(\angle\theta)$
	Secante	$\sec(\angle\theta)$
	Cosecante	$\csc(\angle\theta)$
	Cotangente	$\cot(\angle\theta)$
Trigonómicas Inversas	Arco-seno	$\arcsin(\angle\theta)$
	Arco-coseno	$\arccos(\angle\theta)$
	Arco-tangente	$\arctan(\angle\theta)$
	Arco-secante	$\operatorname{arcsec}(\angle\theta)$
	Arco-cosecante	$\operatorname{arccsc}(\angle\theta)$
	Arco-cotangente	$\operatorname{arccot}(\angle\theta)$

Tabla 19
Símbolos

Tipo	Subtipo	Ejemplo
Letras griegas	Minúsculas	$\alpha, \beta, \chi, \delta, \varepsilon, \phi,$ $\varphi, \gamma, \eta, \iota, \kappa, \lambda,$ $\mu, \nu, \pi, \varpi, \theta, \vartheta,$ $\rho, \sigma, \zeta, \tau, \upsilon, \omega,$ ξ, ψ, ζ
	Mayúsculas	$\Delta, \Phi, \Gamma,$ $\Lambda, \Pi, \Theta,$ $\Sigma, \Upsilon, \Omega,$ Ξ, Ψ

Continua \Rightarrow

Tipo	Subtipo	Ejemplo
Números romanos	Minúsculas	<i>i, v, x,</i> <i>l, c, d,</i> <i>m</i>
	Mayúsculas	<i>I, V, X,</i> <i>L, C, D,</i> <i>M</i>

3.2.6 Presentación de la información

A fin de interactuar con al usuario se presentará la información de los diferentes paneles de entrada y salida con los siguientes formatos que se describen a continuación:

Panel de entrada de expresiones algebraicas

Las expresiones algebraicas dentro de este panel se insertan de manera semántica tomando en cuenta si existe anidación, por esta razón se ha organizado a las operaciones y funciones como estructuras matemáticas así también sus componentes como parámetros.

Para obtener un diseño orientado a objetos y presentar las estructuras matemáticas dentro del árbol de datos, fue necesario clasificarlas (ver tabla 20) según el número de parámetros.

Tabla 20

Estructuras clasificadas por número de parámetros

Nombre	Tipo	Parámetros	# Paráms.
Arco-cosecante	Trigonométrica	Argumento	1
Arco-coseno	Trigonométrica	Argumento	1
Arco-cotangente	Trigonométrica	Argumento	1
Arco-secante	Trigonométrica	Argumento	1
Arco-seno	Trigonométrica	Argumento	1
Arco-tangente	Trigonométrica	Argumento	1
Cosecante	Trigonométrica	Argumento	1
Coseno	Trigonométrica	Argumento	1
Cotangente	Trigonométrica	Argumento	1
Secante	Trigonométrica	Argumento	1

Continua \implies

Nombre	Tipo	Parámetros		# Paráms.
Seno	Trigonométrica	Argumento		1
Tangente	Trigonométrica	Argumento		1
División	Aritmética	Dividendo	Divisor	2
Ecuación	Aritmética	Miembro 1	Miembro 2	2
Fracción	Aritmética	Numerador	Denominador	2
Logaritmo	Aritmética	Base	Argumento	2
Mayor o igual que	Aritmética	Miembro 1	Miembro 2	2
Mayor que	Aritmética	Miembro 1	Miembro 2	2
Menor o igual que	Aritmética	Miembro 1	Miembro 2	2
Menor que	Aritmética	Miembro 1	Miembro 2	2
Multiplicación	Aritmética	Multiplicando	Multiplicador	2
O Lógica	Aritmética	Miembro 1	Miembro 2	2
Potencia	Aritmética	Base	Exponente	2
Raíz	Aritmética	Índice	Radicando	2
Resta	Aritmética	Minuendo	Sustraendo	2
Suma	Aritmética	Sumando 1	Sumando 2	2
Y Lógica	Aritmética	Miembro 1	Miembro 2	2

El resultado fue que las estructuras que poseen 1 parámetro son las funciones trigonométricas directas e inversas mientras que las operaciones aritméticas, de relación y lógicas tienen 2 parámetros. Consecuentemente mediante una base de datos se puede almacenar una plantilla de cada estructura matemática (ver figura 10) a fin de ser manipulada en el árbol de datos cuando el usuario seleccione, ingrese, modifique o elimine una estructura.

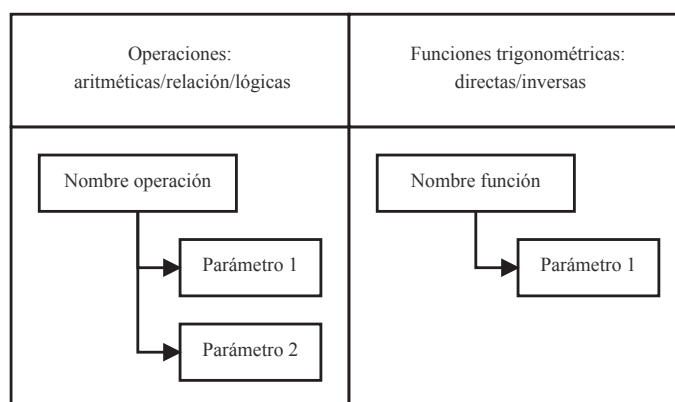


Figura 10 Plantilla de una operación y una función

El mismo hecho de que se utilice un árbol de datos provoca que:

- Solo exista un único nodo raíz, por lo tanto el nodo raíz es “Expresiones” y cada nueva expresión será un subnodo del nodo raíz que se llamará “Expresión”.
- No usar limitadores de expresiones como paréntesis, llaves y corchetes; simplemente anidando una nueva operación al agregar un nuevo nodo hijo a partir las plantillas de la base de datos.

Panel de renderizado

Para presentar una expresión algebraica en este panel se debe convertir los datos ingresados en el árbol de datos al formato MathML, que no es muy diferente al árbol de datos, básicamente se debe etiquetar cada nodo del árbol de datos (ver tablas 21 y 22) e insertar toda esta nueva estructura de datos en un archivo XHTML o página *web* para poder ser visualizado desde el panel de renderizado que contiene el navegador *web Gecko* embebido.

Tabla 21

Etiquetas MathML de las estructuras matemáticas

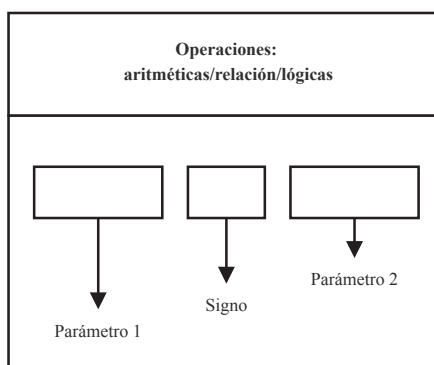
#	Nombre	Etiqueta	#	Nombre	Etiqueta
1	Arco-cosecante (acsc)	mi	16	Mayor o igual que (\geq)	mo
2	Arco-coseno (acos)	mi	17	Mayor que ($>$)	mo
3	Arco-cotangente (acot)	mi	18	Menor o igual que (\leq)	mo
4	Arco-secante (asec)	mi	19	Menor que ($<$)	mo
5	Arco-seno (asen)	mi	20	Multiplicación (\times)	mo
6	Arco-tangente (atan)	mi	21	O Lógica (\mid)	mo
7	Cosecante (csc)	mi	22	Potencia	msup
8	Coseno (cos)	mi	23	Raíz	mroot
9	Cotangente (cot)	mi	24	Resta ($-$)	mo
10	División ($/$)	mo	25	Secante (sec)	mi
11	Ecuación ($=$)	mo	26	Seno (sen)	mi
12	Expresión	mrow	27	Suma ($+$)	mo
13	Expresiones	mrow	28	Tangente (tan)	mi
14	Fracción	mfrac	29	Y Lógica ($\&$)	mo
15	Logaritmo (log)	mlog			

Tabla 22**Etiquetas MathML para los parámetros de las estructuras matemáticas**

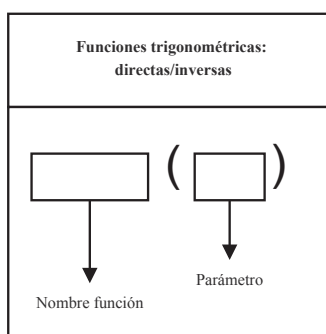
Tipo	Subtipo	Etiqueta
No definidos	- - -	mrow
Definidos	Números	mn
	Letras	mi
	Signos	mo
	Mixtos	mrow

El formato de salida renderizado para las estructuras matemáticas:

- La figura 11 es una plantilla de las operaciones aritméticas básicas, lógicas y de relación.

**Figura 11 Renderización de una operación**

- La figura 12 es una plantilla de las funciones trigonométricas básicas e inversas.

**Figura 12 Renderización de una función**

- La figura 13 es una plantilla de las estructuras matemáticas que poseen anidamiento.

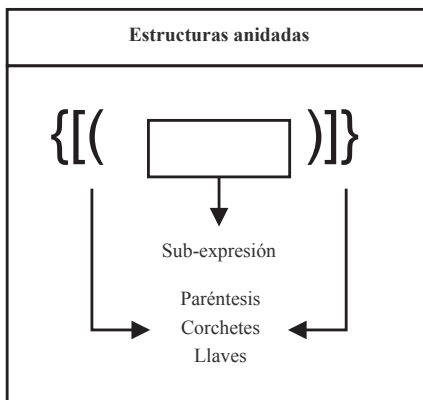


Figura 13 Renderización de una operación anidada

- La figura 14 es una plantilla de las fracciones matemáticas.

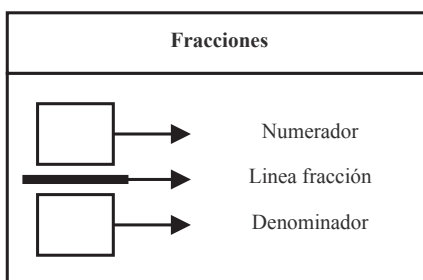


Figura 14 Renderización de una fracción

- La figura 15 es una plantilla de las funciones de logaritmos matemáticos.

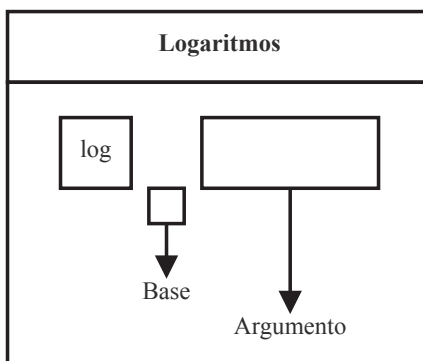


Figura 15 Renderización de un logaritmo

- La figura 16 es una plantilla de las operaciones de potencias matemáticas.

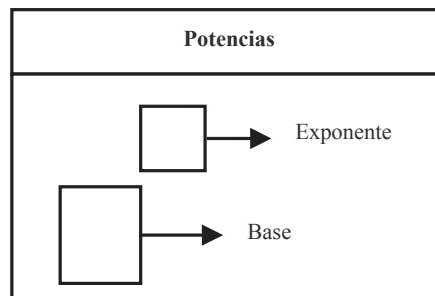


Figura 16 Renderización de una potencia

- La figura 17 es una plantilla de las operaciones de raíces matemáticas.

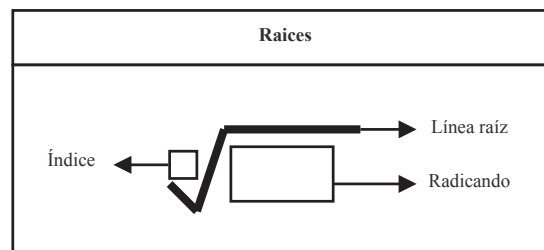


Figura 17 Renderización de una raíz

Panel de entrada de texto tinta

La información ingresada en este panel por medio del teclado del computador ha provocado la necesidad de que exista una realimentación por parte del computador para que el usuario este informado acerca de los eventos provocados al presionar una tecla o un conjunto de las mismas. Se clasifica a continuación dependiendo del modo de lectura:

- Teclas: al presionar cualquier tecla dependiendo del tipo se producen un evento: el primero cuando se presiona una tecla que representa a un carácter imprimible (ver tabla 23) y segundo cuando se presiona una tecla de función (ver tabla 24). Adicionalmente, no todas las teclas son reconocidas como usualmente las reconoce una persona sin conocimientos de programación (el usuario común). En consecuencia al producirse la pulsación de una tecla se produce un eco audible de la misma .

Tabla 23**Eco para cada tecla imprimible**

Caracter	Eco
	Espacio
!	Signo de exclamación cerrado
"	Comillas dobles
#	Numeral
\$	Dólar
%	Porcentaje
&	Andpersand
'	Apóstrofe
(Signo de paréntesis abierto
)	Signo de paréntesis cerrado
*	Asterisco
+	Más
,	Coma
-	Menos
.	Punto
/	Slash
:	Dos Puntos
;	Punto y coma
<	Menor que
=	Igual
>	Mayor que
?	Signo de interrogación cerrado
@	Arroba
A	A mayúscula
B	B mayúscula
C	C mayúscula
D	D mayúscula
E	E mayúscula
F	F mayúscula
G	G mayúscula
H	H mayúscula

Continua \implies

Caracter	Eco
I	I mayúscula
J	J mayúscula
K	K mayúscula
L	L mayúscula
M	M mayúscula
N	N mayúscula
O	O mayúscula
P	P mayúscula
Q	Q mayúscula
R	R mayúscula
S	S mayúscula
T	T mayúscula
U	U mayúscula
V	V mayúscula
W	Doble ve mayúscula
X	X mayúscula
Y	Ye mayúscula
Z	Z mayúscula
[Corchete abierto
	Back Slash
]	Corchete cerrado
◌	Acento circunflejo
_	Guión bajo
˘	Tilde grave
a	a minúscula
b	b minúscula
c	c minúscula
d	d minúscula
e	e minúscula
f	f minúscula
g	g minúscula
h	h minúscula
i	i minúscula
j	j minúscula

Continua \Rightarrow

Caracter	Eco
k	k minúscula
l	l minúscula
m	m minúscula
n	n minúscula
o	o minúscula
p	p minúscula
q	q minúscula
r	r minúscula
s	s minúscula
t	t minúscula
u	u minúscula
v	v minúscula
w	doble ve minúscula
x	x minúscula
y	ye minúscula
z	z minúscula
{	Llave abierta
	Barra Vertical
}	Llave cerrada
	Tilde de la eñe
Retroceso	Retroceso
Tabulación	Tabulación
¡	Signo de exclamación abierto
¿	Signo de interrogación abierto
À	À mayúscula
Á	Á mayúscula
È	È mayúscula
É	É mayúscula
Ì	Ì mayúscula
Í	Í mayúscula
Enter	Enter
Ñ	Eñe mayúscula
Ò	Ò mayúscula
Ó	Ó mayúscula

Continua ⇒

Caracter	Eco
×	Por
Ù	Ù mayúscula
Ú	Ú mayúscula
Û	Û mayúscula
à	à minúscula
á	á minúscula
è	è minúscula
é	é minúscula
ì	ì minúscula
í	í minúscula
ñ	eñe minúscula
ò	ò minúscula
ó	ó minúscula
÷	Para
ù	ù minúscula
ú	ú minúscula
ü	ü minúscula
<i>A</i>	alfa mayúscula
<i>B</i>	beta mayúscula
Γ	gamma mayúscula
Δ	delta mayúscula
<i>E</i>	épsilon mayúscula
<i>Z</i>	zeta mayúscula
<i>H</i>	eta mayúscula
Θ	theta mayúscula
<i>I</i>	iota mayúscula
<i>K</i>	kappa mayúscula
Λ	lambda mayúscula
<i>M</i>	my mayúscula
<i>N</i>	ny mayúscula
Ξ	xi mayúscula
<i>O</i>	ómicron mayúscula
Π	pi mayúscula
<i>P</i>	rho mayúscula

Continua \implies

Caracter	Eco
Σ	sigma mayúscula
T	tau mayúscula
Υ	ípsilon mayúscula
Φ	phi mayúscula
X	ji mayúscula
Ψ	psi mayúscula
Ω	omega mayúscula
α	alfa
β	beta
γ	gamma
δ	delta
ϵ	épsilon
ζ	zeta
η	eta
θ	theta
ι	iota
κ	kappa
λ	lambda
μ	my
ν	ny
ξ	xi
\omicron	ómicron
π	pi
ρ	rho
σ	sigma
τ	tau
υ	ípsilon
ϕ	phi
χ	ji
ψ	psi
ω	omega

Tabla 24**Eco para cada tecla de función**

Tecla	Eco
SHIFT	Sin eco
CONTROL	Sin eco
ALT	Sin eco
CTRL+ALT	Sin eco
Pausa	Pausa
Bloque Mayúsculas	Bloque Mayúsculas
Escape	Sin eco
Retroceder Página	Retroceder Página
Avanzar Página	Avanzar Página
Fin	Fin
Inicio	Inicio
Flecha Izquierda	Sin eco
Flecha Arriba	Sin eco
Flecha Derecha	Sin eco
Flecha Abajo	Sin eco
Insertar	Insertar
Suprimir	Suprimir
Tecla Windows	Tecla Windows
Apps	Apps
F1	Sin eco
F2	Sin eco
F3	Sin eco
F4	Sin eco
F5	Sin eco
F6	Sin eco
F7	Sin eco
F8	Sin eco
F9	Sin eco
F10	Sin eco
F11	Sin eco
F12	Sin eco

Continua \implies

Tecla	Eco
Bloque Numérico	Bloque Numérico
Bloque Desplazamiento	Bloque Desplazamiento

- Palabras: al detectar la pulsación de la barra espaciadora, mediante el sintetizador de voz debe leer la última palabra ingresada antes del caracter ' '.
- Oraciones: al detectar la pulsación de la tecla con del caracter '.', mediante el sintetizador de voz debe leer la última oración ingresada antes del caracter '.'.
- Párrafos: al detectar la pulsación de la tecla *enter*, mediante el sintetizador de voz debe leer la última oración ingresada antes del caracter de salto de línea.
- Lectura completa: simplemente el sintetizador de voz leerá todo el texto ingresado de principio a fin.

Panel de salida de texto braille

La característica más notable de este panel es de soportar en su tipografía los caracteres braille, pero igual que en casos anteriores necesita de una realimentación auditiva (eco) para que el usuario no vidente sepa qué caracter se encuentra en el panel. Los caracteres braille soportados se encuentran en la tabla 25 mientras que las plantillas o modelos para las estructuras en braille se encuentran en la tabla 26.

Tabla 25

Eco para caracteres braille

#	Caracter	Patrón	#	Caracter	Patrón
1	⠠	---	33	⠠	6
2	⠡	1	34	⠡	16
3	⠢	2	35	⠢	26
4	⠣	12	36	⠣	126
5	⠤	3	37	⠤	36
6	⠥	13	38	⠥	136
7	⠦	23	39	⠦	236
8	⠧	123	40	⠧	1236

Continua \implies

#	Caracter	Patrón	#	Caracter	Patrón
9	•	4	41	••	46
10	••	14	42	•••	146
11	•••	24	43	••••	246
12	••••	124	44	•••••	1246
13	•••	34	45	•••••	346
14	••••	134	46	••••••	1346
15	•••••	234	47	••••••	2346
16	••••••	1234	48	•••••••	12346
17	•••	5	49	•••••	56
18	••••	15	50	••••••	156
19	•••••	25	51	•••••••	256
20	••••••	125	52	••••••••	1256
21	••••••	35	53	••••••••	356
22	•••••••	135	54	•••••••••	1356
23	•••••••	235	55	•••••••••	2356
24	••••••••	1235	56	••••••••••	12356
25	•••••••	45	57	•••••••••	456
26	••••••••	145	58	••••••••••	1456
27	••••••••	245	59	••••••••••	2456
28	•••••••••	1245	60	•••••••••••	12456
29	••••••••	345	61	•••••••••••	3456
30	•••••••••	1345	62	•••••••••••	13456
31	•••••••••	2345	63	•••••••••••	23456
32	••••••••••	12345	64	••••••••••••	123456

Tabla 26

Salida braille modelo de las estructuras matemáticas

Nombre	Modelo	Braille
Arco-cosecante	$\text{acsc}()$	$\text{acsc}()$
Arco-coseno	$\text{acos}()$	$\text{acos}()$
Arco-cotangente	$\text{acot}()$	$\text{acot}()$
Arco-secante	$\text{asec}()$	$\text{asec}()$
Arco-seno	$\text{asin}()$	$\text{asin}()$
Arco-tangente	$\text{atan}()$	$\text{atan}()$
Cosecante	$\text{csc}()$	$\text{csc}()$
Coseno	$\text{cos}()$	$\text{cos}()$
Cotangente	$\text{cot}()$	$\text{cot}()$
Secante	$\text{sec}()$	$\text{sec}()$
Seno	$\text{sin}()$	$\text{sin}()$
Tangente	$\text{tan}()$	$\text{tan}()$
División	$() / ()$	$() / ()$
Ecuación	$() = ()$	$() = ()$
Fracción	$() \div ()$	$() \div ()$
Mayor o igual que	$() \geq ()$	$() \geq ()$
Mayor que	$() > ()$	$() > ()$
Menor o igual que	$() \leq ()$	$() \leq ()$
Menor que	$() < ()$	$() < ()$
Multiplicación	$() \times ()$	$() \times ()$
O Lógica	$() ()$	$() ()$
Potencia	$() \wedge ()$	$() \wedge ()$
Resta	$() - ()$	$() - ()$
Suma	$() + ()$	$() + ()$

Continua \implies

Nombre	Modelo	Braille
Y Lógica	() & ()	⠠⠠⠠⠠⠠⠠⠠⠠
Logaritmo	log () ()	⠠⠠⠠⠠⠠⠠⠠⠠⠠
Raíz	raíz() ()	⠠⠠⠠⠠⠠⠠⠠
Sub-expresión	[], ()	⠠⠠⠠⠠⠠⠠⠠⠠

Audiotexto

La información que el computador devuelve al usuario no vidente a través del sintetizador de voz son los datos ingresados en el panel de entrada de expresiones algebraicas convertidos a audiotexto. El modelo o plantilla para cada estructura en audiotexto se muestra en la tabla 27.

Tabla 27

Salida audiotexto modelo de las estructuras matemáticas

Nombre	Audiotexto
Arco-cosecante	“arco cosecante de: ”
Arco-coseno	“ arco coseno de: ”
Arco-cotangente	“ arco cotangente de: ”
Arco-secante	“ arco secante de: ”
Arco-seno	“ arco seno de: ”
Arco-tangente	“ arco tangente de: ”
Cosecante	“ cosecante de: ”
Coseno	“ coseno de: ”
Cotangente	“ cotangente de: ”
Secante	“ secante de: ”
Seno	“ seno de: ”
Tangente	“ tangente de: ”
División	“ dividido para ”
Ecuación	“ igual a ”
Fracción	“ sobre ”
Mayor o igual que	“ mayor o igual que ”
Mayor que	“ mayor que ”
Menor o igual que	“ menor o igual que ”

Continua \implies

Nombre	Audiotexto
Menor que	“ menor que ”
Multiplicación	“ por ”
O Lógica	“ ó ”
Potencia	“ elevado a ”
Resta	“ menos ”
Suma	“ más ”
Y Lógica	“ y ”
Logaritmo	“ logaritmo base [] de ”
Raíz	“ raíz [] de ”
Sub-expresión	“ Inicio 1era expresión [] Fin 1era expresión ”

3.2.7 Accesibilidad

Panel de entrada de texto

Descripción: es un cuadro de texto que conjuntamente con un narrador de pantalla permiten al usuario no vidente editar textos planos. En la tabla 28 se expone cada una de las funciones del panel de entrada de texto con su descripción.

Uso: el usuario puede ingresar, editar o eliminar caracteres del teclado, formar palabras, oraciones, párrafos, entre otros; que conjuntamente con el sintetizador de voz podrá escuchar desde los caracteres hasta todo un documento de texto.

Tabla 28

Funciones del panel de entrada de texto

Función	Detalles
Abrir/Cerrar	Permite al usuario abrir y desplegar información de archivos de texto “.txt” (unicode UTF-8) dentro de este panel que se encuentren almacenados en su computador o cerrar el mismo.
Editar	Permite al usuario ingresar mediante un teclado conectado a su computador ingresar información al panel de entrada de texto, pudiendo añadir, copiar, pegar, modificar o eliminar.
Guardar	Permite al usuario guardar la información del panel de entrada de texto en archivos con extensión “.txt” (unicode UTF-8) en la ubicación predeterminada dentro de su computador.

Continua ⇒

Función	Detalles
Traducir	Permite al usuario convertir la información ingresada en el panel de entrada de texto al sistema braille y presentar la información resultante en el panel braille.
Imprimir	Permite al usuario imprimir la información del panel de entrada de texto en archivos con extensión“.txt” (unicode UTF-8) en la impresora predeterminada para su computador.
Escuchar	Permite al usuario revisar el contenido del panel de entrada de texto al escuchar la información que va ingresando en forma audible mediante el narrador de pantalla. Según el modo de lectura el usuario puede configurar cómo desea escuchar la información por:
No leer	Desactiva la opción de lectura.
Caracter	Lee el caracter al cuando se presiona una tecla.
Palabra	Lee la palabra después de detectar el caracter de espacio en blanco “space”.
Oración	Lee la oración después de detectar el caracter del punto “.”, coma “,” dos puntos “:” o punto y coma “;”.
Párrafo	Lee el párrafo después de detectar el caracter de retorno de carro “enter”.
Todo	Lee todo el documento.

Panel braille

Descripción: es un cuadro de texto para presentar la información requerida en formato braille de seis puntos según la tipografía definida. En la tabla 29 se expone cada una de las funciones de este panel con su descripción.

Uso: el usuario puede revisar antes de imprimir o exportar la conversión de las fórmulas matemáticas al sistema braille a través de mensajes de voz generados por el sintetizador de voz.

Tabla 29

Funciones del panel braille

Función	Detalles
Abrir/Cerrar	Permite al usuario abrir y desplegar información de archivos de texto “.txt” (unicode UTF-8) dentro de este panel que se encuentren almacenados en su computador o cerrar el mismo.

Continua \implies

Función	Detalles
Editar	Permite al usuario ingresar mediante un teclado conectado a su computador ingresar información al panel braille, pudiendo añadir, copiar, pegar, modificar o eliminar.
Guardar	Permite al usuario guardar la información del panel de entrada de texto en archivos con extensión“.txt” (unicode UTF-8) en la ubicación predeterminada dentro de su computador.
Imprimir	Permite al usuario imprimir la información del panel en archivos con extensión“.txt” (unicode UTF-8) en la impresora predeterminada para su computador.

Panel de entrada de expresiones algebraicas

Descripción: posee un contenedor de datos estructurado por niveles o nodos para ingresar cada elemento de una expresión algebraica elemental de manera sistemática (estructuras de datos tipo árbol) ayudando al usuario a reducir la carga cognitiva. Su funcionamiento para ingresar una expresión algebraica en el panel es mediante orden semántico o anidado (operaciones dentro de otras operaciones) inverso de la misma, consecuentemente se debe conocer la estructura de la expresión algebraica para poder ingresarla mediante el panel de entrada. En la tabla 30 se expone cada una de las funciones de este panel con su descripción.

Uso: permite ingresar y editar fórmulas matemáticas, desplazarse en la fórmulas a través de las teclas de dirección y escuchar su contenido con mensajes de voz.

Tabla 30

Funciones del panel de entrada de expresiones algebraicas

Función	Detalles
Abrir/Cerrar	Permite al usuario abrir y desplegar información de archivos de texto “.xml” (unicode UTF-8) dentro de este panel que se encuentren almacenados en su computador o cerrar el mismo.
Guardar	Permite al usuario guardar la información del panel de entrada de texto en archivos con extensión“.xml” (unicode UTF-8) en la ubicación predeterminada dentro de su computador.
Exportar	Permite al usuario exportar la información del panel hacia otros programas de edición (como Editor de Ecuaciones de <i>MS Word</i>) en archivos de texto con extensión “.xml”.

Continúa \Rightarrow

Función	Detalles
Imprimir	Permite al usuario imprimir la información del panel en archivos con extensión“.xml” (unicode UTF-8) en la impresora predeterminada para su computador.
Insertar/Eliminar	Permite al usuario insertar/eliminar operaciones, funciones, estructuras y símbolos al árbol de datos para crear una expresión algebraica llamando al menú dinámico de búsqueda.
Editar	Permite al usuario ingresar mediante un teclado conectado a su computador información a los componentes de las operaciones y funciones dentro del árbol, pudiendo insertar, modificar o borrar información dentro del cuadro de edición.
Localizar	Permite al usuario conocer su localización dentro del árbol de edición.
Traducir	Permite al usuario convertir la información ingresada en el panel de entrada de expresiones algebraicas al sistema braille y presentar la información resultante en el panel braille.
Ver	Permite al usuario convertir la información ingresada en el panel de entrada de expresiones algebraicas a la notación matemática común y presentar la información resultante en el panel de renderizado.
Escuchar	Permite al usuario escuchar la información ingresada en el panel de entrada de expresiones algebraicas y controlar la narración de la información resultante en el panel de audiotexto.

Panel de audiotexto

Descripción: contiene los controles para la reproducción del contenido del panel de entrada de expresiones algebraicas en forma audible. En la Tabla 31 se expone cada una de las funciones de este panel con su descripción.

Uso: permite al usuario escuchar/detener las fórmulas matemáticas ingresadas a través de mensajes de voz usando diferentes voces.

Tabla 31

Funciones del panel de audiotexto

Función	Detalles
Reproducir	Permite al usuario comenzar a escuchar la información del panel de entrada de expresiones algebraicas en forma audible.

Continua \implies

Función	Detalles
Pausar	Permite al usuario interrumpir (para posteriormente volver a escuchar) la reproducción de audio de la información del panel de entrada de expresiones algebraicas.
Detener	Permite al usuario dejar de reproducir completamente el audio de la información del panel de entrada de expresiones algebraicas.

Panel de renderizado

Descripción: contiene la información del panel de entrada de expresiones algebraicas y despliega la misma en notación matemática común. En la Tabla 32 se expone cada una de las funciones de este panel con su descripción.

Uso: permite a los usuarios videntes o con baja visión observar de manera gráfica (o renderizada) como quedarían las fórmulas ingresadas en notación matemática común.

Tabla 32

Funciones del panel de renderizado

Función	Detalles
Ver/Actualizar	Permite convertir la información ingresada a la notación matemática común y presentar la información resultante en el panel de renderizado.

Menú dinámico de búsqueda

Descripción: contiene la lista de todas las operaciones, funciones, estructuras y símbolos para poder ingresar una fórmula dentro del panel de entrada de expresiones algebraicas. Se puede acceder al menú presionando la combinación de teclas (CTRL + SHIFT + I) únicamente desde el panel de entrada de expresiones algebraicas.

Uso: permite al usuario escoger de una lista ordenada alfabéticamente la operación o función matemática (soportada) que quiera ingresar en el panel de entrada de expresiones algebraicas.

Cuadro de edición

Descripción: permite la edición de elementos de las operaciones/funciones por medio del teclado del computador, el modo de lectura por defecto es por caracteres. Se puede acceder al cuadro de edición presionando la tecla (F2) únicamente desde el panel de entrada de expresiones algebraicas.

Uso: el usuario podrá editar los elementos de cada operación o función matemática (soportada) que quiera editar o modificar.

Comandos por teclado

Descripción: permite al usuario no vidente acceder a las funciones propias del programa por medio de teclas rápidas o combinación de teclas, las cuales se muestran en las Tablas 33 y 34.

Uso: permiten a los usuarios no videntes realizar las mismas funciones del programa, sin el uso de botones y del ratón.

Tabla 33**Comandos rápidos por teclado por defecto**

Atajo por teclado	Función
Ctrl+A	Abrir
Ctrl+C	Copiar
Ctrl+F4	Cerrar
Ctrl+G	Guardar
Ctrl+P	Imprimir
Ctrl+V	Pegar
Ctrl+X	Cortar
Enter	Aceptar
Esc	Cancelar
Alt izq.+Shift izq.+Impr. Pant.	Activar/desactivar modo alto contraste
Alt+Flecha Izquierda	Mostrar el grupo previo en el control TreeView
Alt+Flecha Derecha	Mostrar el grupo siguiente en el control TreeView

Tabla 34**Comandos rápidos por teclado nuevos**

Atajo por teclado	Función
Ctrl+Shift+0	No leer
Ctrl+Shift+1	Leer carácter
Ctrl+Shift+2	Leer palabra
Ctrl+Shift+3	Leer oración
Ctrl+Shift+4	Leer párrafo
Ctrl+Shift+5	Leer todo
Ctrl+Shift+6	Subir volumen
Ctrl+Shift+7	Bajar volumen
Ctrl+Shift+8	Subir velocidad

Continua \Rightarrow

Atajo por teclado	Función
Ctrl+Shift+9	Bajar velocidad
Ctrl+Shift+E	Eliminar operación
Ctrl+Shift+I	Insertar operación
Ctrl+Shift+L	Magnificador de pantalla
F1	Comandos rápidos
F2	Editar nodo cuadro de edición
F3	Localizar nodo
F4	Generar render
F5	Reproducir
F6	Pausar
F7	Detener todo audio
F8	Generar braille
F9	Identificar controles
TAB	Mover foco

3.3 Algoritmos

Los algoritmos este programa fueron diseñados para convertir los datos ingresados por el usuario no vidente a las representaciones: braille, visual, auditiva.

3.3.1 Diagramas de flujo

Renderización

Proceso para obtener una representación de forma renderizada de una expresión algebraica ingresada por el usuario.

- **Funcionamiento general:** en el diagrama de flujo de la figura 18 se describe los pasos que realiza este proceso de conversión.
- **Traslado y procesamiento de información:** en el diagrama de flujo de la figura 19 se muestra como dependiendo del formato de conversión elegido se procede a realizar su tratamiento.

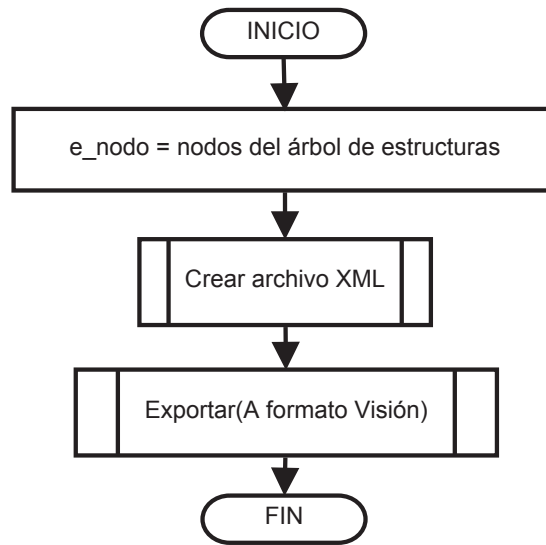


Figura 18 Funcionamiento general del proceso de renderización

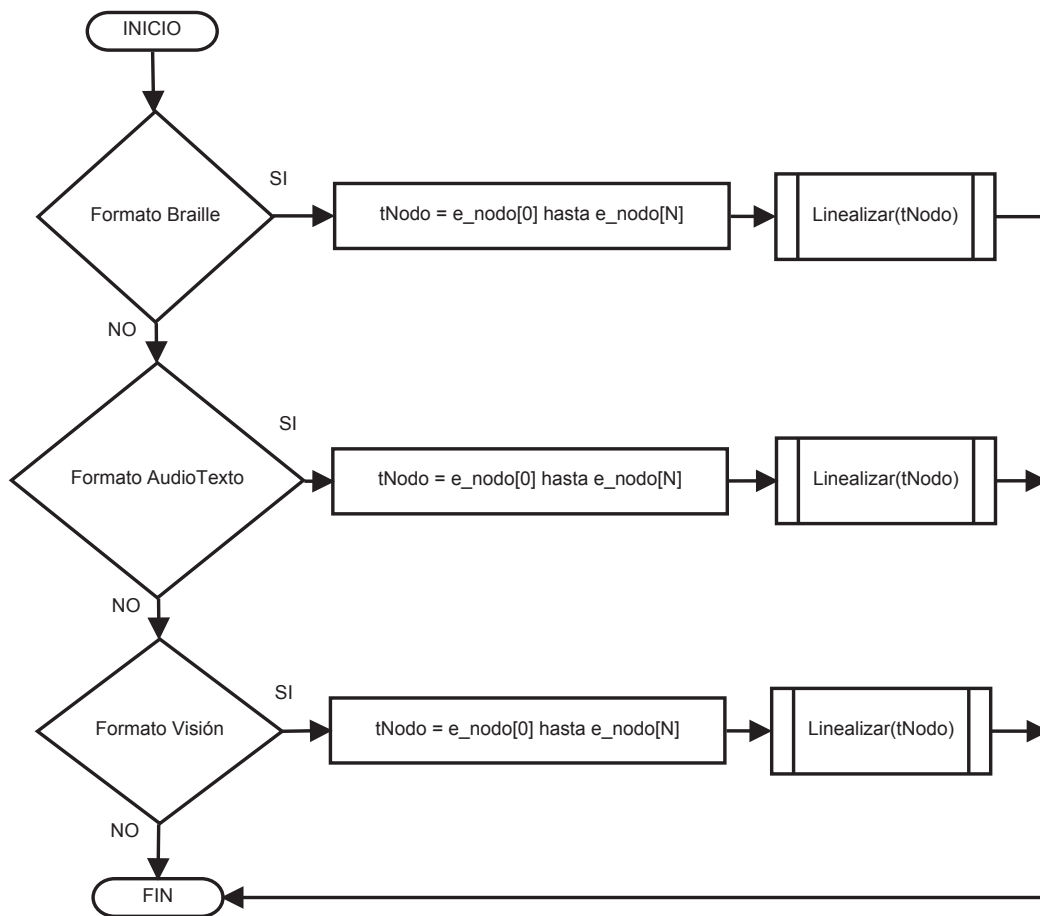


Figura 19 Proceso de exportación

En el diagrama de la figura 20 se muestra como se procede a reordenar de manera recursiva los datos del árbol de estructuras, es decir sus nodos dependiendo del tipo de operación o función matemática de la que se trate y también dependiendo del tipo del formato de conversión escogido.

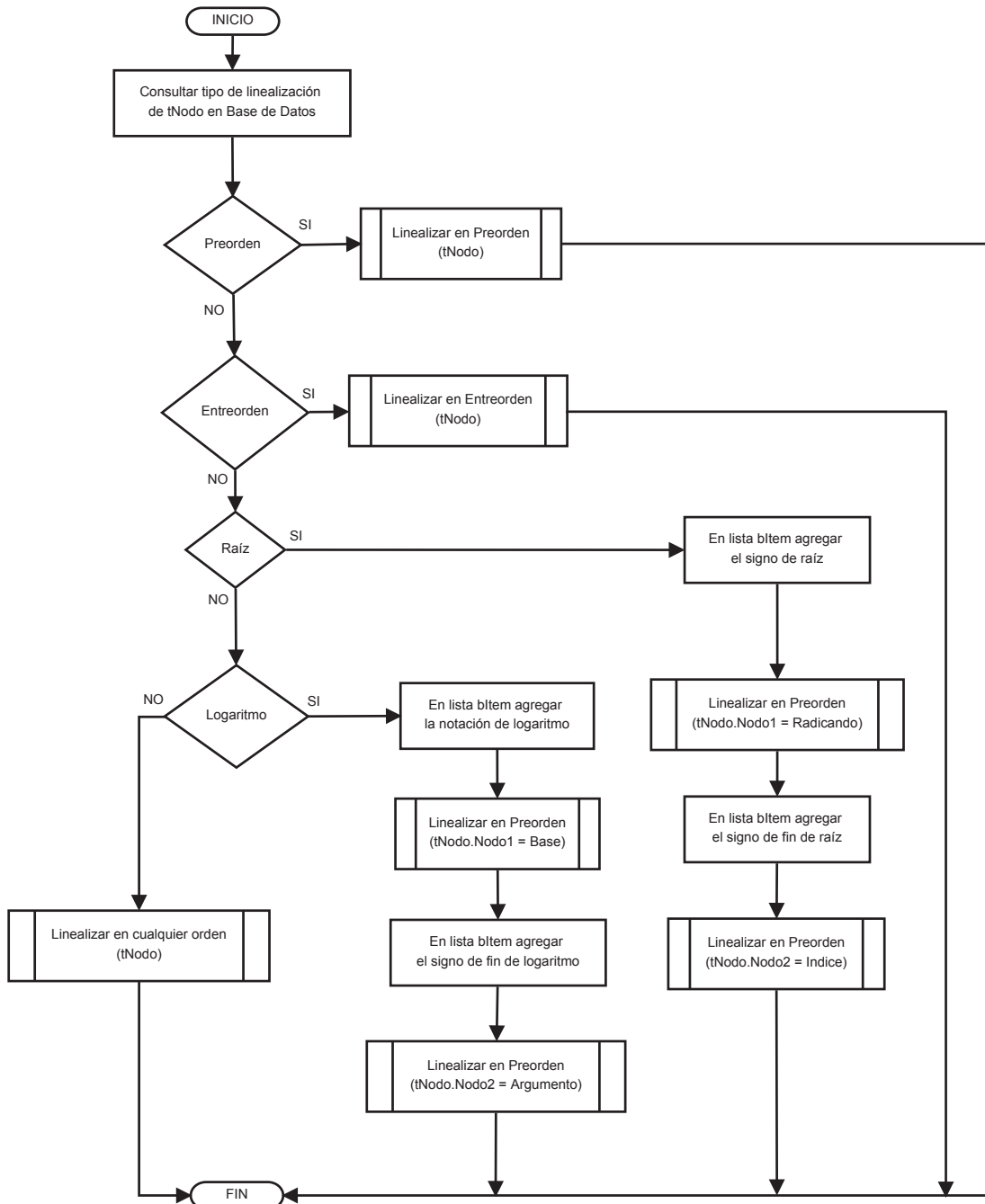


Figura 20 Proceso de linealización

Existe dos tipos de reordenamiento de estructuras matemáticas:

Entre-orden: cuando el símbolo o nombre de la estructura matemática se entre sus sub-componentes que se describe en el diagrama de flujo de la figura 21, por ejemplo: $w * z$ la notación está en entre-orden porque primero se escribe un sub-componente de la operación seguida del símbolo de la multiplicación y finalmente otro sub-componente de la misma operación.

Pre-orden: cuando el símbolo o nombre de la estructura matemática se escribe antes que sus sub-componentes que se describe en el diagrama de flujo de la figura 22, por ejemplo: $\tan(x+y)$ la notación está en pre-orden porque primero se escribe una abreviación del nombre de la función: “tan” seguida entre paréntesis de sus argumento “(x+y)”.

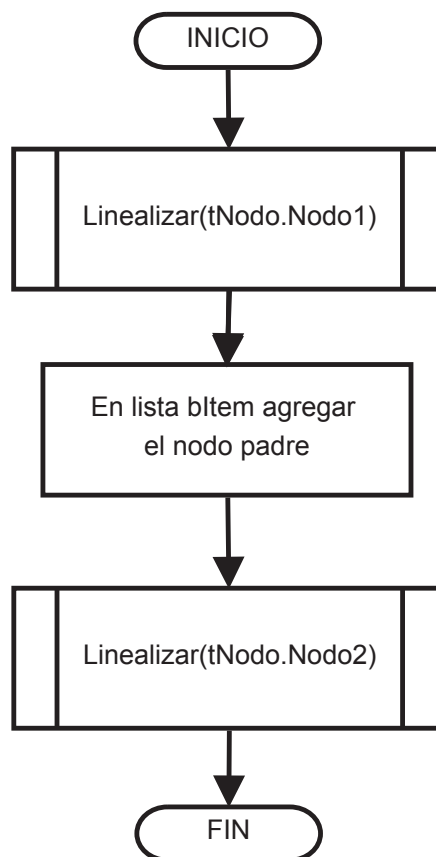


Figura 21 Proceso de reordenamiento entre-orden

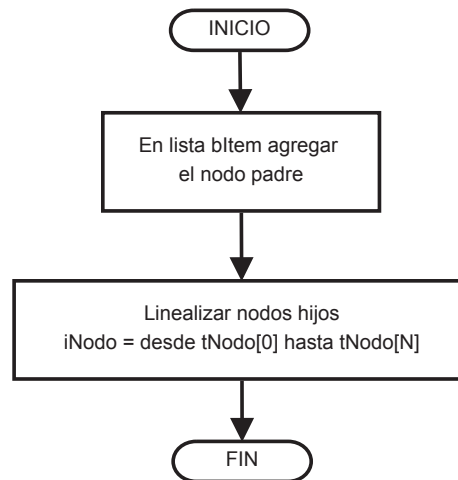


Figura 22 Proceso de reordenamiento pre-orden

- **Creación del documento:** en el diagrama de flujo de la figura 23 se muestra la creación del archivo que contendrá la información ya procesada y convertida, en el cual se configura ciertos parámetros del formato del archivo.

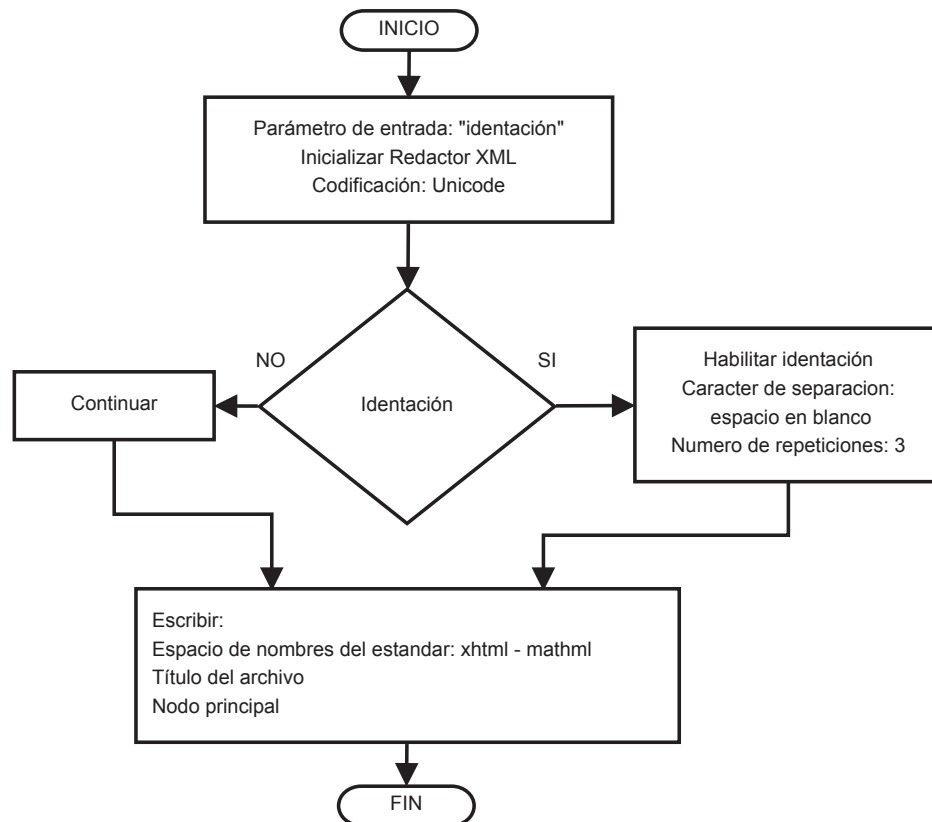


Figura 23 Creación y configuración del archivo de salida

Audiotexto

Proceso para obtener una representación de forma auditiva verbalizada de una expresión algebraica ingresada por el usuario.

- **Funcionamiento general:** en el diagrama de flujo de la figura 25 se describe los pasos que realiza este proceso de conversión.
- **Traslado y procesamiento de información:** en los diagramas de flujo de las figuras 19, 20, 21 y 22 se muestra como dependiendo del formato de conversión elegido se procede a realizar su tratamiento nuevamente.
- **Preparación de los mensajes de voz:** en el diagramas de la figura 24, se muestra el proceso de configuración del sintetizador de voz y la preparación para verbalizar las expresiones algebraicas intercambiando el tono de voz entre masculino y femenino.

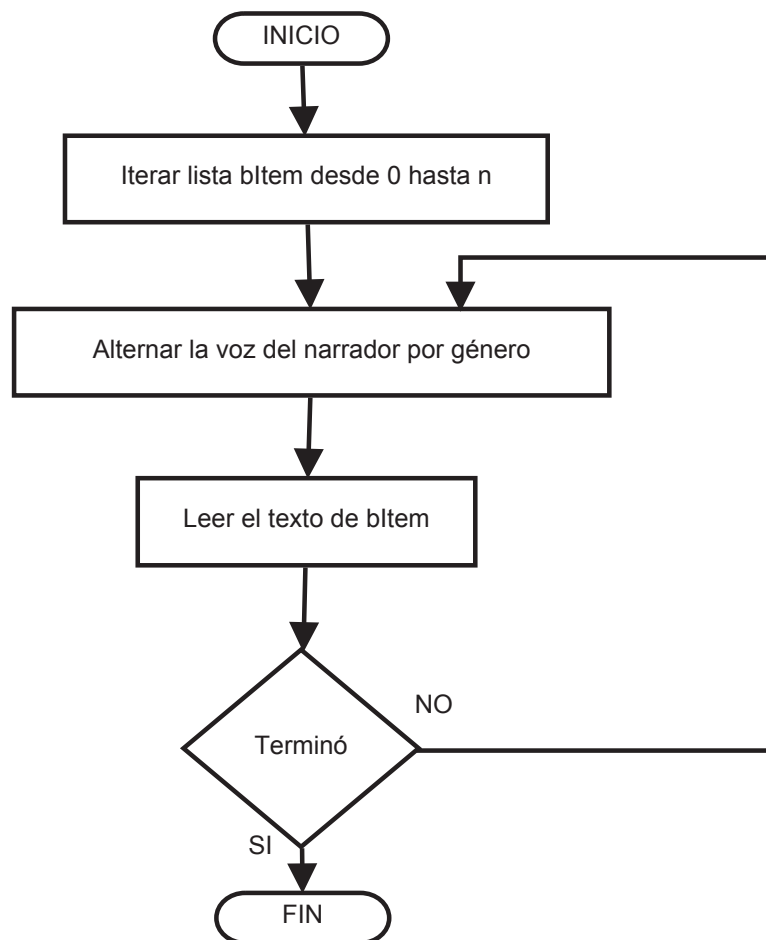


Figura 24 Preparación de los mensajes de voz

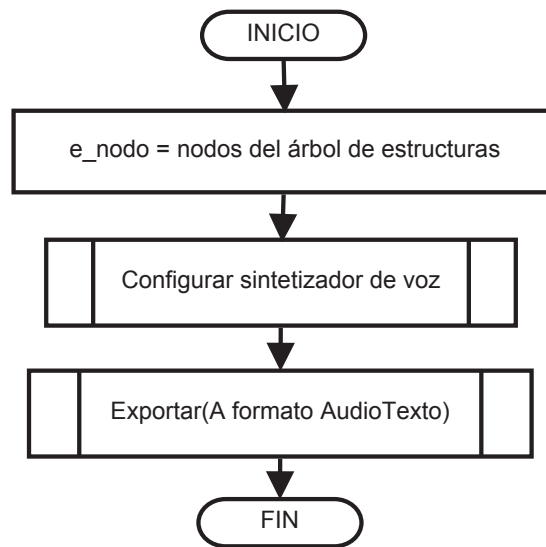


Figura 25 Funcionamiento general de conversión a audiotexto

Braille

Proceso para obtener una representación en el sistema braille de una expresión algebraica ingresada por el usuario.

- **Funcionamiento general:** en el diagrama de flujo de la figura 26 se describe los pasos que realiza este proceso de conversión.

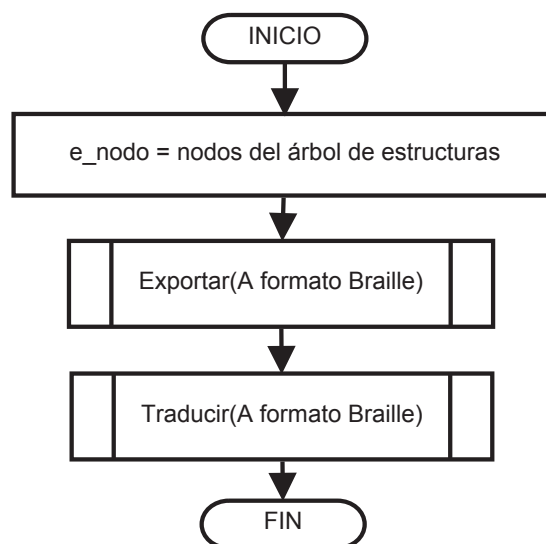


Figura 26 Funcionamiento general de conversión a braille

- **Traslado y procesamiento de información:** en los diagramas de flujo de las

figuras 19, 20, 21 y 22 se muestra como dependiendo del formato de conversión elegido se procede a realizar su tratamiento nuevamente.

- **Proceso de adaptación y traducción:** en el diagramas de la figura 27 se muestra el orden de ejecución de los procesos de adaptación y traducción al braille.

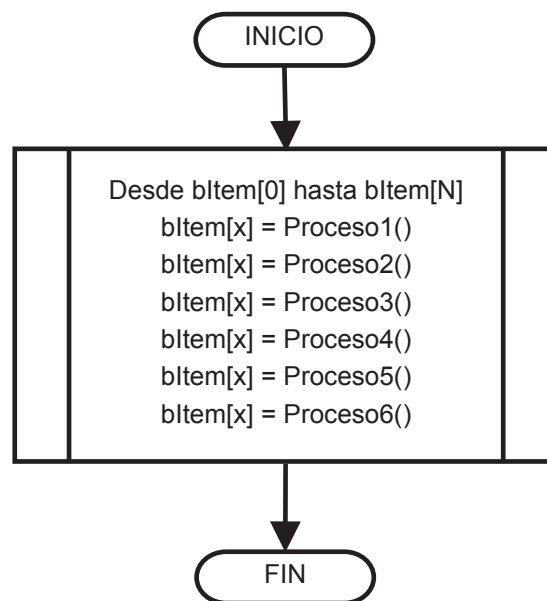


Figura 27 Proceso de adaptación y traducción

En el diagrama de flujo de la figura 28 muestra el proceso para traducir las palabras reservadas es decir los nombres de las operaciones y funciones matemáticas por sus respectivas notaciones en braille.

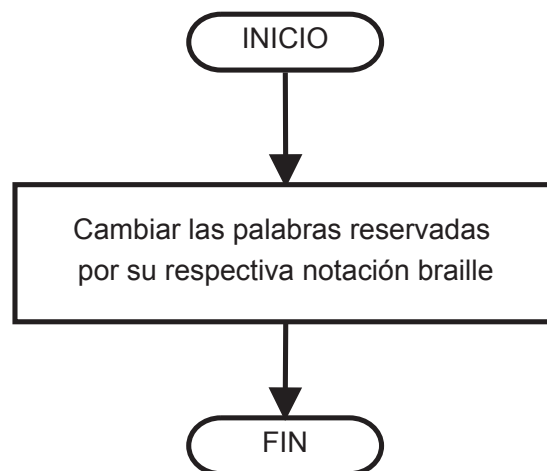


Figura 28 Proceso de traducción N1

En el diagrama de flujo de la figura 29 muestra el proceso para incluir un carácter prefijo para letras mayúsculas en braille.

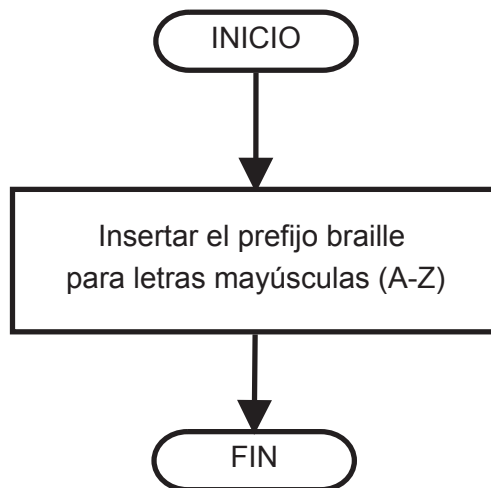


Figura 29 Proceso de traducción N2

En el diagrama de flujo de la figura 30 muestra el proceso para incluir un carácter prefijo para algunas letras del abecedario para no confundirlas con los números en braille.

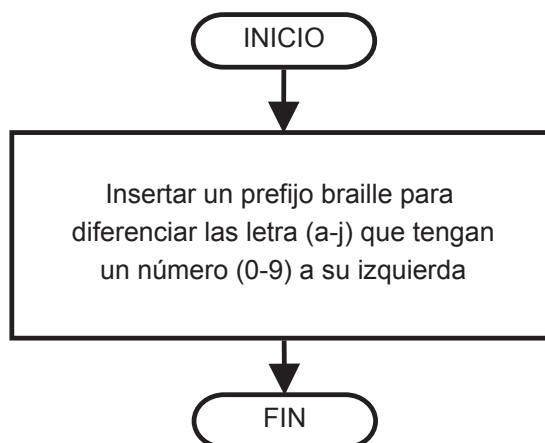


Figura 30 Proceso de traducción N3

En el diagrama de flujo de la figura 31 muestra el proceso para incluir un carácter prefijo para los números en braille.

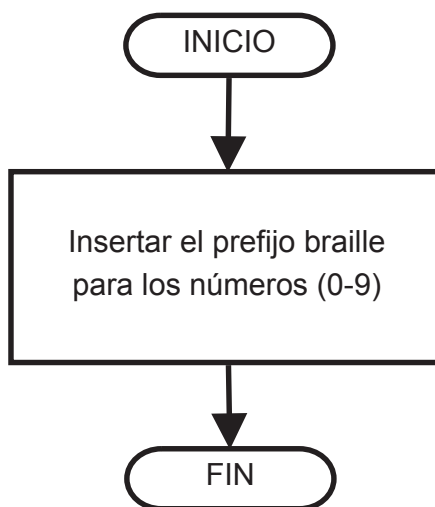


Figura 31 Proceso de traducción N4

En el diagrama de flujo de la figura 32 muestra el proceso para traducir con equivalentes braille con dos símbolos.

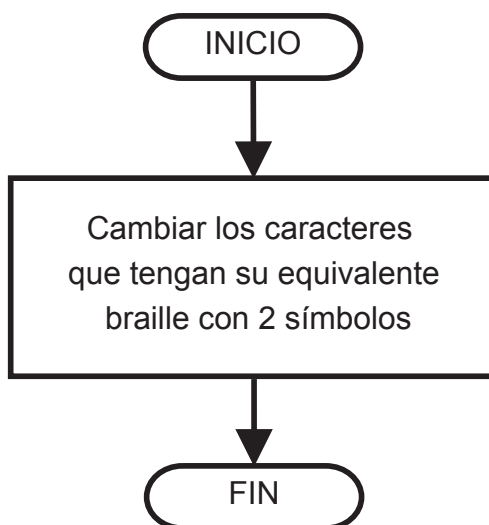


Figura 32 Proceso de traducción N5

En el diagrama de flujo de la figura 33 muestra el proceso para traducir con equivalentes braille con un símbolo.

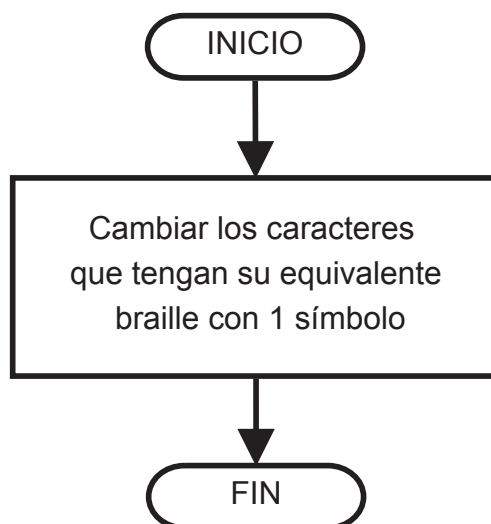


Figura 33 Proceso de traducción N6

3.3.2 Diagramas UML

Siguiendo el patrón modelo-vista-controlador se ha organizado todas las funciones y eventos del programa:

Modelo: Una expresión algebraica está conformada por estructuras matemáticas, las estructuras matemáticas a su vez por elementos y estos a su vez al ser traducidos al braille tienen sus propios caracteres. Esa es la razón por la cual se ha creado la clase “Estructura” (figura 36), la clase “Elemento” (figura 35) y “Caracteres” (figura 34) que utilizan recursos comunes cuyos modelos permiten la creación sistemática y organizada de expresiones algebraicas.

Vista: La ventana principal de edición es controlada por la clase “Usuario” que se encarga de controlar eventos, los cuales se describen figura 40 a fin de asistir al usuario no vidente en la edición de expresiones algebraicas dentro de esta ventana.

Las ventanas de diálogo para abrir, guardar e imprimir archivos son cuadros de diálogo simples para que un usuario no vidente pueda realizar estas actividades sin mayor dificultad. Por esa razón se han creado las clases “Abrir”, “Guardar” e “Imprimir” cuyos modelos controlan los eventos propios de estas ventanas, en las figuras 37, 38 y 39 se describen los modelos de estas clases.

Controlador: el control de la ejecución principal del programa y de los mensajes de voz por medio del sintetizador de voz se encuentran definidos por las clases “Principal” (figura 42) y “Audio” (figura 41) se describen sus métodos y datos miembro.

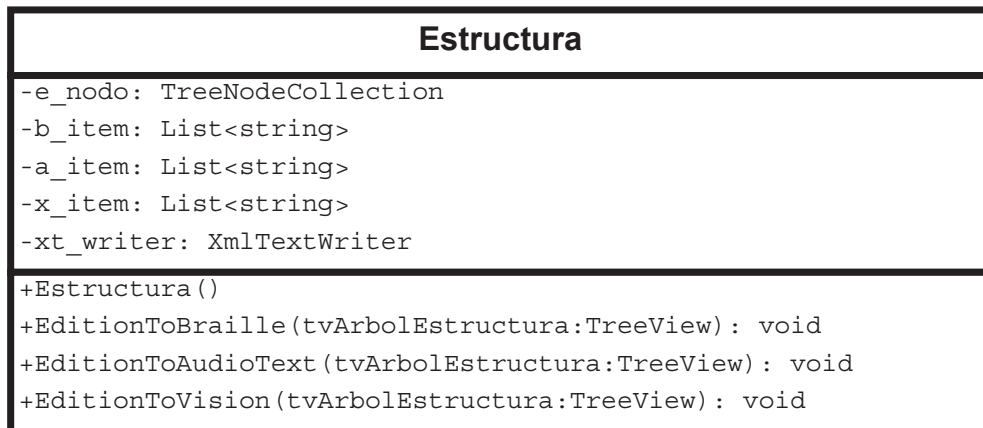


Figura 34 Diagrama de la clase Estructura

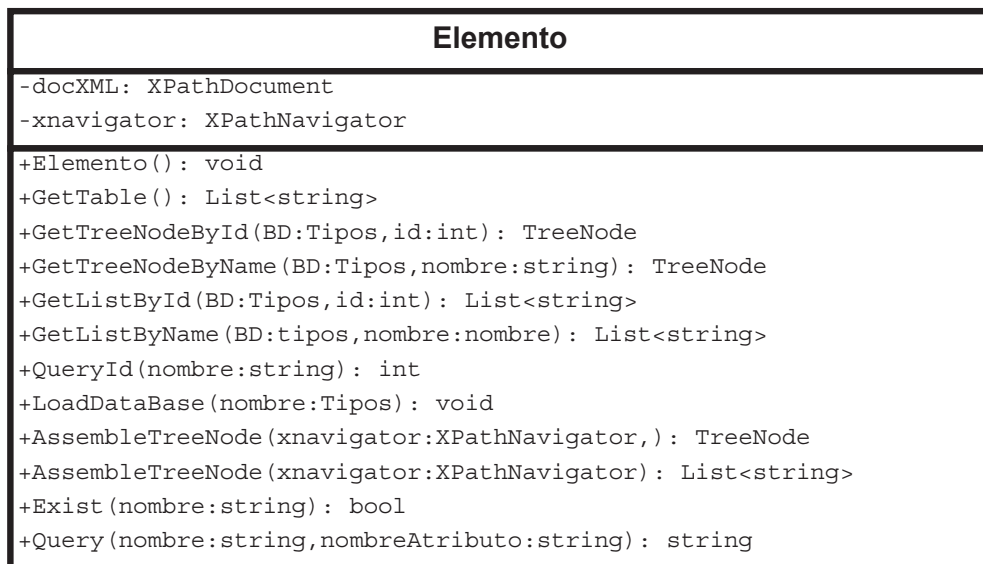


Figura 35 Diagrama de la clase Elemento

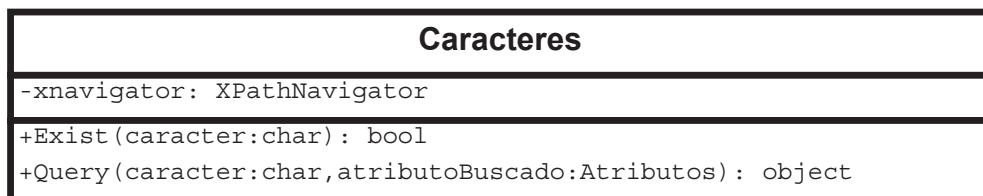
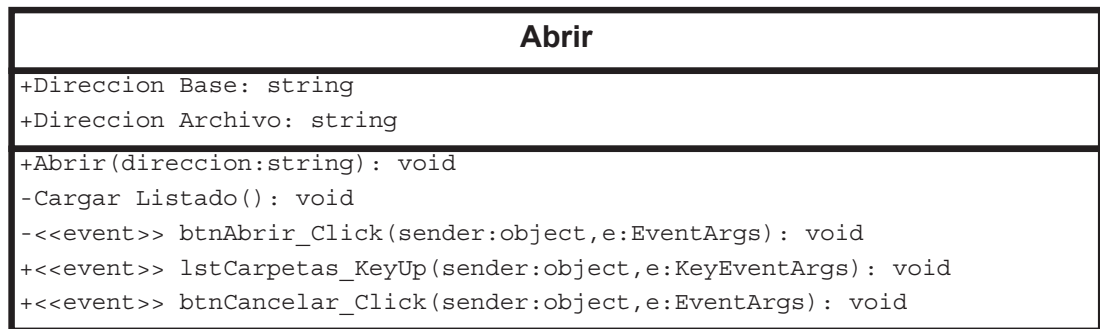
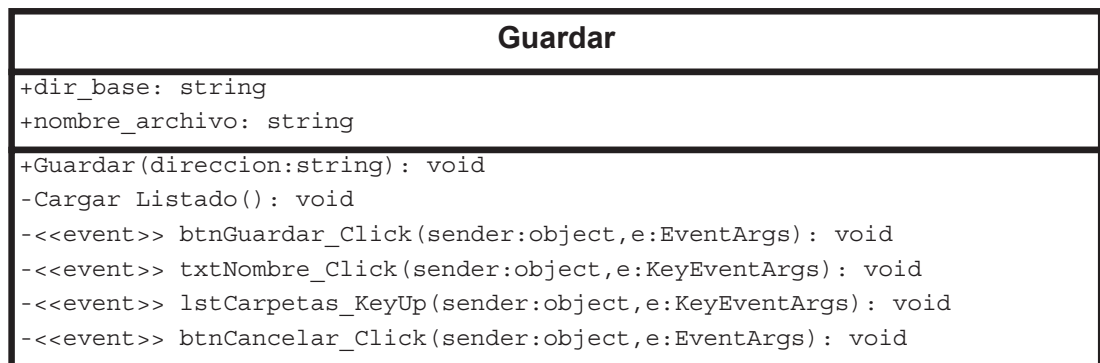
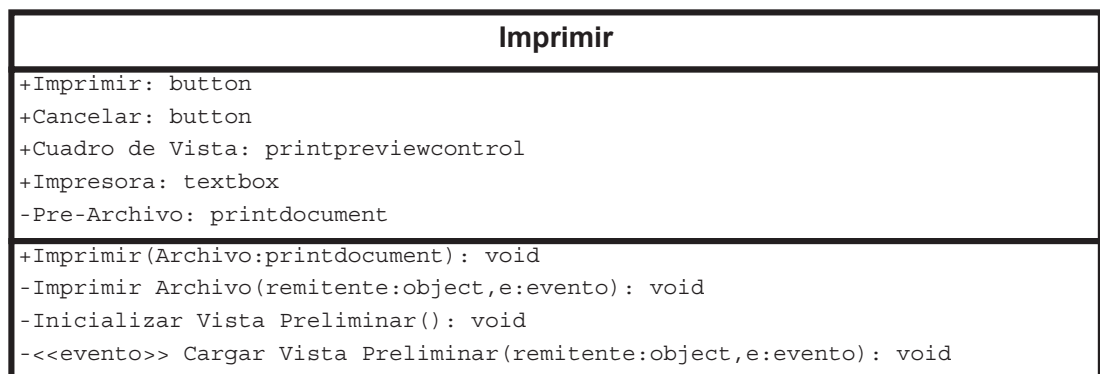


Figura 36 Diagrama de la clase Caracter

**Figura 37 Diagrama de la clase Abrir****Figura 38 Diagrama de la clase Guardar****Figura 39 Diagrama de la clase Imprimir**

Usuario
-Ventana abrir: Abrir -Ventana guardar: Guardar -Ventana imprimir: Imprimir -v_AcercaDe: AcercaDe
<pre> +Usuario(): void -InicializarConfiguracionesUsuario(): void -<<event>> b_Abrir_Click(sender:object,e:EventArgs): void -<<event>> b_Imprimir_Click(sender:object,e:EventArgs): void -<<event>> b_Guardar_Click(sender:object,e:EventArgs): void -<<event>> b_Salir_Click(sender:object,e:EventArgs): void -<<event>> b_Nuevo_Click(sender:object,e:EventArgs): void -<<event>> b_Cerrar_Click(sender:object,e:EventArgs): void -<<event>> b_Aceptar_Imprimir(Sender:object,e:PrintEvent): void -TreeViewToXML(path:string): void -SaveNode(Nodo:TreeNode): void -XMLToTreeView(path:string,trv:TreeView): void -OpenNode(parent_nodes:TreeNodeCollection,xml_nodes:XmlNode): void -<<event>> bInsertar_Click(sender:object,e:EventArgs): void -<<event>> bEliminar_Click(sender:object,e:EventArgs): void -<<event>> bModificar_Click(sender:object,e:EventArgs): void -<<event>> bLocalizar_Click(sender:object,e:EventArgs): void -<<event>> bGenerarRender_Click(sender:object,e:EventArgs): void -<<event>> bAudiotexto_Click(sender:object,e:EventArgs): void -<<event>> bDetener_Click(sender:object,e:EventArgs): void -<<event>> bPausar_Click(sender:object,e:EventArgs): void -<<event>> bBraille_Click(sender:object,e:EventArgs): void -<<event>> bLeerCaracteres_Click(sender:object,e:EventArgs): void -<<event>> bLeerPalabras_Click(sender:object,e:EventArgs): void -<<event>> bLeerOraciones_Click(sender:object,e:EventArgs): void -<<event>> bLeerParrafos_Click(sender:object,e:EventArgs): void -<<event>> bLeerTodo_Click(sender:object,e:EventArgs): void -<<event>> bSubirVolumen_Click(sender:object,e:EventArgs): void -<<event>> bBajarVolumen_Click(sender:object,e:EventArgs): void -<<event>> bSubirVelocidad_Click(sender:object,e:EventArgs): void -<<event>> bBajarVelocidad_Click(sender:object,e:EventArgs): void -<<event>> bActivarLeerVozAlta_Click(sender:object,e:EventArgs): void -<<event>> bDesactivarLeerVozAlta_Click(sender:object,e:EventArgs): void -<<event>> bAcercaDe_Click(sender:object,e:EventArgs): void -<<event>> bAyudaPrograma_Click(sender:object,e:EventArgs): void -<<event>> mcBusquedaDinamica_KeyDown(sender:object,e:KeyEventArgs): void -<<event>> mcBusquedaDinamica_KeyUp(sender:object,e:KeyEventArgs): void -<<event>> tvArbolVista_KeyUp(sender:object,e:KeyEventArgs): void -<<event>> txtCuadroEdicion_KeyPress(sender:object,e:KeyPressEventArgs): void -<<event>> txtCuadroEdicion_KeyDown(sender:object,e:KeyEventArgs): void -<<event>> btnOK_Click(sender:object,e:EventArgs): void -<<event>> rtbTinta_PreviewKeyDown(sender:object,e:PreviewKeyDownEvent): void -<<event>> rtbTinta_KeyDown(sender:object,e:KeyEventArgs): void -<<event>> rtbTinta_KeyUp(sender:object,e:KeyEventArgs): void -<<event>> rtbTinta_KeyPress(sender:object,e:KeyPressEventArgs): void </pre>

Figura 40 Diagrama de la clase Usuario

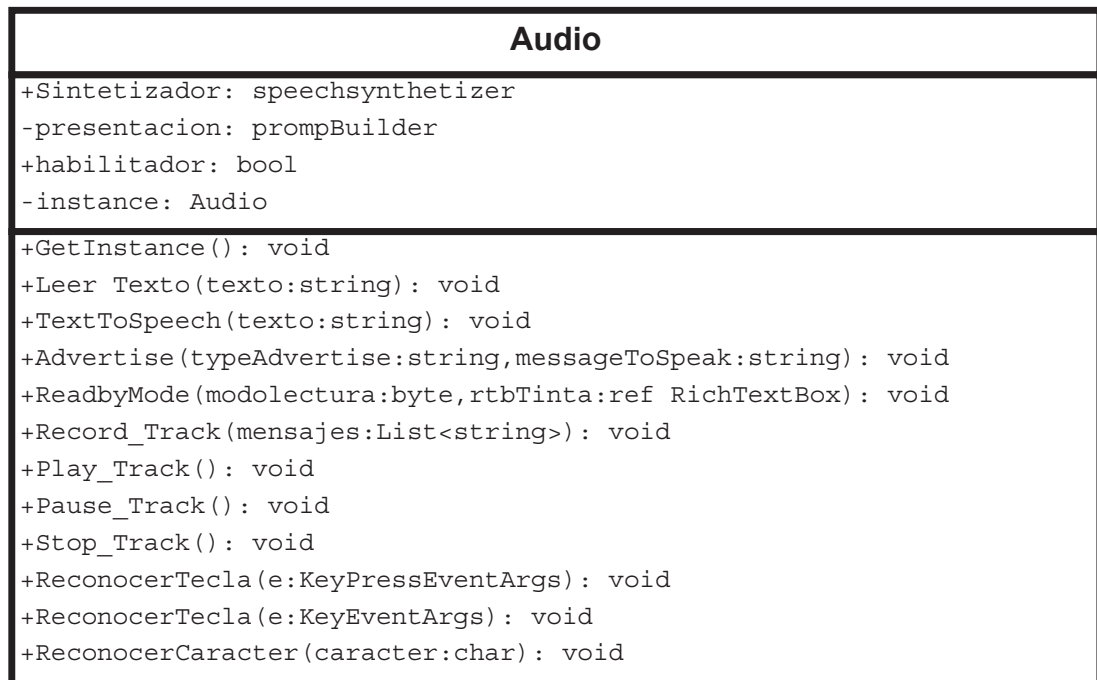


Figura 41 Diagrama de la clase Audio

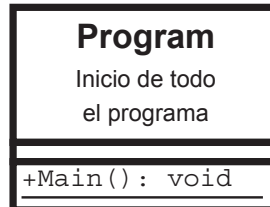


Figura 42 Diagrama de la clase Principal

Capítulo 4

Implementación

En este capítulo se documenta la implementación del programa según el diseño expuesto en el capítulo anterior. Los programas de terceros y/o marcas comerciales que se mencionan en la Tabla 35 fueron utilizadas para el desarrollo de este proyecto con licencia estudiantil o versiones de prueba temporales a fin de evitar problemas con las licencias de uso.

El lenguaje de programación utilizado fue Visual C# para utilizar los recursos locales que brinda el sistema operativo de Microsoft y reducir el tamaño del programa, también se escogió al sintetizador de voz de IVONA por tener una aproximación de la voz semejante a la humana y el navegador Gecko que puede ser embebido dentro de una aplicación de C# por permitir utilizar el estándar MathML.

Tabla 35

Herramientas de Software

	Nombre	Versión
Entorno de desarrollo integrado (IDE)	Microsoft Visual Studio 2010 Express	10.0.30319.1 RTMRel
Kit de Desarrollo de Software (SDK)	Microsoft .NET Framework	4.5.50709 RTMRel
Lenguaje de programación	Microsoft Visual C#	2010
Licencia	Estudiantil	- - -
Lenguaje de marcado	MathML	3
Navegador Web Embebido	GeckoFX	14
Sintetizador de Voz	IVONA Penélope y Miguel	1.1

4.1 Programación de la interfaz de usuario

Antes de comenzar con la implementación de los algoritmos se requirió en primer lugar comenzar con la interfaz de usuario (de ahora en adelante IU) necesaria tanto

como para ejecutar pruebas de funcionamiento.

Las UI implementadas, se muestran a continuación con capturas de pantalla:

- Ventana del Editor

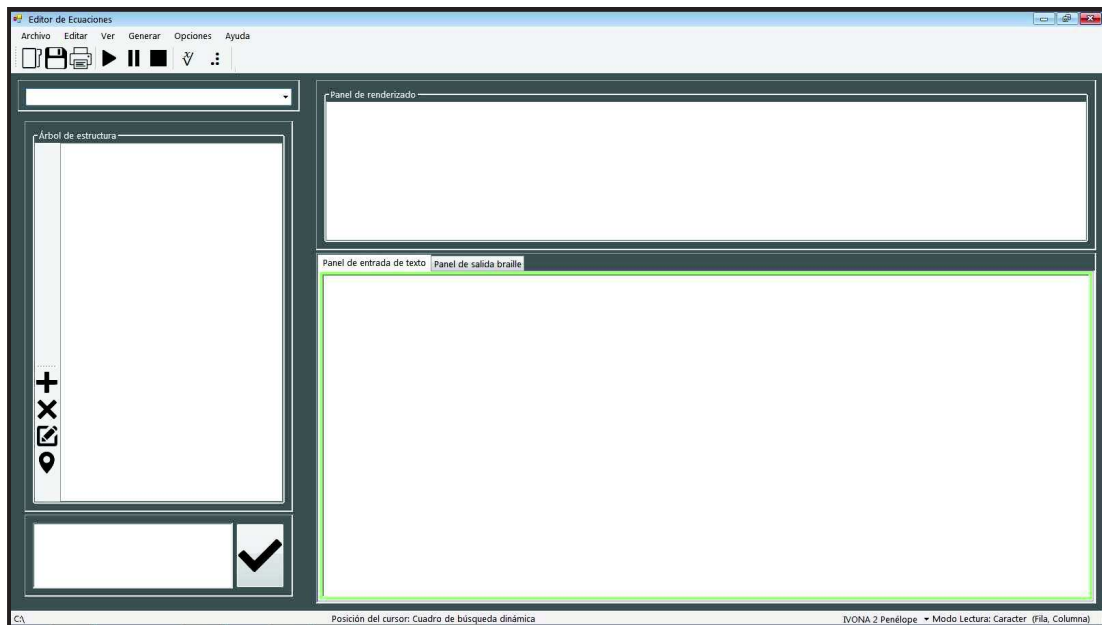


Figura 43 Ventana del editor implementada

- Ventana Abrir

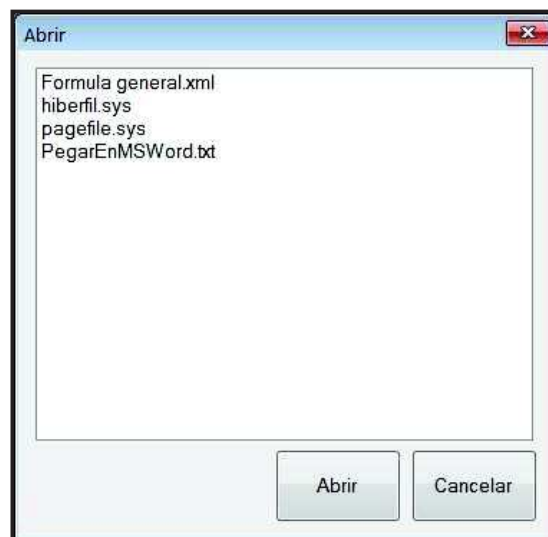


Figura 44 Ventana abrir implementada

- Ventana Guardar como



Figura 45 Ventana guardar implementada

- Ventana Imprimir

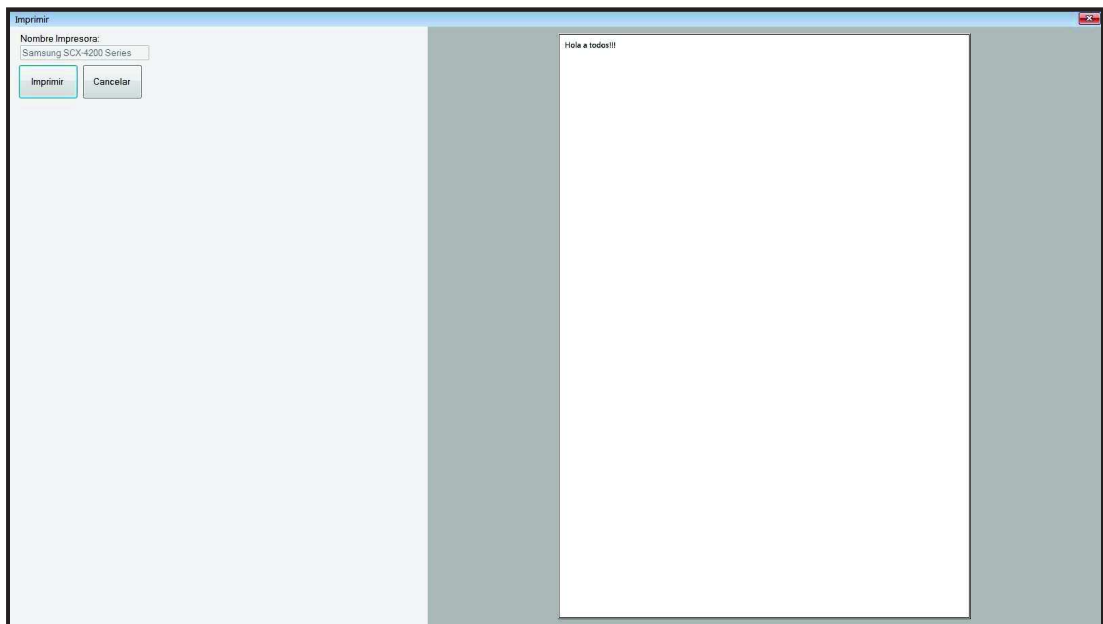


Figura 46 Ventana imprimir implementada

4.2 Programación de los algoritmos

Las funciones especiales del programa fueron implementadas a partir de los diagramas de flujo:

- Implementación de las funciones de conversión a representación braille, audio y visual.

```
public void EditionToBraille(TreeView
    tvArbolEstructura)
{
    //Paso 1: Carga los datos del usuario que ingreso
        en los campos del panel de entrada de
        estructuras matematicas
    eNodo = tvArbolEstructura.SelectedNode;

    //Paso 2: Exporta los campos a una pila o lista
        para procesamiento
    Export(Tipos.Braille);

    //Paso 3: Traduce los campos mediante un
        algoritmo
    Translate();
}
```

```
public void EditionToAudioText(TreeView
    tvArbolEstructura)
{
    cont = 0;
    //Paso 1: Carga los datos del usuario que ingreso
        en los campos del panel de entrada de
        estructuras matematicas
```



```
eNodo = tvArbolEstructura.SelectedNode;

//Paso 2: Exporta los campos a una pila o lista
           para procesamiento
Export (Tipos.AudioText);
}

public void EditionToVision(TreeView
tvArbolEstructura, string path)
{
//Paso 1: Carga los datos del usuario que ingreso
           en los campos del panel de entrada de
           estructuras matematicas
eNodo = tvArbolEstructura.SelectedNode;

//Paso 2: Crear archivo .xhtml (FIREFOX) o .xml (MS
           WORD)
CreateXmlFile(path, true);

//Paso 3: Escribir el encabezado
WriteHeading(true);

//Paso 4: Escribir el cuerpo
Export (Tipos.Vision);

//Paso 5: Escribir el pie de pagina
WriteEnding(true);
}
```

- Implementación de las funciones para la creación del archivo xhtml para la re-

presentación visual:

```
public void CreateXmlFile(string fileName, bool
    indentacion)
{
    //Crea el documento con la direccion y la
        codificación especificadas
    XT_Writer = new XmlTextWriter(fileName, System.
        Text.Encoding.UTF8);

    if (indentacion)
    {
        //Se habilita las indentaciones en el
            documento
        XT_Writer.Formatting = Formatting.Indented;

        //Se establece el caracter ( ' ' = espacio en
            blanco) que hará la separación
        XT_Writer.IndentChar = ' ';

        //Se establece cuantas veces se utilizará el
            caracter de sangría
        XT_Writer.Indentation = 3;
    }
    else
    {
        //Sin tabulaciones o indentaciones
        XT_Writer.Formatting = Formatting.None;
    }
}
```

- Implementación de las funciones de configuración del sintetizador de voz para la representación auditiva:

```
/// <summary>
/// Graba en un objeto la cadena de texto equivalente
/// a la expresión algebraica ingresada por el
    usuario
/// </summary>
/// <param name="mensajes"></param>
public void Record_Track(List<string> mensajes)
{
    bool p = false; //false = Miguel, true = Penélope
    //Cancela todas las reproducciones de audio
    sintetizador.SpeakAsyncCancelAll();

    //Instancia una presentacion como por medio de un
        guión de teatro
    presentacion = new PromptBuilder();
    presentacion.StartVoice(VoiceGender.Male);

    foreach (string mensaje in mensajes)
    {
        if (mensaje != null) //Valida no hay
            mensaje de texto
        {
            //Paso 1: Selecciona la voz cada vez que
                inicia una nueva expresión
            if (mensaje.Contains("Inicio"))
            {
                if (p)
```

```

        {
            presentacion.EndVoice();
            presentacion.StartVoice(
                VoiceGender.Male);
            p = false;
        }
        else
        {
            presentacion.EndVoice();
            presentacion.StartVoice(
                VoiceGender.Female);
            p = true;
        }
    }
    //Paso 2: Prepara el mensaje
    presentacion.AppendText(mensaje,
        PromptEmphasis.None);
    }
}
presentacion.EndVoice();
}

```

- Implementación de la función Exportar:

```

/// <summary>
/// Exporta los campos a una pila o lista para
    procesamiento
/// </summary>
/// <param name="Shape">Representación deseada</param
    >
public void Export(Tipos Shape)

```

```
{
    switch (Shape)
    {
        case Tipos.Braille:
            Linealizar(eNodo, false);
            break;
        case Tipos.AudioText:
            Linealizar_AudioText(eNodo, false);
            break;
        case Tipos.Vision:
            Linealizar_Vision(eNodo, false);
            break;
        default:
            break;
    }
}
```

- Implementación de las función Linealizar:

```
private void Linealizar(TreeNode tNodo, bool brackets
)
{
    string cadena = Elemento.Query(tNodo.Text, "tipo"
);

    if (tNodo.Text.Contains("Expresión"))
    {
        cadena = "x";
    }

    switch (cadena)
```

```

    {
        case "1":
            preorden(tNodo, brackets);
            break;
        case "2":
            inorden(tNodo, brackets);
            break;
        case "raíz":
            //Añade el símbolo de raíz
            bItem.Add(Elemento.Query(tNodo.Text, "
                notación_braille"));
            //Visitar el nodo del índice de la raíz
            inorden(tNodo.Nodes[0], false);
            //Añade el símbolo de fin
            bItem.Add("\u2831");
            //Visitar el nodo del radicando de la raíz
                z
            inorden(tNodo.Nodes[1], false);
            break;
        default:
            //Para sumando, minuendo, base, argumento
            , etc, etc
            inorden(tNodo, true);
            break;
    }
}

```

- Implementación de las funciones linealizar en preorden y entreorden:

```

private void preorden(TreeNode tNodo, bool brackets)
{

```

```

        //Visitar el nodo Padre
        bItem.Add(tNodo.Text);

        //Comprueba si tiene nodos Hijos
        if (tNodo.Nodes.Count == 0)
            return;

        //Visitar nodos Hijos
        foreach (TreeNode iNodo in tNodo.Nodes)
        {
            Linealizar(iNodo, brackets);
        }
    }

private void inorden(TreeNode tNodo, bool brackets)
{
    //Verifica si tiene nodos hijos
    if (tNodo.Nodes.Count > 0)
    {
        //Inserta parentesis de apertura
        if (brackets)
            bItem.Add("{");

        //Visita el nodo Hijo 1
        if (new Elemento().GetTreeNodeByName(Tipos.
            Edition, tNodo.Text) != null)
            //Llaves en braille '{' = B(5)B(123) '}' =
            B(456)B(2)
            Linealizar(tNodo.Nodes[0], true);
    }
}

```

```
else
    Linealizar(tNodo.Nodos[0], false);

//Recorre los demás nodos hijos
for (int i = 1; i < tNodo.Nodos.Count; i++)
{
    //Guarda el nodo Padre
    bItem.Add(tNodo.Text);

    //Visita el siguiente nodo
    if (new Elemento().GetTreeNodeByName(
        Tipos.Edition, tNodo.Text) != null)
        Linealizar(tNodo.Nodos[i], true);
    else
        Linealizar(tNodo.Nodos[i], false);
}

//Inserta parentesis de cierre
if (brackets)
    bItem.Add("}");
}
else
{
    //Guarda un nodo sin hijos
    if (VerificarNodoHoja(tNodo.Text))
        bItem.Add(tNodo.Text);
    else
    {
        bItem.Add("(");
```



```

        bItem.Add(tNodo.Text);
        bItem.Add(" ");
    }
}
}

```

- Implementación de las función Traducir:

```

/// <summary>
/// Traduce los campos mediante las reglas de la
    Comisión de Braille Española
/// </summary>
private void Translate()
{
    bItemTinta = bItem;
    ToString(Tipos.Edition);

    for (int i = 0; i < bItem.Count; i++)
    {
        //Paso 1: Convierte las palabras reservadas
        bItem[i] = Procesos1(bItem[i]);

        //Paso 2: Inserta los prefijos para letras
            latinas mayúsculas
        bItem[i] = Proceso2(bItem[i]);

        //Paso 3: Inserta los prefijos para letras
            precedida de un número
        bItem[i] = Proceso3(bItem[i]);

        //Paso 4: Inserta los prefijos para números
    }
}

```

```

        bItem[i] = Proceso4(bItem[i]);

        //Paso 5: Convierte los caracteres que tienen
            un equivalente de dos signos braille
        bItem[i] = Proceso5(bItem[i]);

        //Paso 6: Convierte los caracteres al braille
        bItem[i] = Proceso6(bItem[i]);
    }
}

```

- Implementación de las subfunciones o procesos de traducción:

```

/// <summary>
/// Convierte las palabras conocidas con sus
    respectivas notaciones
/// </summary>
/// <param name="cadena"></param>
/// <returns></returns>
private string Proceso1(string cadena)
{
    string resultado;
    if (Elemento.Exist(cadena))
    {
        resultado = Elemento.Query(cadena, "notación
            n_braille");
        if (resultado == null)
            return cadena;
        else
            return resultado;
    }
}

```

```
        return cadena;
    }

    /// <summary>
    /// Si el caracter[i-1] es una letra de la 'A' (65) a
        la 'Z' (90)
    /// entonces se inserta el caracter braille B(46) =
        prefijo para letras latinas mayúsculas
    /// en unicode es '\u2828'
    /// </summary>
    /// <param name="cadena"></param>
    /// <returns></returns>
private string Proceso2(string cadena)
{
    char[] escalera = cadena.ToCharArray();
    cadena = "";

    for (int i = 0; i < escalera.Length; i++)
    {
        if (escalera[i] >= 'A' && escalera[i] <= 'Z')
            cadena = cadena + '\u2828' + Caracteres.
                Query(escalera[i], Atributos.Braille);
        else
            cadena = cadena + escalera[i];
    }

    return cadena;
}
```

```
/// <summary>
/// Si el caracter[i-1] es un número del '0' (48) al
/// '9' (57)
/// y el caracter[i] es una letra de la 'a' (97) a la
/// 'j' (106)
/// entonces se inserta el caracter braille B(5) =
/// prefijo para letras latinas minúsculas
/// en unicode es '\u2810'
/// </summary>
/// <param name="cadena"></param>
/// <returns></returns>
private string Proceso3(string cadena)
{
    char[] escalera = cadena.ToCharArray();
    cadena = "" + escalera[0];

    for (int i = 1; i < escalera.Length; i++)
    {
        if ((escalera[i - 1] >= '0' && escalera[i -
            1] <= '9') && (escalera[i] >= 'a' &&
            escalera[i] <= 'j'))
            cadena = cadena + '\u2810' + Caracteres.
                Query(escalera[i], Atributos.Braille);
        else
            cadena = cadena + escalera[i];
    }
    return cadena;
}
```

Capítulo 5

Análisis de Resultados

5.1 Pruebas

5.1.1 Pruebas de la interfaz

Objetivo: comprobar la usabilidad del programa conociendo la respuesta de un usuario no vidente al interactuar con la interfaz de usuario a fin de determinar cómo mejorar la experiencia del usuario final y de la interfaz gráfica.

Descripción: primero el usuario debe capacitarse para conocer de que se trata el programa y cómo puede utilizarlo. Segundo el usuario ingresará, editará, eliminará fórmulas matemáticas, probará las funciones de los paneles del programa.

Primera prueba

Esta prueba fue realizada con la ayuda del doctor en jurisprudencia José Benavides (no vidente), encargado de la sección de no videntes de la biblioteca “Alejandro Segovia” de la Universidad de las Fuerzas Armadas - ESPE, posee habilidad en el manejo del computador y los sintetizadores de voz para lectura de pantalla. Después de haber realizado la prueba, los puntos favorables del programa según Dr. Benavides fueron:

- Se tiene comandos por teclado ya conocidos.
- Existe retroalimentación por mensajes de voz cuando aparecen y desaparecen nuevas ventanas.
- Inclusión del árbol de estructuras.
- No existe latencia en el eco del teclado.
- Uso de recursos propios del sistema operativo.

Y las sugerencias realizadas al programa fueron:

- Incluir en el panel de entrada de “Texto Tinta”, la lectura de caracteres por medio de las flechas de dirección en el teclado.

- Realizar una realimentación por medio de mensajes de voz cuando un nodo del árbol de estructuras haya sido modificado.
- Alineación de los nodos del árbol de estructuras de manera horizontal no vertical.
- Realizar una realimentación por medio de mensajes de voz cuando no existan nodos en el árbol de estructuras.
- Incluir un comando por teclado para silenciar todo tipo de mensajes de voz.
- Diferenciar auditivamente entre las opciones del menú dinámico de búsqueda y los nodos del árbol de estructuras.
- Incluir un magnificador de pantalla y modo de alto contraste.
- Incluir una ayuda de los comandos por teclado.

Segunda prueba

La prueba realizada fue con el licenciado en Matemáticas Wladimir Ganchala (no vidente), actualmente se desempeña como asesor pedagógico particular de personas no videntes y videntes en el área de matemáticas, fue colaborador en la adaptación de textos al sistema braille en la FENCE y posee un habilidad en el manejo del computador. Los puntos favorables del programa según el Lic. Ganchala fueron:

- Se tiene comandos por teclado ya conocidos.
- Inclusión de múltiples voces.
- Inclusión del árbol de estructuras.
- Audiotexto de expresiones algebraicas que requieren de anidación de operaciones.

Y las sugerencias realizadas al programa fueron:

- Incluir varias expresiones o multimodo.
- Incluir también cuadro de edición de nodos, la lectura de caracteres por medio de las flechas de dirección en el teclado.
- Realizar una realimentación por medio de mensajes de voz cuando se inserte o elimine un nodo del árbol de estructuras.

- Permitir al usuario elegir la voz principal del sintetizador de voz.
- Cambiar el mensaje de voz “Inicio expresión 1” y “Fin expresión 1” por “Inicio primera expresión” y “Fin primera expresión” respectivamente.
- Cambiar el mensaje de voz “OR”, “AND” e “Igual” por “Ó Lógica”, “Y Lógica” y “Ecuación” respectivamente.
- No eliminar el contenido previo de un nodo en el árbol de estructuras cuando se modifique, dejar esta decisión al usuario.

Medidas aplicadas

Después de haber realizado estas pruebas se han aplicado las sugerencias obtenidas por parte de los expertos en el programa modificando el diseño de la aplicación a fin de tomar en consideración dichas observaciones.

5.1.2 Pruebas con ejemplos

Objetivo: ingresar ejemplos de expresiones algebraicas que varían en complejidad y tamaño por medio del programa de edición de fórmulas matemáticas para determinar si el programa cumple con los objetivos propuestos del proyecto.

Descripción: para esta prueba se ha utilizado ocho ejemplos de expresiones algebraicas de “MathML Torture Test” [19]. La prueba consiste que el usuario no vidente utilice el programa para ingresar correctamente cada ejercicio propuesto y se cronometra el tiempo transcurrido de inicio a fin.

Una dificultad para realizar esta prueba fue encontrar a un voluntario no vidente sea de nacimiento o no, que tenga habilidad en el manejo de la computadora y las matemáticas. Gracias a la amable ayuda del Dr. Benavides se procedió a realizar esta prueba, los ejercicios de esta prueba se pueden ver en la tabla 36.

Tabla 36

Ejercicios

Núm.	Ejemplo
1	x^2y^2
2	$\frac{x+y^2}{k+1}$
3	$x + y^{k+1}$

Continua \implies

Núm.	Ejemplo
4	$\frac{a}{b/2}$
5	$a + \frac{1}{b + \frac{1}{c + \frac{1}{d + \frac{1}{e}}}}$
6	$\sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + x}}}}}}$
7	2^{2^x}
8	x^{2^y}

5.2 Resultados

Después de haber realizado la prueba, se procede a determinar por lo menos 4 factores de importancia:

- Duración: el tiempo que se demoró para ingresar cada operación (ver tabla 37).
- Cantidad: la cantidad de operaciones por ejercicio (ver tabla 37).
- Velocidad: la cantidad de operaciones por minuto (ver tabla 37).
- Complejidad: la dificultad de cada ejercicio, usando una ponderación (del 1 al 6) a cada operación (ver tabla 38 y 39).

Tabla 37

Duración, número de operaciones y velocidad

Núm. Ejemplo	Hora inicio	Hora fin	Duración (min.)	Cantidad de Operaciones	Velocidad (oper./min.)
1	10:22 AM	10:28 AM	7	3	0,429
2	10:36 AM	10:44 AM	9	4	0,444
3	10:48 AM	10:55 AM	8	5	0,625
4	10:59 AM	11:01 AM	3	2	0,667
5	11:09 AM	10:16 AM	8	8	1,000
6	11:20 AM	10:29 AM	10	12	1,200
7	11:39 AM	10:43 AM	5	3	0,600
8	11:44 AM	11:49 AM	6	2	0,333

Tabla 38

Complejidad ejercicios 1-4

	Ejm. 1	Ejm. 2	Ejm. 3	Ejm. 4
Suma (1 pto.)	0	2	2	0
Multiplicación (2 ptos.)	1	0	0	0
División (3 ptos.)	0	0	0	1
Fraccion (4 ptos.)	0	1	1	1
Potencia (5 ptos.)	2	1	1	0
Raíz (6 ptos.)	0	0	0	0
Total	12	11	11	7

Tabla 39

Complejidad ejercicios 5-8

	Ejm. 5	Ejm. 6	Ejm. 7	Ejm. 8
Suma (1 pto.)	4	6	0	0
Multiplicación (2 ptos.)	0	0	0	1
División (3 ptos.)	0	0	0	0
Fraccion (4 ptos.)	4	0	0	0
Potencia (5 ptos.)	0	0	3	1

Continua \Rightarrow

	Ejm. 5	Ejm. 6	Ejm. 7	Ejm. 8
Raíz (6 ptos.)	0	6	0	0
Total	20	42	15	7

Al examinar estos parámetros se pudo notar que:

- Por cada ejercicio el usuario se demoró en promedio 7 minutos.
- Por cada ejercicio el usuario ingreso en promedio 4,875 operaciones.
- Cada ejercicio tuvo una dificultad promedio 15,625 puntos.
- Por cada minuto el usuario ingresó 0,696 operaciones o por cada operación el usuario se demoró 1 minuto con 26 segundos.

Mediante el gráfico de la figura 47 se puede observar simultáneamente 3 parámetros: duración, complejidad y cantidad.

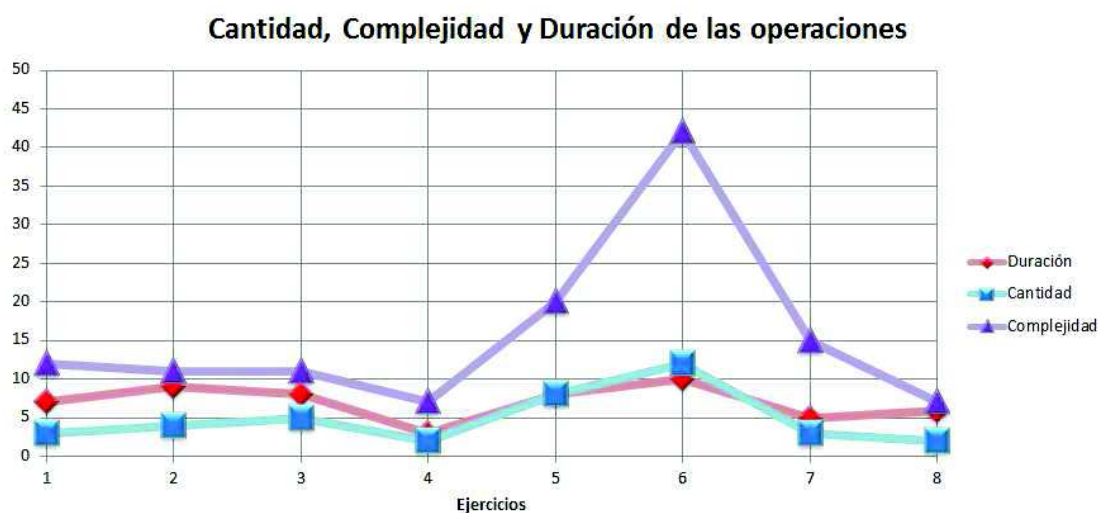


Figura 47 Gráfico Parámetros por Ejercicio

Al examinar esta gráfica se pudo notar que:

- Existe una relación proporcional entre las curvas de duración y cantidad de operaciones, lógicamente la tiempo que se demora el usuario depende no únicamente de la cantidad de operaciones.

- Existe una relación proporcional entre la curva de complejidad con la duración, sin embargo también entre mas habilidad tenga el usuario al utilizar el programa los tiempos se reducen, como ocurre en el ejercicio 6.
- No existe relación entre la curva de cantidad y complejidad.
- Al incrementar la complejidad de los ejercicios - 5, 6 y 7 -, se puede observar que no existe un incremento radical en la duración de estos ejercicios, lo cual demuestra que la aplicación continúa permitiendo realizar las operaciones con relativa facilidad.

Al examinar los resultados esta prueba se puede concluir que el programa permite al usuario final es decir un persona con ceguera o baja visión poder ingresar, editar y modificar expresiones algebraicas elementales para elaborar documentos que contengan fórmulas matemáticas de una manera sistemática y organizada, también toma en consideración las necesidades especiales de estos usuarios en su diseño e implementación y así mejorando su calidad de vida.

Capítulo 6

Conclusiones y Recomendaciones

6.1 Conclusiones

- Los pocos o inexistentes programas sociales especializados para personas con discapacidad visual, docentes capacitados, materiales de apoyo didáctico e información al público son obstáculos para formación académica de las personas no videntes sumándole también el hecho de que las personas no videntes que han vencido estos obstáculos deben “ingeniarse” para vencer las dificultades que presentan las matemáticas y su notación.
- El programa de edición permite a los usuarios no videntes ingresar, editar e imprimir usando una impresora braille o una impresora común expresiones algebraicas elementales usando un computador mediante un menú con una lista de operaciones y funciones matemáticas, una estructura de datos llamada árbol, un cuadro de edición de textos y mensajes de voz.
- El uso de árboles de datos y mensajes de voz permite al usuario no vidente vencer la dificultades de la unidimensionalidad, tamaño y carga cognitiva al representar las fórmulas matemáticas bidimensionales en braille, debido a que en esta aplicación los procesos de escribir y ubicar los elementos de las funciones y operaciones matemáticas se han separado para ayudar al usuario a centrar su atención en el contenido más que en su ubicación.
- El usuario no vidente al utilizar este programa puede realizar sus actividades escolares con o sin asistencia de un tutor, es decir el usuario gana independencia y evitando realizar un doble trabajo: primero al realizar sus actividades en braille y segundo al traducirlas.
- En comparación al utilizar el sistema braille, cuadernos con papel especial (95 gramos), punzones, regletas, ábacos y hasta de la máquina de escribir braille, el usuario no vidente tiene la ventaja al utilizar este programa de usar archivos digitales para poder intercambiar su información como si fuera un usuario común.
- La aplicación es ligera con un tamaño menor a 100 *Megabytes* pues no tiene como requisitos de instalación programas de otros desarrolladores y utiliza los

recursos propios del sistema operativo de *Windows 7* o superior, lo que permite que sea portable y no cause latencias producidas por los mensajes de voz en el computador.

- Según los resultados de las pruebas con ejercicios, el usuario se demora aproximadamente 1 minuto con 26 segundos aproximadamente en ingresar una operación o función matemática al computador, tomando en consideración que el usuario va ganando habilidad este tiempo puede ser menor, lo cual demuestra que el programa facilita el proceso de aprendizaje de una persona no vidente.
- Un docente que no tenga o tenga conocimiento limitado del sistema braille pueden utilizar el programa para ayudar a los alumnos no videntes a redactar documentos que contengan expresiones algebraicas elementales, facilitando la labor de la enseñanza.
- El programa permite el ingresar y editar fórmulas matemáticas a usuarios únicamente con discapacidad visual, si posee alguna discapacidad adicional no tendría sentido utilizar este programa.
- El programa fomenta los objetivos nacionales del buen vivir como: auspiciar la inclusión, mejorar la calidad de vida de la población y fortalecer las capacidades de la ciudadanía.
- Según la opinión del Dr. Benavides: “el programa permite una interacción docente / alumno más simple, porque permite al docente saber cómo está aprendiendo el alumno en ese mismo instante”.
- Según la opinión del Lic. Ganchala: “el programa beneficiaría mi labor de enseñanza puesto que para ingresar una fracción en el computador tenía que usar tanto *Excel* como *Word* para lograrlo”.

6.2 Recomendaciones

- Crear una versión gratuita del programa para familias de escasos recursos además de cursos de capacitación para el uso del programa.
- Usando la licencia de *Visual Studio Ultimate* que requiere de pago, hacer de este programa una extensión del Editor de Ecuaciones de *Microsoft Word*.
- Permitir importar expresiones algebraicas desde el Editor de Ecuaciones de *Microsoft Word* diseñando un algoritmo inverso al renderizado.

- El control *RichTextBox* realiza un mejor control de la ubicación del cursor que el control *TextBox* motivo por el cual se reemplazó a los controles *TextBox*.
- Usar un teclado extendido, es decir con bloque numérico y bloque de desplazamiento para un mejor uso de las personas no videntes porque tiene un mayor relieve para distinguir las teclas.
- Usar a plenitud las características de los sintetizadores de voz tanto al utilizar diferentes voces y variaciones tonales, es decir no escatimar al utilizar este recurso.
- Entrevistar a más personas con discapacidad visual en especial a quienes están cursando bachillerato.
- Revisar proyectos similares realizados en la Universidad de las Fuerzas Armadas - ESPE, a fin de formar un grupo de investigación para el desarrollo de herramientas para la accesibilidad para personas no videntes.
- Realizar más pruebas con casos y de uso para corregir posibles errores no controlados.

6.3 Trabajos futuros

- Integración de los proyectos de impresión y digitalización (*scanner*) de textos en braille con el programa editor de expresiones algebraicas.
- Búsqueda de nuevos avances en la verbalización de fórmulas matemáticas.
- Nuevas funciones para realizar cálculos matemáticos.
- Nuevos métodos para reducir los tiempos de ingreso de operaciones y funciones matemáticas.
- Seguridad del programa contra la piratería.
- Métodos con síntesis de voz como el audio 3D para ubicación del cursor en el programa a fin de mejorar la usabilidad del programa.

Bibliografía

- [1] A. E. Noboa and D. S. Noboa, “Diseño e implementación de un sistema electrónico con interface a PC para automatizar una máquina de escribir de braille.” 2014.
- [2] S. Fajardo, M. Andrade, A. Flores, and J. Lazaro, “MathML to ASCII-Braille and Hierarchical Tree Converter.” *Universidad de Colima*, 2008.
- [3] D. Archambault, “Non Visual Access to Mathematical Contents: State of the Art and Prospective,” *Université Pierre et Marie Curie-Paris 6*, 2010.
- [4] C. Bledsoe and A. Karshmer, “Access to Mathematics by Blind Students: Introduction to the Special Thematic Session,” *University of South Florida*, 2002.
- [5] H. Ferreira and F. Diamantino, “Leitura de Fórmulas Matemáticas para Cegos e Amblíopes: A Aplicação AudioMath.” *Universidade do Porto.*, 2006.
- [6] “El editor Lambda para matemáticas — Web de la ONCE.” [Online]. Available: <http://www.once.es/new/servicios-especializados-en-discapacidad-visual>
- [7] IndexBraille, “Basic-D V4: small, portable top-selling braille embosser/ braille printer,” 2015. [Online]. Available: <http://www.indexbraille.com/en-us/braille-embossers/basic-d-v4>
- [8] A. Constituyente, *Constitución de la República del Ecuador*. Registro Oficial No.449, Oct. 2008. [Online]. Available: http://www.asambleanacional.gov.ec/documentos/constitucion_de_bolsillo.pdf
- [9] CORDICOM, “En 2014 se presentaron 2018 personas con discapacidad a rendir pruebas para acceder a la educación superior.” *Consejo de Regulación y Desarrollo de la Información y Comunicación (CORDICOM)*, 2014. [Online]. Available: <http://www.cordicom.gob.ec>
- [10] CONADIS, “Registro Nacional de Discapacidades,” *Consejo Nacional de Discapacidades (CONADIS)*, Sep. 2014. [Online]. Available: <http://www.consejodiscapacidades.gob.ec>
- [11] V. d. I. R. d. Ecuador, “Niños y niñas no videntes acceden a la información y al conocimiento gracias a computadoras con lector de pantalla,” 2015. [Online]. Available: <http://www.vicepresidencia.gob.ec>

- [12] J. Fernández, “Braille y Matemática.” *Organización Nacional de Ciegos Españoles (ONCE)*, 2004. [Online]. Available: <http://www.once.es>
- [13] Larousse, “Discapacidad,” Barcelona, 2015. [Online]. Available: <https://www.google.com.ec/#q=Definicion+discapacidad>
- [14] OMS, “Ceguera y discapacidad visual,” 2014. [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs282/es/>
- [15] H. Deitel and P. Deitel, *C/C++ Cómo programar y Java*, 4th ed. USA: Pearson, Prentice Hall, 2004.
- [16] MSDN, “Árboles de expresión (C# y Visual Basic),” 2015. [Online]. Available: <http://www.msdn.microsoft.com>
- [17] Y. Takiguchi, M. Okada, and Y. Miyake, “A Fundamental Study of Output Translation from Layout Recognition and Semantic Understanding System for Mathematical Formulae,” *IEEE, Computer Society*, 2005.
- [18] W3C, “Mathematical Markup Language (MathML) Version 3.0 2nd Edition,” *World Wide Web Consortium*, 2014. [Online]. Available: <http://www.w3.org/TR/MathML3/>
- [19] MDN, “MathML Torture Test,” 2015. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/MathML_Project
- [20] MathJax, “MathJax Documentation,” 2015. [Online]. Available: <http://docs.mathjax.org/en/latest/start.html>
- [21] CodePlex, “MathView,” 2014. [Online]. Available: <http://mathview.codeplex.com/>
- [22] I. Alexandru, “FMath: Flash and javascript library for mathematics on web pages,” 2014. [Online]. Available: <http://fmath.info/>
- [23] D. Science, “MathPlayer. Accessible math for all!” 2015. [Online]. Available: <http://www.dessci.com/en/products/mathplayer/>
- [24] SourceForge, “GeckoFX is the open-source component for embedding the Gecko (Mozilla Firefox) rendering engine in .NET applications.” 2008. [Online]. Available: <http://sourceforge.net/projects/geckofx/>

- [25] F. Scientific, “JAWS Software,” 2015. [Online]. Available: <http://www.freedomscientific.com/>
- [26] Amazon, “Ivona Software,” 2015. [Online]. Available: <https://www.ivona.com/>
- [27] M. W. , “Microsoft Narrator.” [Online]. Available: <http://windows.microsoft.com/hear-text-read-aloud-narrator>
- [28] R. Fateman, “How can we speak math?” *University of California at Berkeley*, 2013. [Online]. Available: <http://www.cs.berkeley.edu/fateman/papers/speakmath.pdf>
- [29] A. Nemeth, “Mathspeak,” 1995. [Online]. Available: http://www.nfbcal.org/s_e/list/0033.html
- [30] “Guías de la Comisión Braille Española - Signografía básica,” 2005. [Online]. Available: <http://www.once.es>
- [31] “Guías de la Comisión Braille Española - Signografía matemática,” 2005. [Online]. Available: <http://www.once.es>
- [32] P. Ponsa and A. Granollers, “Diseño de Pantalla,” *Universitat Politècnica de Catalunya*, 2000. [Online]. Available: <http://www.epsevg.upc.edu/hcd/material/lecturas/pantalla.pdf>
- [33] M. Corporation, “Windows Vista and 7 User Experience Interaction Guidelines,” Sep. 2010. [Online]. Available: <http://www.microsoft.com/en-my/download/details.aspx?id=2695>
- [34] —, “Windows 8 desing and coding guidelines,” May 2014. [Online]. Available: go.microsoft.com/fwlink/p/?linkid=258743
- [35] B. B. Chaudhuri and U. Garain, “An Approach for Recognition and Interpretation of Mathematical Expressions in Printed Document,” *Indian Statistical Institute*, 2000.
- [36] M. Hersh and M. Johnson, *Assistive Technology for Visually Impaired and Blind People*. Springer, 2008.
- [37] W. Jakjoud, “Representation, handling and recognition of mathematical objects: state of the art,” *IEEE, Computer Society*, 2009.

- [38] P. GNOME, “Guías de la interfaz humana de GNOME,” 2014. [Online]. Available: <https://developer.gnome.org/hig/stable/>
- [39] W. Wongkia, K. Naruedomkul, and N. Cercone, “Better Access to Math for Visually Impaired,” *IEEE, Computer Society*, 2009.
- [40] R. Zanibbi and D. Blostein, “Recognition and Retrieval of Mathematical Expressions,” *National Science Foundation and Engineering Research Council of Canada and the Xerox Foundation*, n.d.