ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

APLICACIONES DE SOFTWARE EDUCATIVO Y EMPRESARIAL

DESARROLLO DE UN MÓDULO DE SONIFICACIÓN DE ECUACIONES ALGEBRAICO-MATEMÁTICAS PARA SISTEMA CASVI 2.0

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

MATEO NICOLÁS SALVADOR VALLEJO

mateo.salvador@epn.edu.ec

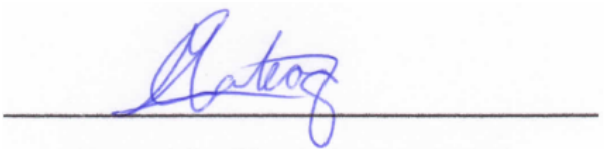
DIRECTOR: PhD. ANA MARÍA ZAMBRANO VIZUETE

ana.zambrano@epn.edu.ec

DMQ, Abril 2023

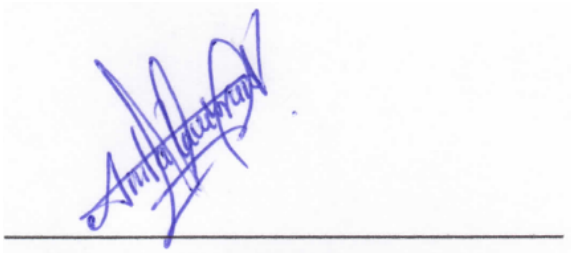
CERTIFICACIONES

Yo, Mateo Nicolás Salvador Vallejo declaro que el Trabajo de Integración Curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Mateo Nicolás Salvador Vallejo

Certifico que el presente trabajo de integración curricular fue desarrollado por Mateo Nicolás Salvador Vallejo, bajo mi supervisión.

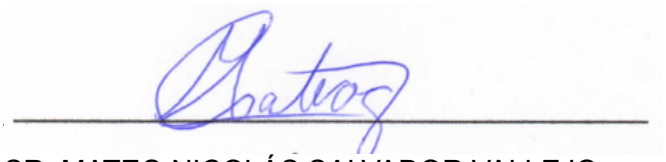


Dra. Ana María Zambrano Vizuete

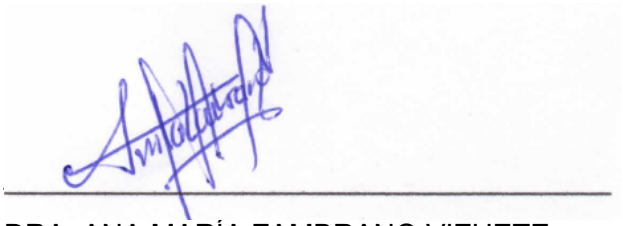
DIRECTOR

DECLARATORIA DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



SR. MATEO NICOLÁS SALVADOR VALLEJO



DRA. ANA MARÍA ZAMBRANO VIZUETE

DEDICATORIA

A mis padres y hermana por haberme apoyado incondicionalmente en cada etapa de vida, en especial durante los momentos en los que más motivación necesitaba, por lo cual no hubiese conseguido todos mis objetivos sin su ayuda.

A mi familia en general por haberme animado y acompañado durante toda esta etapa y ofrecerme su compañía incondicional en cada momento.

A mis amigos por no permitir que me rinda y siempre estar presentes en mi desarrollo como persona y como profesional.

AGRADECIMIENTO

Un total agradecimiento a la Dra. Ana María Zambrano, directora de este Trabajo de Integración Curricular, Directora de la Carrera de Tecnología de la Información y docente de la Escuela Politécnica Nacional, por haber confiado la oportunidad de desarrollar este trabajo y guiarme en cada paso de éste.

Al Dr. Felipe Grijalva por haberme brindado pautas que permitieron completar este trabajo. Junto a ellos agradezco a los docentes que estuvieron presentes en mi formación dentro de la Escuela Politécnica Nacional.

ÍNDICE DE CONTENIDOS

RESUMEN

La sonificación es una técnica que permite la representación de datos (figuras, texto o elementos multimedia) mediante la generación de audios y sonidos que funcionen como la herramienta descriptiva de estos. Esta técnica puede ser utilizada como una herramienta mediante la cual expresiones matemáticas pueden ser representadas e interpretadas.

En el presente Trabajo de Interacción Curricular se plantea el desarrollo de un módulo de sonificación para el sistema CASVI 2.0, dirigido a usuarios invidentes, el cual permita la descripción de expresiones matemáticas. Para el proceso de sonificación se utilizan las herramientas de *Text to Speech* para la lectura descriptiva de la expresión y la inserción de tonos como pistas auditivas; donde todo el procesamiento es realizado en el lado de Servidor, utilizando el lenguaje de programación de *Python*.

En el primer capítulo se investiga el Estado del Arte en la sonificación de ecuaciones, así como el Estado del Arte del desarrollo software para usuarios no videntes. Además de esto se procede a explorar las distintas herramientas necesarias para la implementación del módulo de sonificación. Por último se define una metodología que permita la organización y desarrollo del Trabajo de Integración Curricular.

En lo que respecta al segundo capítulo, se definen las diferentes consideraciones de diseño del módulo, que permiten que éste se ajuste a las patrones coincidan con las pautas de accesibilidad dirigida a usuarios no videntes. Además de que se definen los distintos requerimientos funcionales y no funcionales, permitiendo que el módulo sea de agrado del usuario. Finalmente en este capítulo se detalla la implementación realizada para el correcto funcionamiento del módulo de sonificación.

Por último, se realizan las pruebas del módulo dentro de un escenario controlado mediante el cual con la ayuda individuos que simulan ser usuarios no videntes, se verifica el correcto funcionamiento del módulo ya integrado en el sistema y se realizan las correcciones necesarias para que esté cumpla con todos los requerimientos establecidos.

PALABRAS CLAVE: Sonificación, Text to Speech, pistas auditivas, usuario no vidente, expresión matemática

ABSTRACT

Sonification is a technique that allows the representation of data (images, text or media) through the generation of sounds and audios that works as a descriptive tool. This technique can be used as a resource with which mathematical expressions can be represented and interpreted.

The actual Curricular Integration Paper arises from the development of a sonification module for the CASVI 2.0 system, which is aimed at visually impaired users, and allows the description of mathematical expressions. For the sonication process, the use of Text to Speech tools to get a descriptive reading of the expression and the insertion of tones as audio keys, are implemented; where all the processing is executed on the Server side, using Python as programming language.

In the first chapter, the State of Art of equation sonification and the State of Art of software development for visually impaired people, are investigated. Also in this chapter, different resources and tools for the implementation of the sonification module, are investigated. Finally a methodology that allows the scheduling and development of the Curricular Integration Paper, is defined.

In the second chapter, the design considerations for the module are defined, allowing the module to adjust with the patterns of accessibility for visually impaired users. Also the functional and not functional requirements are established, with the aim of being pleasant for the user.

Finally, with the help of a group of individuals, a series of tests are executed within a controlled scenario where the individuals simulate to be blind users, helping to determine the correct functioning of the module integrated in the system, and if is necessary, apply the corresponding correction to make the module meet all the established requirements.

KEY WORDS: Sonification, Text to Speech, audio keys, visually impaired user, mathematical expression.

1. INTRODUCCIÓN

Considerar la accesibilidad como una característica más, dentro del desarrollo del software, es un aspecto importante para conseguir una sencilla interacción por parte de un usuario no vidente con la aplicación. Si bien existen muchos recursos, como lectores de pantalla o traductores a braille, que pueden servir como herramientas de soporte, muchas veces no presentan una óptima compatibilidad con los programas y es necesario que éstos tengan sus propios componentes de apoyo, como lo son sintetizadores de voz o sonidos característicos que funcionen como guía.

Dentro de la implementación de un sistema matemático, que busque ser accesible para un usuario no vidente, como lo es CASVI 2.0, es necesario que éste cuente con los elementos necesarios para poder representar las diferentes expresiones obtenidas de la resolución de distintas operaciones matemáticas.

A lo largo del presente Trabajo de Integración Curricular se aborda el desarrollo e integración de un módulo de sonificación de ecuaciones sobre el sistema CASVI 2.0, con el cual sea posible representar la salida de una ecuación procesada, mediante la combinación de una voz sintetizada y una serie de pistas auditivas que permitan al usuario identificar y guiarse dentro de la estructura de la expresión matemática.

1.1 OBJETIVOS

1.1.1 Objetivo General

El objetivo general de este Proyecto de Integración Curricular es implementar un módulo de sonorización de ecuaciones sobre el existente prototipo web matemático CASVI 2.0, con el fin de que el usuario con discapacidad visual tenga la oportunidad de interpretar ecuaciones a través de la generación de sonido espacial. Junto a esto, el presente Proyecto de Integración Curricular presenta los siguientes objetivos específicos:

- Analizar la metodología Kanban y las herramientas necesarias para el desarrollo del módulo.
- Diseñar el flujo de trabajo para otorgar sonificación de ecuaciones matemáticas dentro del sistema.
- Implementar el módulo en base a lo establecido en su diseño.
- Incorporar y comprobar el correcto funcionamiento en base a pruebas realizadas en un escenario simulado.

1.2 ALCANCE

Dentro del presente Proyecto de Integración Curricular se propone el desarrollo de un complemento de software basado en la sonificación de ecuaciones para el mejoramiento sistema CASVI 2.0, el cual por el momento presenta una lectura lineal y no muy amigable/agradable para el usuario, de manera que la presentación auditiva de estas sea de agrado y correcto entendimiento para los usuarios del sistema, que en este caso corresponde a personas no videntes.

La lógica del módulo permitirá procesar la salida de la ecuación en el sistema, de manera que sea capaz de generar una secuencia de audio con la cual el usuario no vidente pueda reconocer, de la mejor manera, la expresión generada.

Para esto se requerirá la aplicación de diferentes características del sonido (frecuencia, timbre, canales, entre otros), con las que sea posible establecer un audio espacial que concuerde con los datos procesados.

En primera instancia se procederá a estudiar el Estado del Arte de software para personas no videntes, así como el Estado del Arte de sonificación de ecuaciones. Se analizarán los diferentes parámetros del sonido como: frecuencia, ritmos armónicos, tiempo, entre otros; y a la par se estudiarán las características de audio.

Se procederá a estudiar la estructura y funcionamiento del Sistema CASVI 2.0 junto con sus respectivos lenguajes y tecnologías [1][2][3] que lo componen.

Por último, se examinará la metodología Kanban [4] para aplicarla en Trabajo de Integración Curricular, con el propósito de obtener un desarrollo ágil, a través de la correcta planificación de tareas en periodos de tiempo determinados.

En la segunda fase se establecerán una serie de actividades con sus respectivos tiempos para desarrollo, logrando con esto una mejor organización en todas las fases del proyecto, mediante el uso del tablero de Kanban. Junto a esto se utilizarán diagramas UML que permitan reflejar el funcionamiento deseado del módulo.

En la tercer fase, partiendo de lo establecido en la fase de diseño, junto con el estudio del funcionamiento de CASVI 2.0 y tomando como guía el método de Kanban, se procederá instalar el IDE para desarrollo, los paquetes de lenguaje de programación y los scripts que componen al sistema CASVI 2.0. Consecuentemente se continuará con el desarrollo del módulo mediante el cual se procesan las ecuaciones ingresadas dentro del sistema y generar la secuencia de audio que las representa.

Culminada la implementación y verificado el cumplimiento del tablero de Kanban, se continuará con la verificación de los requerimientos y se realizarán pruebas dentro de un ambiente controlado, en el cual se simulará la falta de visión, para así comprobar la correcta interpretación de las ecuaciones generadas dentro del sistema. En caso de ser necesario se realizarán las debidas correcciones del módulo, con el fin de cumplir los objetivos establecidos dentro del Trabajo de Integración Curricular.

1.3 MARCO TEÓRICO

En esta sección se presenta el estudio del Estado del Arte de software para personas invidentes, así como el Estado del Arte de la sonificación de ecuaciones, que establecen una base para el desarrollo de un **Módulo de Sonificación,** dentro de un sistema matemático orientado a usuarios invidentes; así como las recursos del Sistema CASVI 2.0. Este es aplicado como base referencial para el desarrollo del Proyecto de Integración Curricular. Además se define la metodología de Kanban que se utilizará como apoyo para la planificación y avance de las diferentes fases del proyecto.

1.3.1 ESTADO DEL ARTE DE SOFTWARE PARA PERSONAS INVIDENTES

Las personas con discapacidad visual han presentado dificultades con la navegación e interacción dentro de las tecnologías y programas informáticos [10][5], es así que con el surgimiento de nuevos avances en este campo, se ha buscado que éstas presenten un modelo en el que se consideren una serie de criterios y estándares [11] de manera que aseguren la accesibilidad por parte de cualquier usuario.

Usuarios con deficiencias visuales suelen presentar problemas para entender y guiarse en la estructura de una interfaz gráfica, en especial si éstas tienen integrado elementos como tablas, formularios o elementos con cambios de color en la información. Así también presentan problemas para utilizar el teclado y operar con las diferentes entradas y contenidos que son destinadas a interactuar mediante el uso del ratón [9].

El desarrollo de software orientado a usuarios no videntes, es un campo en el cual se debe considerar un enfoque bajo los principios de diseño universal [6], de manera que éste pueda presentar una óptima usabilidad y adaptabilidad por parte del usuario. Como lo muestra la W3C (*World Wide Web Consortium*) dentro la *“accesibilidad de un software se debe ser más flexible, manejable, extensible y beneficioso para una amplia gama de usuarios”* [15]. y sus tecnologías de asistencia como se muestra en la Figura 1.1 [15].

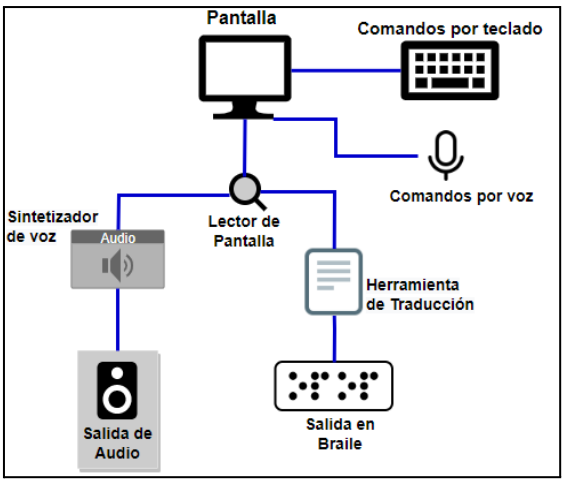


Figura 1.1 Adaptación de software a tecnologías de asistencia.

Es así que, como lo muestra la Tabla 1.1, este tipo de programas se deben manejar bajo los conceptos de perceptibilidad, operabilidad, comprensibilidad y robustez; consiguiendo con ello una fácil navegación del lado del usuario.

Tabla 1.1 Conceptos de Accesibilidad en Aplicaciones Web según W3C [13].

Concepto	Descripción
Perceptibilidad	La información y componentes de la interfaz de gráfica deben ser presentadas a los usuarios sin importar la manera en que puedan percibirlos
Operabilidad	Los componentes y navegación de la interfaz deben ser fáciles de operar.
Comprensibilidad	La información y operación de la aplicación deben ser fáciles de entender
Robustez	El contenido del programa tiene que ser lo suficientemente robusto para que pueda ser interpretado por una gran variedad de usuarios y sus herramientas de asistencia

En la etapa de Diseño, el software debe partir de la idea de que la presentación del contenido dentro del programa debe aplicar una orientación coherente y no muy cargada. Es por esto que, se debe utilizar una perspectiva simple, intuitiva y de uso flexible, para que el usuario pueda interactuar sin dificultad, a través de todas las funcionalidades con las cuales la aplicación puede contar.

Si bien los usuarios no videntes presentan herramientas de apoyo, como los lectores de pantalla o la representación de la interfaz mediante el lenguaje braille, muchas veces estos

medios, no son compatibles con navegadores (incompatibilidad con aplicaciones web) o con el contenido que presenta la interfaz gráfica del usuario desarrollada para la aplicación [6].

Además, cabe mencionar que si bien como se muestra en la Figura 1.2 existe una variedad de lectores de pantalla (tales como *NVDA* [1]*, JAWS* [2]*, Talkback* [3], entre otros) y son un recurso muy útil para guiar al usuario, se pueden presentar dificultades con la representación textual de diagramas y elementos visuales, como son dibujos, ecuaciones, enlaces, pósters, imágenes, entre otros. Es por esto que surge la necesidad de implementar dentro del programa módulos que permitan representar tanto el contenido gráfico, como textual dentro de la interfaz, así como guías que faciliten la navegación dentro de ésta.

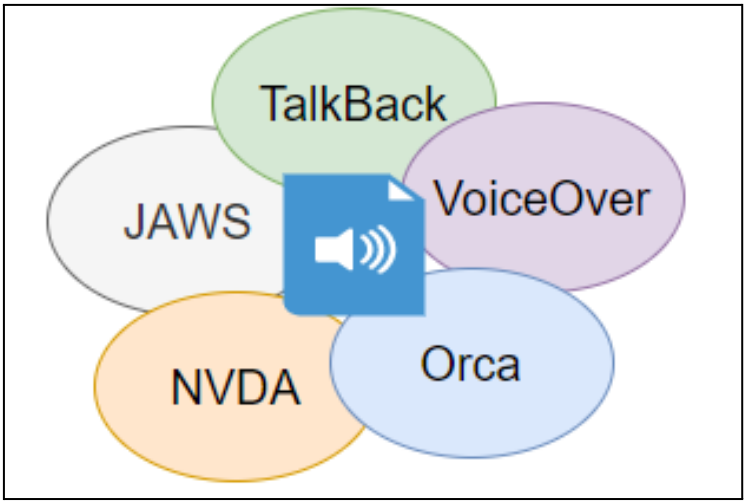


Figura 1.2 Lectores de pantalla.

La implementación de sintetizadores de voz que trabajan como traductores de texto a voz (*Text to Speech* - TTS), como lo presenta la Figura 1.3, permiten establecer una ayuda dentro del software, no sólo para describir los diferentes componentes de la interfaz, sino para servir como una guía mediante la cual el usuario pueda ser guiado dentro de las diferentes operaciones del software. Es así que es necesario que mientras el software sea utilizado, el sistema proporcione una constante retroalimentación de los distintos componentes de la interfaz, así como las diferentes interacciones realizadas (i.e teclas pulsadas); pero considerando evitar una repetición excesiva del contenido, y la opción de saltarla si el usuario lo desea [7].

[1] NVDA (NonVisual Desktop Access) es un software lector de pantalla de acceso libre.
[2] JAWS (Job Access with Speech) es un software lector de pantalla licenciado.
[3] TalkBack es un servicio de accesibilidad de google que permite la interacción con dispositivos.

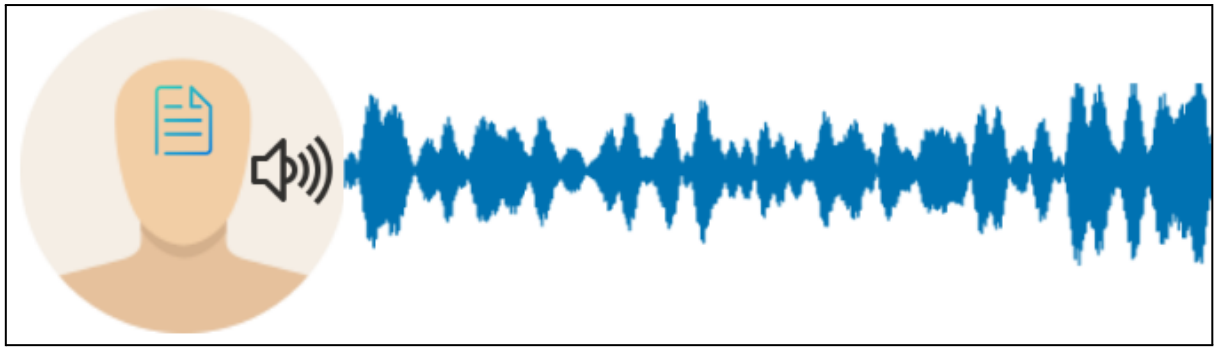


Figura 1.3 *Text to Speech* como herramienta de apoyo.

En los parámetros de accesibilidad se debe incluir una sencilla interacción con el sistema. Para esto es necesario que se disponga de atajos o *"shorcuts"*, a través de combinaciones de teclas, de manera que funcionen como comandos que sustituyan las funcionalidades de un ratón [12] y faciliten la navegación y operación dentro de la interfaz.

Al ser un software dirigido a usuarios no videntes hay que considerar que sus campos deben ser de rápido y fácil acceso, así como presentar un orden lógico y sin sobrecarga de la pantalla. Además, se debe tener en cuenta que la navegación dentro de la aplicación debe ser circular[4] y con un rápido acceso [14].

Tomando como referencia los criterios que un sitio web debe tener para ser considerado como accesible para usuarios con discapacidad visual, el contenido de una interfaz gráfica debe contar con [9]:

- Aplicación de texto alternativo que determine una descripción breve de elementos visuales como animaciones.
- Correcta estructuración de las pantallas de manera que se facilite la síntesis de voz y el desplazamiento secuencial entre ellas.
- El contenido de las tablas debe ser realizado de manera que puedan ser leídas línea por línea.
- Utilizar controladores estándar que sean compatibles con el sistema sin necesidad de instalar plugins adicionales como Flash Player o Java Virtual Machine.

Además, posterior a una búsqueda extensa, como se muestra en la Tabla 1.2, se ha conseguido encontrar algunos proyectos dirigidos a usuarios no videntes.

[4] Una navegación circular significa que la interacción con toda la interfaz pueda realizarse en diferentes direcciones sin afectar la lógica de la aplicación.

Tabla 1.2 Proyectos de software dirigido a usuarios no videntes.

Proyecto	Descripción
Aprendizaje Móvil de Ciencias para Ciegos (AudioNature) [16]	Desarrollo de una interfaz de audio para aplicación móvil, con el propósito de promover el estudio de ciencias naturales a estudiantes invidentes. A través del motor TTS todos los componentes de la aplicación son representados a través de voces sintetizadas y sonidos reales con el objetivo de describir diferentes elementos de un ecosistema. Toda la interacción con la aplicación se realiza mediante el tacto y recibe una retroalimentación auditiva de todos las acciones realizadas.
Sistema Informático para Niños Discapacitados Visuales en Etapa Preescolar [17]	Implementación de un Sistema Informático Especializado para el desarrollo integral de niños con discapacidad visual en etapa preescolar, por medio del uso de recursos didácticos como juegos e historias. El sistema aplica estimulación multisensorial a través de sonidos y motores de síntesis de voz interpretados por agentes virtuales para la representación de todos los elementos que conforman la interfaz.
Asistente auditivo lector de texto como apoyo a personas con discapacidad visual [19]	Desarrollo de una aplicación para *Smartphones* que realiza la conversión de texto embebido dentro de una imagen, a una salida de audio descriptivo, con la ayuda de APIs desarrolladas en Android.
Aprendizaje de Ciencias a través de Audio en Niños Ciegos (AudioLink) [18]	Desarrollo de un software interactivo dirigido a niños no videntes, el cual consiste de un juego de rol que permite la navegación dentro de un mundo virtual. La descripción del entorno del juego se realiza mediante un archivo XML, que es interpretado por un sistema de audio capaz de generar pistas auditivas, diálogos y alertas que permitan la comprensión de los ítems que componen el espacio virtual. Para la interacción con la interfaz, el usuario utiliza una serie de teclas predefinidas con las cuales el avatar del juego podrá hacer diferentes acciones.

1.3.2 ESTADO DEL ARTE DE SONIFICACIÓN DE ECUACIONES

La lectura de una ecuación es un proceso simple para personas videntes, debido a que la representación escrita de ésta, permite el dimensionamiento de la estructura que la compone, así como posibilitar un análisis de adelante hacia atrás y viceversa, de todos los elementos que la componen [8]. En el caso de personas no videntes, la complejidad se presenta en la necesidad de exhibir la estructura de la ecuación de una manera sencilla, que les permita no sólo memorizar la composición de las expresiones, sino también su comprensión.

Representar e interpretar expresiones matemáticas no es igual que hacerlo con texto, pues las matemáticas pueden llegarse a considerar como tal un lenguaje más, por lo que presentarlas de una manera accesible es una tarea compleja. La aplicación de métodos auditivos permite el uso de sintetizadores y sonidos como una forma de leer y representar las ecuaciones [20].

Para la sonificación de las ecuaciones es indispensable representarlas a través de alguna estructura de datos o lenguaje de marcado (tanto en presentación como en contenido), tales como *MathML* [37] o *LaTeX* [36], que se encargue de la traducción de la sintaxis de la fórmula y entregue una expresión cercana a como se leería [22], facilitando la generación de lenguaje natural que pueda ser leído por un sintetizador de voz.

Si bien no existe un conjunto de reglas estandarizadas, es necesario establecer una serie de pautas y preferencias que generen la correcta salida de “voz”, mediante las cuales se pueda ser más específico y consistente en la descripción de la ecuación (evitando así ambigüedades en su interpretación). Por ejemplo si se toma la fracción “$\frac{7}{12}$”, esta puede ser leída como “siete doceavos”, pero manteniendo una expresión general esta puede ser leída como “siete sobre doce” [21].

Con una estructura fácil de navegar y leer, es posible pre procesar y traducir la expresión a un formato de texto que pueda ser transmitido a TTS y genere un audio renderizado a través de una voz sintetizada. [20]

Si bien el uso de la tecnología TTS cuenta con la ventaja de que la renderización de audio no presenta un alto nivel de sofisticación [8], puede llegar a presentar limitaciones mientras más compleja se vuelva la ecuación. Si se toman algunas fórmulas de la Figura 1.4 como ejemplo, éstas pueden presentar expresiones que tienen un inicio y un final, y no pueden ser diferenciadas a través de una “lectura plana”.

Fórmulas			
$\frac{x+1}{x-1}$	$\sum_{x=b}^{y} b$	$\pi \approx 22/7$	$3 \leq x \leq 5$
$x^5 + y^4 + z$	$a \in A$	$\forall x : (x - x = 0)$	$\cos(\sin(x) + x^2)$
$\sqrt[m]{a}$	$\int \cos = -sin$	$(fog)(x) = f(g(x))$	$\log_3 x + \ln a$
$\frac{\lambda^2 f(x,y)}{\mu}$	$\int_0^1 x^2 + 2x\,dx$	$b \equiv \neg\neg b$	$A \otimes B$
$x^2 \to a^2$	$A \times B$	$6 \geq y \geq 2$	$\lim_{x \to \pi} \cos(x)$

Figura 1.4 Estructuras de las fórmulas más comunes [22].

1.3.2.1 Recursos Adicionales en la Sonificación de Ecuaciones

Además de la herramienta de TTS, es necesario utilizar recursos adicionales en el proceso de sonorización, como se presenta en la Tabla 1.3, que contribuyan a una correcta percepción y entendimiento por parte del usuario.

Tabla 1.3 Recursos complementarios para la sonificación de ecuaciones [8].

Recurso	Descripción
Pistas Léxicas	El uso de este recurso hace referencia a la aplicación de señales textuales que funcionan como delimitadores dentro de la ecuación (i.e “inicio de la raíz cuadra” y “fin de la raíz cuadrada”), es decir, que indiquen dónde empieza y dónde termina la expresión, evitando así casos de ambigüedad en estructuras anidadas.
Pistas Prosódicas	La utilización de la prosodia consiste en la integración de diferentes componentes auditivos como pausas, decaimientos de tono y variaciones de amplitud, para generar entonaciones y acentos característicos al momento de generar salidas de voz.
Pistas Auditivas	Consiste en el empleo de una sucesión de tonos arreglados de una forma característica, con el propósito de representar los diferentes componentes de la ecuación, tales como: paréntesis, raíces, potencias, expresiones geométricas, fracciones, entre otros.

1.3.2.2 Parámetros y Efectos del Sonido

Al momento de representar ecuaciones y expresiones matemáticas a través del sonido se debe a tener a consideración lo diferentes parámetros y efectos del sonido con los cuales se puede trabajar y son descritos en la Tabla 1.4

Tabla 1.4 Parámetros y Efectos de Sonido.

Parámetro	Descripción
Frecuencia [24]	Número de oscilaciones que una puede presentar dentro de una unidad de tiempo; determinando si el sonido será más grave o más agudo. A una menor frecuencia, el tono del sonido será más grave, mientras que a una mayor frecuencia se tendrán tonos más agudos.
Amplitud [24]	Altura que presenta la onda del sonido y definirá la intensidad de este. Amplitudes bajas definirán un sonido más leve y amplitudes grandes un sonido más intenso.
Timbre [24]	Forma en la que se presenta la onda del sonido y permite diferenciar entre sonidos con una misma frecuencia y amplitud pero de una diferente fuente.
Vibrato [25]	Variación de la ondulación del sonido, ya sea en frecuencia o amplitud, provocada por cambios en la vibración del tono.
Decaída (Fade out) [24]	Reducción progresiva del volumen de un tono hasta llegar a cero.

Después de una búsqueda exhaustiva, como se presenta en la Tabla 1.5, existen varios proyectos que se pueden presentar como referencia para la sonificación de ecuaciones.

Tabla 1.5 Proyectos para la sonificación de ecuaciones.

Proyectos	Descripción
Prototipo para la Accesibilidad de Expresiones Matemáticas [22]	Desarrollo de un módulo de software para aplicaciones web que extrae expresiones matemáticas codificadas en *Content-MathML* y las convierte a una estructura de árbol con la cual es posible la representación de ecuaciones en lenguaje natural, permitiendo la lectura de éstas por parte de un lector de pantalla.
***Web Sonification Sandbox* [52]**	Desarrollo de un entorno de pruebas para el mapeo de información y ecuaciones, que permite su respectiva estructuración y sonificación a través de la librería *Web Audio* API[5], que generará sonidos de una fuente definida, funcionando como pistas auditivas.
***Math Genie* [51]**	Software encargado del escaneo de ecuaciones matemáticas ingresadas, que luego de su lectura serán estructuradas en un esquema determinado que permitirá la reproducción de una voz sintetizada que realice el dictado de la expresión elemento por elemento.
***MathSpeak* [20]**	Software para la sonificación de expresiones matemáticas representadas en formato AMS-LaTex y traducidas a lenguaje natural a través de un motor TTS que generará una salida de audio mediante una voz sintetizada.

1.3.3 HERRAMIENTAS DEL MÓDULO DE SONIFICACIÓN

Para el desarrollo del **Módulo de Sonificación** es necesario contar con una serie de herramientas de código que permitan el procesamiento y generación de salidas de audio que faciliten representar una expresión matemática, por lo cual como se presenta en la Tabla 1.6 se analizan las diferentes librerías de código cuyas funciones nos permiten la implementación del módulo dentro del sistema.

[5] Interfaz de programación que permite la reproducción de audio dentro de una aplicación *Web*

Tabla 1.6 Librerías de Código para el Módulo de Sonificación

Recurso	Descripción
Mathlive [44]	Librería de código libre de *JavaScript* que permite el procesamiento de expresiones matemáticas en formato LaTeX, incluyendo la funcionalidad de transformar esta estructura de datos a una descripción detallada en lenguaje natural.
Googletrans [45]	Librería de código libre de *Python* que permite implementar la *API* de *Google Translator* para la traducción de cadenas de texto de un idioma a otro.
Regular Expression (re) [46]	Librería de *Python* que permite el manejo de expresiones regulares[6] (*regex)* y la búsqueda de patrones coincidentes, en cadenas de texto, mediante éstas.
Operating system (os) [47]	Módulo de *Python* mediante el cual es posible aplicar funcionalidades dependientes de un sistema operativo dentro de la lógica del código.
Time [48]	Módulo de *Python* con el cual es posible aplicar funciones relacionadas con el tiempo, como lo puede ser la aplicación de pausas o descansos en la ejecución de un código.
Pyttsx3 [43]	Librería de *Python* de código libre mediante la cual es posible la conversión de texto a una voz sintetizada (TTS).
Pydub [49]	Librería de código libre de *Python* que permite el manejo y manipulación de archivos de audio.
Tones [50]	Librería de código abierto de *Python* que permite la generación de tonos y la manipulación de sus diferentes parámetros.

[6] Una expresión regular es una secuencia de caracteres que permite establecer un patrón de búsqueda dentro de una *string.*

1.3.4 METODOLOGÍA DE KANBAN

La metodología de Kanban es un concepto japonés que fue introducido en la década de 1950, en la industria manufacturera. La palabra Kanban se traduce como cartel o tablero; y es usada para definir un sistema de producción bajo calendario o planeación. Sus principios parten de un control del flujo de trabajo, de manera que todas las actividades y tareas involucradas en el desarrollo de un proyecto sean ejecutadas y entregadas justo a tiempo (sistema de trabajo *just in time*) [26].

Kanban al ser un sistema de trabajo, *just in time,* busca que dentro de la gestión de un proyecto se evite los excesos innecesarios de tiempo y esfuerzo, evitando así la sobrecarga de trabajo. La gestión mediante Kanban permite que se genere una disminución en el tiempo de ejecución de tareas y a la vez un mayor enfoque sobre ellas, sin necesidad de tener que asumirlas todas a la vez [4].

Como lo muestra la Figura 1.5 la metodología de Kanban toma un proyecto y lo descompone en N tareas, las cuales deben entregar un resultado y no interrumpir la secuencia del resto de actividades, evitando así retrasos dentro del desarrollo del proyecto. Un trabajador involucrado dentro de un proyecto controlado por esta metodología, no puede iniciar con una actividad a menos de que Kanban indique que esta tarea es requerida para la continuación de otra actividad [27].

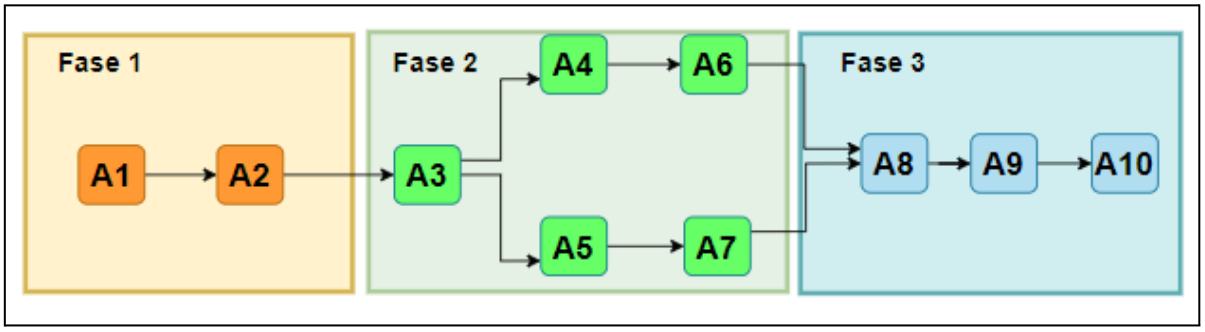


Figura 1.5 Descomposición de un proyecto en tareas [27].

Dentro del desarrollo de un proyecto se pueden presentar problemas como largos tiempos de ejecución, falta de calidad en el resultado, desperdicios, velocidad de trabajo irregular, organizaciones confusas y entregas con fallas [30]; con lo cual para la corrección de estos problemas, dentro de la implementación de la metodología de Kanban se tiene una serie de principios entre los que se puede nombrar [28]:

- Limitación del trabajo en proceso.
- Transparencia en el proceso de desarrollo.

- Determinación del valor del proceso.
- Aumento en el rendimiento.
- Inclusión de calidad.
- Uso de una priorización fija en las tareas.

La aplicación de estos principios permite enfocarse en la correcta realización del trabajo en el tiempo deseado y utilizando todas las habilidades y recursos disponibles, eliminando cualquier desperdicio de esfuerzo durante los avances del proyecto.

El tablero de Kanban es una herramienta que permite visualizar y coordinar todas las actividades necesarias dentro de la ejecución de un proyecto y su secuencia de realización. Estas actividades son representadas mediante tarjetas, en las cuales se especifican las diferentes características para limitar el trabajo en proceso, y son ubicadas en una columna que indica su nivel de progreso (i.e “por hacer”, “en proceso” y “realizado”) [29].

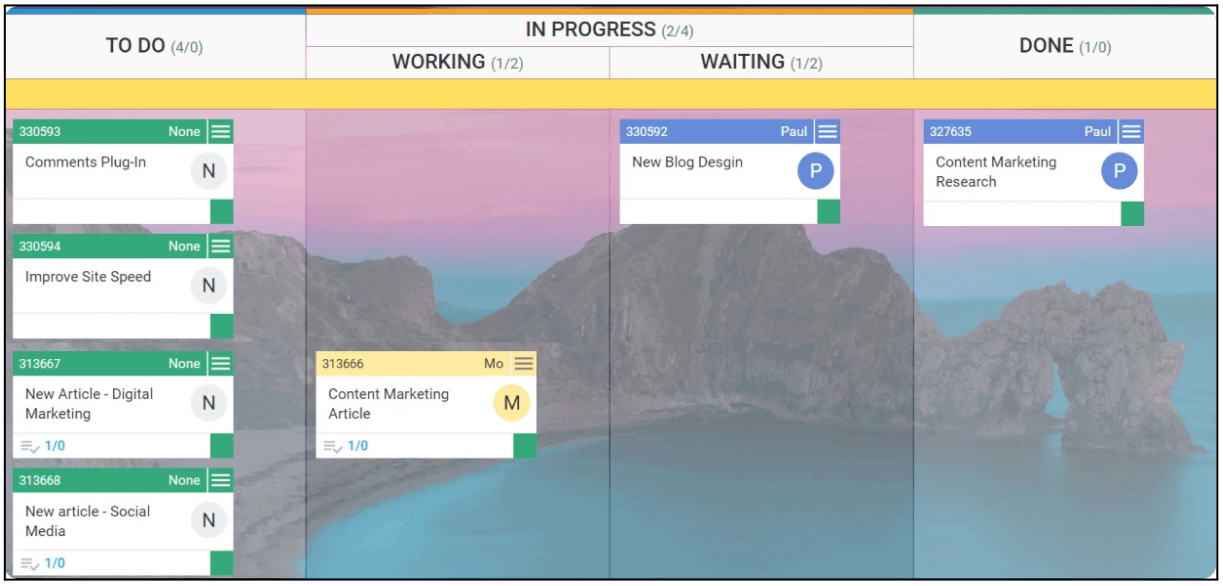


Figura 1.6 Tablero de Kanban

Como se muestra en la Figura 1.6 un tablero de Kanban gestiona las tareas dentro de un proyecto, a través de una estructura que las clasifica por un nivel prioridad y una etapa dentro del flujo de trabajo; logrando así una mejor administración de recursos y manejo del tiempo.

2. METODOLOGÍA

En este capítulo se presentarán los detalles del módulo de sonificación a llevar a cabo con su debida integración sobre el sistema CASVI 2.0, las diferentes consideraciones de diseño establecidas y los distintos pormenores implementados en el proceso de desarrollo.

En el transcurso del **Apartado 2.1 DISEÑO**, se definirán los criterios primordiales para el desarrollo del módulo, mientras que en el **Apartado 2.2 IMPLEMENTACIÓN**, se examinarán las diferentes decisiones de desarrollo ejecutadas.

2.1 DISEÑO

2.1.1 ESTABLECIMIENTO DEL TABLERO DE KANBAN

Para el planteamiento del tablero de Kanban se procede a utilizar la herramienta de *Microsoft Planner* [42], en la cual se definen las diferentes actividades a efectuar, así como se actualizan los resultados en la medida que los objetivos establecidos sean completados.

Como se observa en la Figura 2.1, dentro de la fase de Diseño, se deben realizar cinco actividades, cuya apropiada culminación permitirá obtener todos los elementos necesarios para una satisfactoria implementación e integración del Módulo de Sonificación dentro del sistema CASVI 2.0.

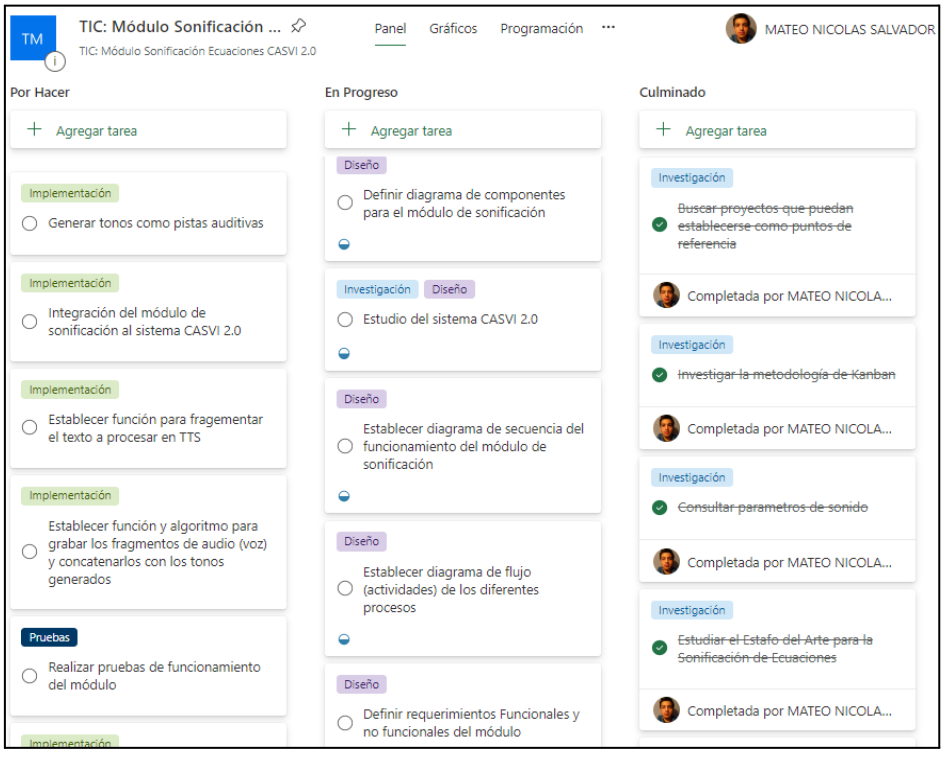


Figura 2.1 Establecimiento de Tablero de Kanban en Fase de Diseño.

2.1.2 CASVI 2.0

CASVI 2.0 se determina como un *"Sistema Prototipo De Cálculo Numérico-Algebraico Interactivo Para Estudiantes Con Discapacidad Visual"* que funciona como una aplicación web, bajo un modelo Cliente-Servidor. Este sistema consigue que tanto usuarios videntes como invidentes puedan interactuar con una interfaz que ofrece un espacio que permite trabajar con diferentes expresiones matemáticas; desde la ejecución de operaciones básicas (suma, resta, multiplicación y división) hasta la resolución de ecuaciones o la expansión de polinomios. Como se muestra en la Tabla 2.1, CASVI 2.0 se compone de 4 proyectos de código, que se encargarán de todo el manejo de la aplicación.

Tabla 2.1 Proyectos que componen CASVI 2.0.

Proyecto	Descripción
Front End Master	Proyecto implementado bajo el framework de *Vue.js* y la infraestructura de *Node.js*, encargado de toda la interacción con el usuario. Es responsable de la presentación de la aplicación y contará con todos los componentes necesarios para guiar al usuario no vidente en el entorno de trabajo.
Back End Master	Proyecto implementado bajo la Infraestructura de *Node.js* encargado de la gestión de usuarios (acceso*)* y la comunicación de la aplicación con la base de datos y el servidor.
API Master	Proyecto implementado en *Python*, que funciona como una *API REST*, responsable del procesamiento de las expresiones algebraicas y la comunicación con el motor de *MAXIMA* para su resolución.
SnuggleTex	*API* utilizada para la transformación de fragmentos en formato *LaTeX* a *MathML.*

Como muestra la Figura 2.2, para la representación de las ecuaciones y sus salidas, una vez que la ecuación es ingresada para su resolución, el *Front-End*, mediante una solicitud *HTTP*, se comunica con el Servidor, el cual a través del motor de cálculo de *MAXIMA*, procesa la entrada y devuelve la solución de la expresión. Una vez generada la salida, junto con su respectiva entrada serán estructuradas en formato LaTeX [36] y texto plano, para ser enviadas al *Front End*, donde los elementos en texto plano aplicando TTS serán leídos por un

sintetizador de voz. La salida en formato LaTeX servirá como el elemento de entrada dentro del módulo de sonificación

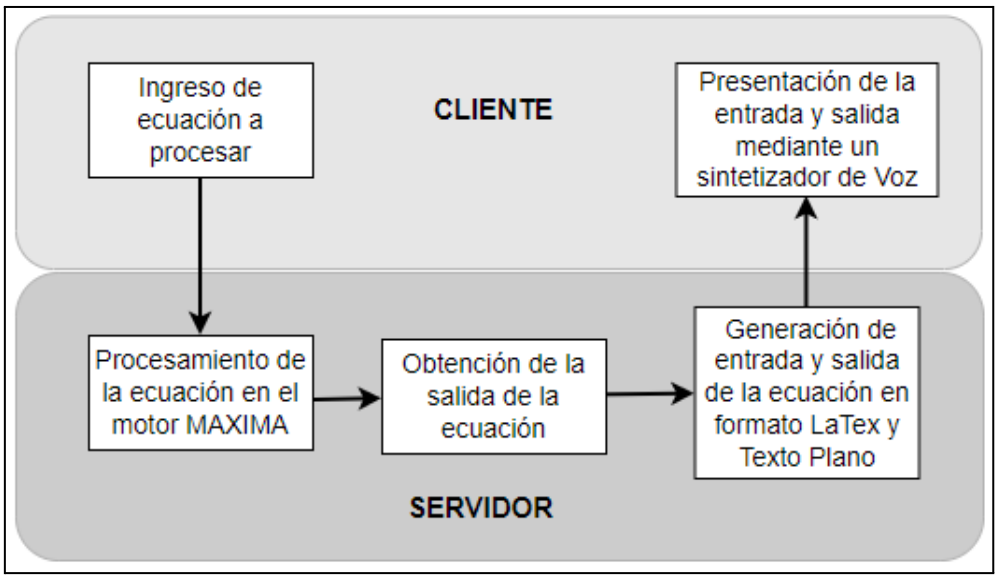


Figura 2.2 Lógica de funcionamiento para la presentación de las ecuaciones.

Actualmente la representación de la salida, solo se hace mediante una de voz sintetizada que realiza una lectura lineal, por lo que en muchos escenarios pueden existir ambigüedades en la compresión de donde empiezan y terminan los diferentes elementos de la expresión, provocando confusión por parte del usuario final (no vidente). Por ejemplo como se presenta en la Figura 2.3 la lectura de la expresiones no permite diferenciar qué argumentos componen a la raíz, así como no permite establecer dónde inician y terminan los elementos que contienen al numerador y denominador de la fracción, con lo cual el usuario no podría determinar de manera exacta la estructura de la salida obtenida.

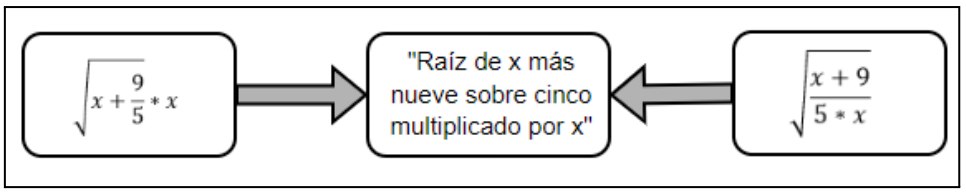


Figura 2.3 Lectura de las expresiones matemáticas

Es por esto que es necesario el desarrollo de un **Módulo de Sonificación**, que mediante el uso de las voces sintetizadas y los diferentes tipos de pistas mencionadas en el **Apartado 1.3.2.2 Parámetros y Efectos del Sonido,** se puede diferenciar los diferentes componentes de una expresión matemática. Para un análisis profundo del sistema CASVI 2.0 adjunto el **Anexo I**

2.1.2.1 Recursos Implementados en el Sistema CASVI 2.0

Como se presenta en la Tabla 2.2, los proyectos que componen al Sistema CASVI 2.0 implementan un conjunto de lenguajes de programación, estructuras de datos, librerías de código y entornos de ejecución[7], como recursos necesarios para su correcto funcionamiento.

Tabla 2.2 Recursos necesarios en el Sistema CASVI 2.0

Herramienta	Descripción
JavaScript **[2]**	Lenguaje de programación que se compila en tiempo de ejecución, el cual es utilizado para el desarrollo de páginas y aplicaciones web, y se caracteriza por ser multiparadigma y con soporte orientado a objetos [2].
Python **[31]**	Lenguaje de programación de alto nivel, multiplataforma y compilado en tiempo de ejecución, que se aplica en el desarrollo de software, aplicaciones web y análisis de datos.
Node.js **[32]**	Entorno de ejecución basado en código libre y carácter asíncrono y no multihilo[8] destinado al desarrollo de software basado en JavaScript, que permite el diseño e implementación de aplicaciones web.
MAXIMA **[33]**	Motor matemático multiplataforma desarrollado por el *Massachusetts Institute of Technology* y basado en código libre, el cual permite la interacción y resolución de alta precisión de expresiones algebraicas.
MongoDB **[34]**	Base de datos noSQL distribuida, de uso gratuito y basada en documentos que se caracteriza por una fácil escalabilidad, disponibilidad y flexibilidad, en la cual los datos son almacenados en formato JSON[9] y agregados en tiempo real.
Vue.js **[35]**	Entorno de ejecución basado de JavaScript, utilizado para el desarrollo de interfaces de usuario, construidas mediante HTML[10], CSS y *JavaScript*[11].

[7] Se entiende como entorno de ejecución a los programas que permiten, mediante servicios, la ejecución de un software.
[8] Multihilo se refiere a que varios procesos pueden ser ejecutados al mismo tiempo dentro del mismo programa.
[9] JSON (*JavaScript Object Notation*) es un formato de datos compatible con JavaScript.
[10] Lenguaje de Marcado de hipertexto utilizado para estructuración de páginas web
[11] *Common Style Sheet* (CSS) es lenguaje de estilos que permiten la personalización de páginas web

***LaTeX* [36]**	Sistema de preparación de documentos que permite estructurar datos mediante comandos (etiquetas), utilizado para una presentación visual de alta calidad de textos y el tipeo de fórmulas matemáticas.
***Content-MathML* [37]**	Lenguaje de marcado que se basa en el formato XML, que es utilizado para representar expresiones matemáticas dentro de una estructura de datos.
***SnuggleTeX* [38]**	Librería de código abierto basada en *Java,* orientada a la conversión de expresiones matemáticas en formato LaTex a un formato *MathML*.
***WebSpeech* [39]**	Librería de código de *JavaScript* que permite la representación de información dentro de una aplicación web mediante el reconocimiento de datos (*Speech Recognition)* y la reproducción de voces sintetizadas (*Text to Speech*).
***Flask* [40]**	Entorno de ejecución basada en código libre, que permite el desarrollo de aplicaciones web desarrolladas en *Python*.
***Maven* [41]**	Software de gestión de proyectos basado en el concepto de **Programación Orientada a Objetos (POO)**, que permite la ejecución de códigos basados en Java[12].

2.1.3 DEFINICIÓN DE REQUERIMIENTOS

El establecimiento de Requerimientos Funcionales y No Funcionales, es un punto muy importante para sustentar el desarrollo apropiado de cualquier módulo de software. Dentro de este Apartado, se determinan todos los aspectos que deben cumplirse en el desarrollo e integración del **Módulo de Sonificación** dentro del sistema CASVI 2.0.

Previo a definir los distintos requerimientos, cabe destacar que las aplicaciones dirigidas a personas con discapacidad visual, deben considerar los diferentes conceptos de accesibilidad nombrados en el **Apartado 1.3.1 Estado Del Arte De Software Para Personas Invidentes**, con el objetivo de asegurar una correcta interacción por parte del usuario.

[12] Lenguaje de programación orientado al desarrollo de de sistemas basados en objetos

2.1.3.1 Requerimientos Funcionales

Los requerimientos de carácter funcional puntualizan todas aquellas funcionalidades que son necesarias de integrar en la versión definitiva del módulo, con el propósito de cumplir con todos los objetivos planteados dentro del Proyecto de Integración Curricular.

Para la recolección de requerimientos, primero se debe tomar a consideración que el Sistema CASVI 2.0 presenta tres actores dentro de su uso: Un usuario ***"Administrador"*** encargado de la administración de los usuarios del sistema; un usuario ***"Profesor"*** encargado de la gestión de los cursos, estudiantes, actividades y lecciones; y un usuario ***"Estudiante"*** que cuenta con las funciones de acceder y realizar las actividades y lecciones establecidas en el curso que pertenece, así como tener una retroalimentación de éstas.

Se debe mencionar que el actor de interés para el desarrollo del **Módulo de Sonificación** es el ***"Estudiante"*** ya que es un usuario no vidente, quien necesitará una descripción auditiva de la salida obtenida de la resolución de la expresión matemática ingresada por éste. La presentación de los diagramas Casos de Uso de cada uno de los actores se encuentra adjuntado en el **Anexo II**, con el fin de definir los Requerimientos Funcionales de mejor manera.

Tabla 2.3 Historia de Usuario para Emisión de Salida de una Ecuación.

Historia de Usuario #1	
Código: **RF-1**	Nombre: **Emisión de Salida de la Ecuación**
Usuario: **Estudiante No Vidente**	
Prioridad en Proyecto: **Alto**	Riesgo en el Proyecto: **Alto**
Proceso: Un estudiante no vidente de Ingeniería, requiere obtener el resultado de una ecuación cuadrática para la resolución de un sistema de movimiento parabólico. Para la obtención del resultado, el estudiante utiliza el espacio de trabajo del Sistema CASVI 2.0, en el cual mediante teclado ingresará la ecuación a solucionar y ejecutará el comando "Ctrl + R" para su respectiva resolución. Una vez obtenida la salida de la ecuación, el estudiante ejecutará el comando "Ctrl + L" y podrá obtener una descripción auditiva de la expresión matemática que compone el resultado de la ecuación.	
Requerimiento: Se debe generar una salida de audio clara y descriptiva que permita al usuario final poder comprender y reconocer toda la expresión matemática.	

Como se logra observar en la Tabla 2.3, se procede a la Historia de Usuario #1, obtenida a través del análisis de las pruebas piloto ejecutadas en el Sistema CASVI 2.0.

2.1.3.2 Requerimientos No Funcionales

Los requerimientos de tipo no funcional establecen parámetros que complementan lo definido por los Requerimientos Funcionales. A través de éstos se busca asegurar la conformidad por parte del usuario mediante un software dotado de calidad.

Los requerimientos No Funcionales más importantes son definidos continuación:

Tabla 2.4 Historia de Usuario para Emisión de Salida de una Ecuación.

Historia de Usuario #2	
Código: **RNF-1**	Nombre: **Reproducción de Salida de la Ecuación**
Usuario: **Estudiante No Vidente**	
Prioridad en Proyecto: **Alto**	Riesgo en el Proyecto: **Alto**
Proceso: Un estudiante no vidente de Ingeniería, ha mandado a procesar una ecuación y una vez obtenida la salida de la ecuación, el estudiante ejecutará el comando “Ctrl + L” y podrá obtener una descripción auditiva de la expresión matemática que compone el resultado de la ecuación y comenzará a reproducirse.	
RNF1: La velocidad de reproducción de la salida de audio generada debe contribuir a que el usuario no pierda la atención. **RNF2:** La salida de audio generada debe contar con un nivel de volumen dentro de los rangos que eviten afectaciones a la salud del usuario. **RNF3:** La activación del módulo de sonificación debe ser fácil de invocar por parte del usuario.	

Además de los Requerimientos No Funcionales definidos en la Historia de Usuario #2, se debe tener en consideración que dentro del proceso de generación del audio descriptivo se utilizará una API online mediante el *Google Translator*, por lo que se define:

RNF4: El sistema debe contar con una conexión a Internet.

2.1.4 CONSIDERACIONES DE DISEÑO

Para el desarrollo de un **Módulo de Sonificación** óptimo que ayude al usuario no vidente a comprender la salida de expresión resuelta y cumpla con los requerimientos definidos se han determinado una serie de consideraciones dentro de su diseño:

CD1. Se debe generar una descripción auditiva de los diferentes componentes que conforman la estructura de la expresión matemática

CD2. Se deben reproducir tonos característicos que permitan identificar los diferentes componentes de salida de la operación matemática (i.e raíces, fracciones, presencia de paréntesis, funciones trigonométricas, entre otros), con la finalidad de que se tenga una referencia de los elementos más usados, que integran el inicio y final de la expresión.

CD3. La pronunciación por parte de la voz sintetizada debe ser clara, de manera que se eviten ambigüedades y malentendidos en la comprensión de la salida reproducida.

CD4. Los tonos generados como pistas auditivas no deben ser estruendosos ni molestos, de manera que no afecten la atención del usuario. Esto se logrará evitando el uso de tonos muy agudos y amplitudes muy altas.

CD5. El procesamiento del módulo de sonificación no deberá demorar más de 5 segundos, con el objetivo de evitar que una demora muy pronunciada para la reproducción de la salida, produzca inconformidades de parte del usuario.

CD6. Las pistas a utilizar no deben ser sobrecargadas o de larga duración (mayor a los 0.5 segundos), con el propósito de evitar confusiones por parte del usuario, debido a la gran cantidad de información a asimilar y comprender.

CD7. Al momento de utilizar una combinación entre los recursos auditivos y las voces sintetizadas, se debe estimar la carga cognitiva con la cual el usuario puede lidiar, pues como lo muestra el Centro de Estadísticas y Evaluación de la Educación de Australia: *“la memoria de trabajo de una persona promedio solo es capaz de retener aproximadamente cuatros fragmentos de información”* [23]. Es por esta razón que se limitará al uso de cuatro pistas auditivas diferentes que representarán diferentes categorías de operador.

- Categoría 1: Fracciones.
- Categoría 2: Raíces y potencias.
- Categoría 3: Paréntesis.
- Categoría 4: Funciones trigonométricas.

La determinación de estas categorías también radica en que aunque el Sistema CASVI 2.0 tiene la capacidad de resolver operaciones de alto nivel gracias a MAXIMA, las expresiones matemáticas de nivel medio (raíces, potencias, operaciones trigonométricas, entre otros) son las más aplicadas y con ello las más necesarias de que contar con una pista auditiva. Por lo cual no es necesario determinar operadores de mayor complejidad, ya que esto solo afectaría la comprensión del usuario al sobrepasar su carga cognitiva [8].

CD8. La invocación del **Módulo de Sonificación** se logrará a través de un evento generado mediante el uso de un atajo de teclado.

2.1.5 DIAGRAMA FLUJO DEL MÓDULO DE SONIFICACIÓN DE ECUACIONES

Como se comentó en apartados anteriores, el Sistema CASVI 2.0 actualmente presenta limitaciones en la representación auditiva de la salida de la operación matemática, pues por el momento se da una lectura lineal de la expresión, sin determinar los diferentes elementos que la componen, lo cual puede llegar a confundir a los usuarios finales; es por este motivo que, se ha planteado desarrollar un **Módulo de Sonificación** para la salida obtenida en el Sistema CASVI 2.0.

Como se presenta en la Figura 2.3, el **Módulo de Sonificación** debe permitir tomar la estructura de una expresión matemática en un formato de lenguaje de marcado como lo es LaTeX, transformarla en una cadena de texto en *"lenguaje natural"* que describa la composición de la expresión matemática y sea leída por una voz sintetizada, acompañada por una secuencia de tonos y efectos de sonido que funcionarán como pistas auditivas[13].

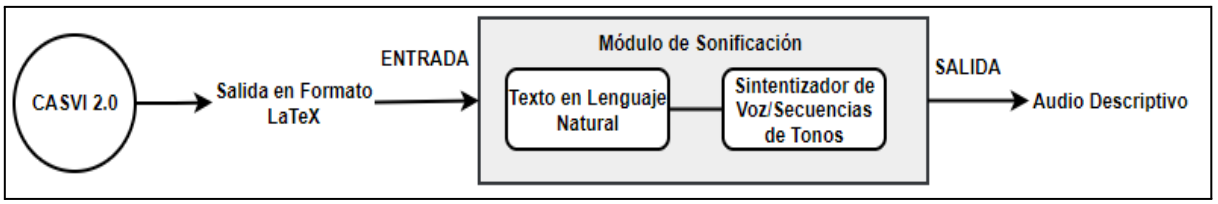


Figura 2.3 Sonificación de la expresión matemática

Con esto en consideración, el desarrollo y utilización de éste módulo contará con las siguientes pautas:

[13] Se define como pistas auditivas a los recursos que permitan ayudar en la comprensión de una estructura u operador matemático, mediante el uso de tonos y efectos distintivos generados por la manipulación de los parámetros del sonido.

- La sonificación de la salida será activada mediante atajos de teclado:
    - Alt + T → Reproducción en velocidad lenta.
    - Alt + G → Reproducción en velocidad media.
    - Alt + B → Reproducción en velocidad rápida.

- Partiendo de la salida en formato LaTeX, desde el lado del Cliente (*Front-End*), como se muestra en Figura 2.4, mediante un componente encargado de modificar la salida de LaTeX, se realizarán los cambios pertinentes en su estructura de manera que sean compatibles con las funcionalidades de *Mathlive*[44] que será la librería encargada de realizar la conversión de LaTeX a lenguaje natural. Por esto es necesario revisar si la expresión matemática en la salida obtenida contiene fracciones o potencias cuyo exponente contenga una fracción. Esto se debe a que en la salida de LaTeX original se utiliza la etiqueta "*\over*" para representar la presencia de una fracción ("*{numerador}\over{denominador}*"), lo cual presenta limitaciones al momento de procesarse con *Mathlive,* pues no permite delimitar el inicio y final de la fracción. Para la corrección de este problema se debe reemplazar esta estructura a una que posea la etiqueta "*\frac*", lo cual permitirá a *Mathlive* determinar los argumentos que componen al numerador y denominador de la fracción ("*\frac{numerador}{denominador}*") y así poder describir el inicio y final de la fracción.

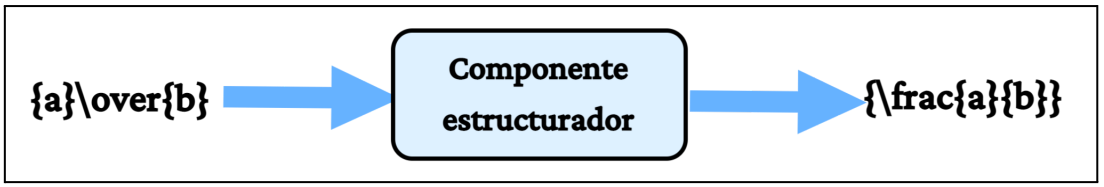


Figura 2.4 Cambio de estructura de una expresión matemática.

- Con la estructura de LaTeX conseguida, se procederá a utilizar la librería de *Mathlive* para realizar una conversión de este formato a una cadena de texto en lenguaje natural, ya que la generación de audio mediante la herramienta TTS necesita como entrada el texto descriptivo que será leído por la voz sintetizada. Es por esto también que la cadena de texto en lenguaje natural debe presentar a detalle todos los elementos (aclarando su inicio y fin) que forman parte de la expresión algebraica, funcionando así como una pista léxica que guiará al usuario entre los diferentes argumentos y operadores que componen a la expresión; por ejemplo como se logra observar en la Figura 2.5, se toma la estructura de LaTeX que representa una raíz cuadrada y mediante la librería de *Mathlive* se tendrá la descripción de la expresión en

lenguaje natural. Este *string* será procesado por el Módulo de Sonificación en el lado del Servidor.

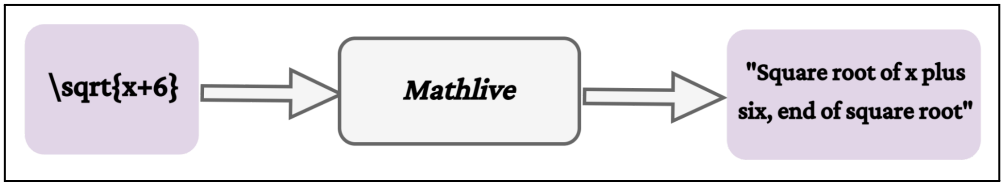


Figura 2.5 Representación de expresión matemática en lenguaje natural

- La conversión a lenguaje natural realizada por la librería de *Mathlive* generará una cadena de texto en inglés, por lo cual como se muestra en la Figura 2.6, en el **Módulo de Sonificación** en el lado del Servidor, antes de procesar este *string* es necesario traducirlo al español.

Figura 2.6 Traducción de cadena de texto al español.

- Antes de iniciar con el proceso de traducción al español, a la cadena de texto se deben realizar cambios que faciliten la traducción, ya que muchas veces las contracciones generadas en el inglés o el uso de símbolos para representar operadores, pueden provocar incongruencias en la cadena traducida. Por ejemplo, como se observa en la Figura 2.7, en el inglés un paréntesis puede ser representado con la palabra *paren*, pero al momento de utilizar un traductor se puede generar la confusión de que esto se traducido a “padre”, por lo cual para evitar este problema, se debería hacer un reemplazo de *paren* por *parenthesis*.

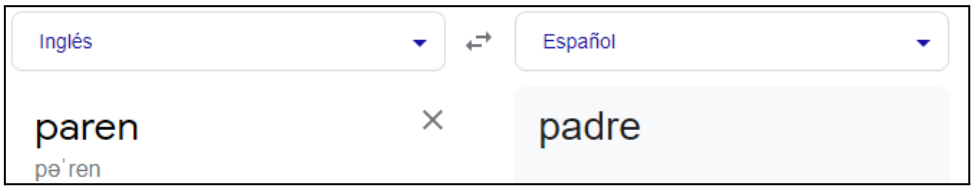


Figura 2.7 Traducción de *paren* al español.

- Debido a que el audio será reproducido en el *Front-End* desde el lado del Cliente y el procesamiento para generarlo lo realiza el **Módulo de Sonificación** desde el lado del Servidor, la comunicación entre estos se realizará mediante una solicitud *HTTP* del tipo *“POST”,* que enviará como argumento la cadena de texto en lenguaje natural.

- Como se presenta en la Figura 2.8, la cadena de texto recibida será procesada con el propósito de identificar los diferentes operadores matemáticos que han sido definidos y separarla en porciones de texto, denominadas *"fragmentos"*, que serán delimitados por el respectivo operador.

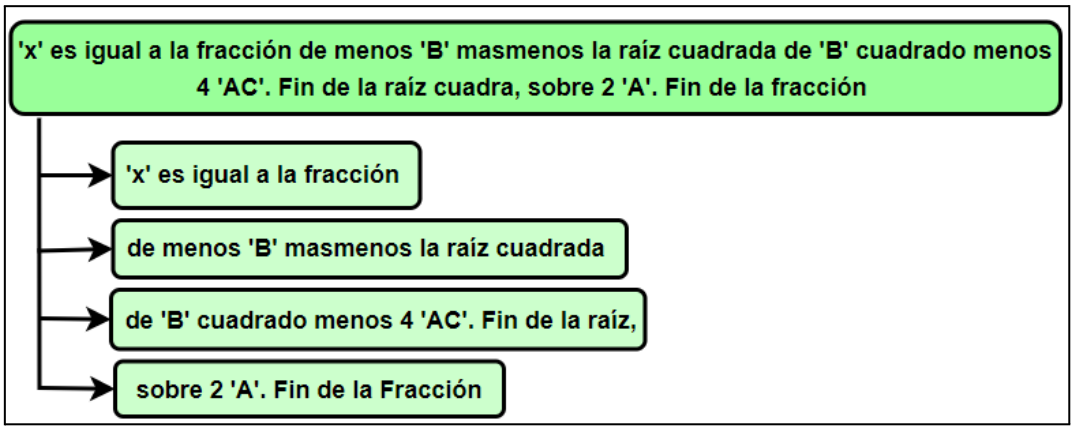


Figura 2.8 Transformación de cadena de texto en fragmentos.

- Los operadores matemáticos utilizados como delimitadores de los fragmentos a generar, serán aquellos que puedan contener uno o más elementos dentro de ellos. Estos pueden ser fracciones, raíces, paréntesis, funciones trigonométricas, entre otros. Como se muestra en la Figura 2.8 las separaciones en la cadena de texto se realizan al inicio y final de la fracción y de la raíz, pues permitirá reconocer los argumentos que se encuentran dentro de estos operadores.

- Como lo presenta la Figura 2.9, cada fragmento de texto será grabado con una voz sintetizada, combinado con el efecto de *Fade Out*[14] (funcionando como guía de que el fragmento está por terminar) y concatenado con el tono respectivo que cumplirá con el rol de pista auditiva para identificar al operador matemático que está funcionando como delimitador.

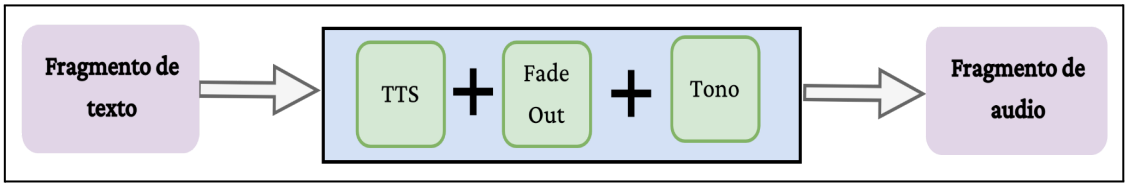


Figura 2.9 Grabación de fragmentos.

- Todos los fragmentos grabados serán consolidados en un solo archivo de audio que será enviado al *Front End* para ser reproducido.

[14] *Fade Out* se refiere al efecto de sonido que genera un decaimiento progresivo del nivel de volumen

- Las secuencias de audio deben evitar el sobrelapamiento entre ellas, de manera que no generen una desorientación al momento de que el usuario reproduzca las diferentes diferentes celdas que conforman el área de trabajo.

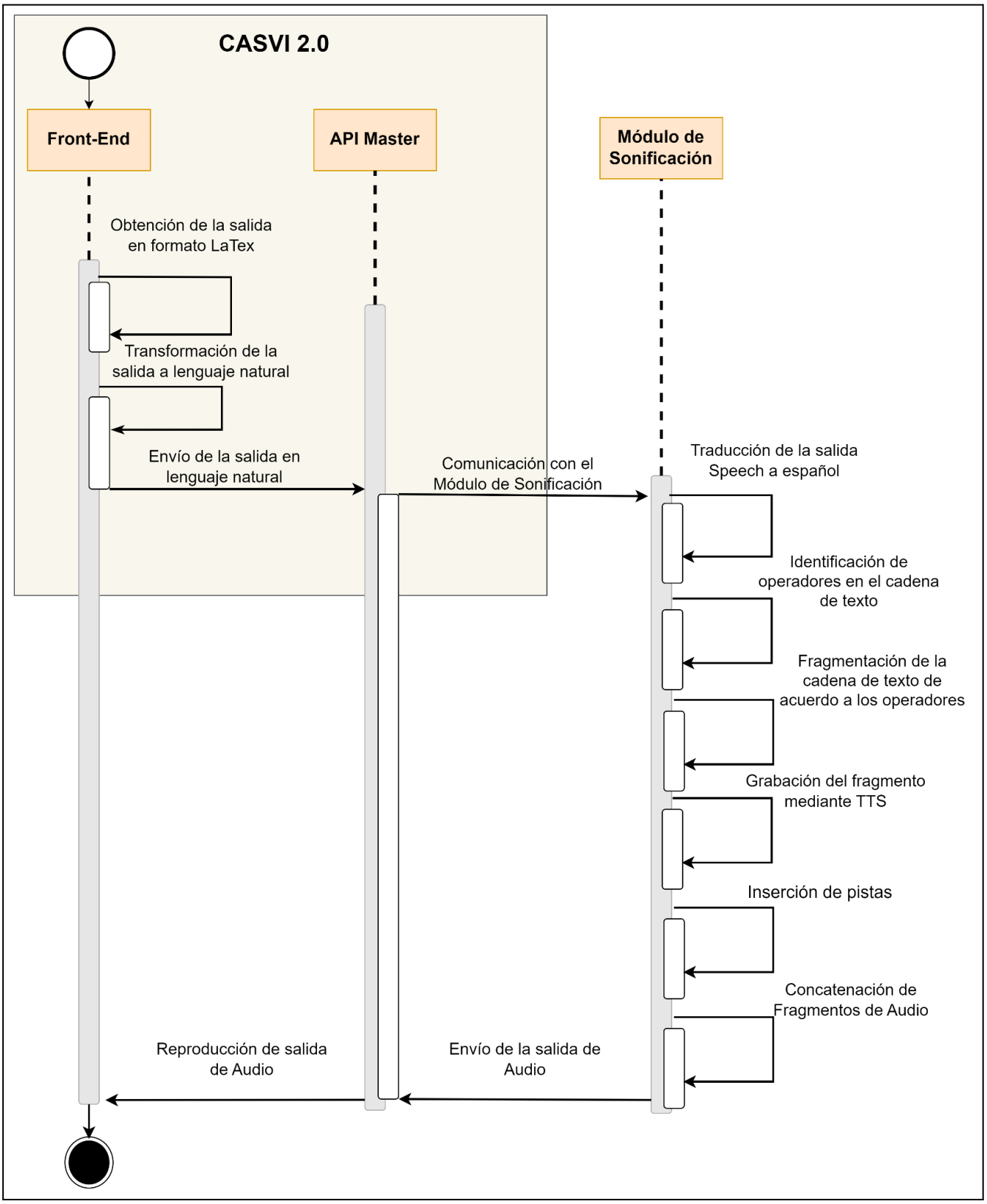


Figura 2.10 Diagrama de Flujo del proceso sonificación de ecuación.

Como se muestra en la Figura 2.10 el proceso sonificación comenzará desde el lado del Cliente, para luego comunicarse con el **Módulo de Sonificación** mediante el proyecto ***API-Master*** mencionado en el **Apartado 2.1.2 CASVI 2.0 y su Estructura**, hacia el lado del Servidor, el cual procesa toda la cadena texto y la convierte en una salida de audio compuesta del dictado del *string* por parte de la voz sintetizada junto con las pistas auditivas añadidas.

Figura 2.11 Diagrama de flujo del Módulo de Sonificación.

Para un análisis más profundo del diseño del **Módulo de Sonificación,** es posible observar el diagrama de flujo especificado en la Figura 2.11, donde se parte de una entrada de texto en

lenguaje natural (*Speech)* que describe la estructura de la expresión matemática y uno de los tres niveles de referencia de velocidad (lento, medio y rápido) que será elegido por el usuario, debido a que muchos de estos han conseguido un mayor desarrollo de su oído y se han acostumbrado a escuchar a mayores velocidades. Cuando la entrada de texto es obtenida esta es traducida al español para luego ser separada en fragmentos delimitados por los operadores mencionados con anterioridad en las pautas de Diseño. Con los fragmentos obtenidos se procede a recorrerlos de manera iterativa y de acuerdo a la longitud del fragmento se aplica una configuración de volumen y velocidad con la que la voz sintetizada será grabada. Una vez que se haya aplicado la herramienta de TTS se integrará una pista auditiva de decaimiento (*Fade Out*) y la nueva pieza de audio se irá concatenando a los fragmentos anteriores. Dependiendo del operador delimitador que posea el fragmento, se integra su respectiva pista auditiva y en el momento que se hayan recorrido todos los fragmentos se procederá a guardar el audio concatenado y se procederá a enviarlo para su correspondiente reproducción en el lado Cliente.

La consideración de las velocidades de referencia y longitudes de las cadenas para asignar su respectiva aumento de velocidad y volumen se obtuvo de manera de empírica mediante el análisis de diferentes expresiones matemáticas y la generación de sus fragmentos, donde en promedio los fragmentos cortos son menores a 30 caracteres, los fragmentos medios poseen no más de 50 carácteres y fragmentos extensos serán mayores a los 50 carácteres. Es así que con el fin de no perder la atención del usuario se busca aumentar la velocidad y volumen (para contribuir en el entendimiento de audios más rápidos) de lectura del fragmento de acuerdo a su longitud, evitando que la presencia de muchos fragmentos extensos generen audios muy largos.

2.2 IMPLEMENTACIÓN

2.2.1 ACTUALIZACIÓN DEL TABLERO DE KANBAN.

En la Figura 2.12 es posible observar que se encuentran culminadas todas las tareas correspondientes a la fase de Diseño, además de que todas las tareas relacionadas con la fase de Implementación ahora se encuentran dentro de la columna de "Actividades en Progreso" y la tarea de realizar las pruebas de funcionamiento del **Módulo de Sonificación"** se establecería cómo la única que ubica en la columna de "Actividades por Hacer". Se puede destacar que si bien el número de tareas para la fase de Implementación es mucho mayor que en las fases de Investigación y Diseño, muchas de estas pueden ser realizadas de manera simultánea.

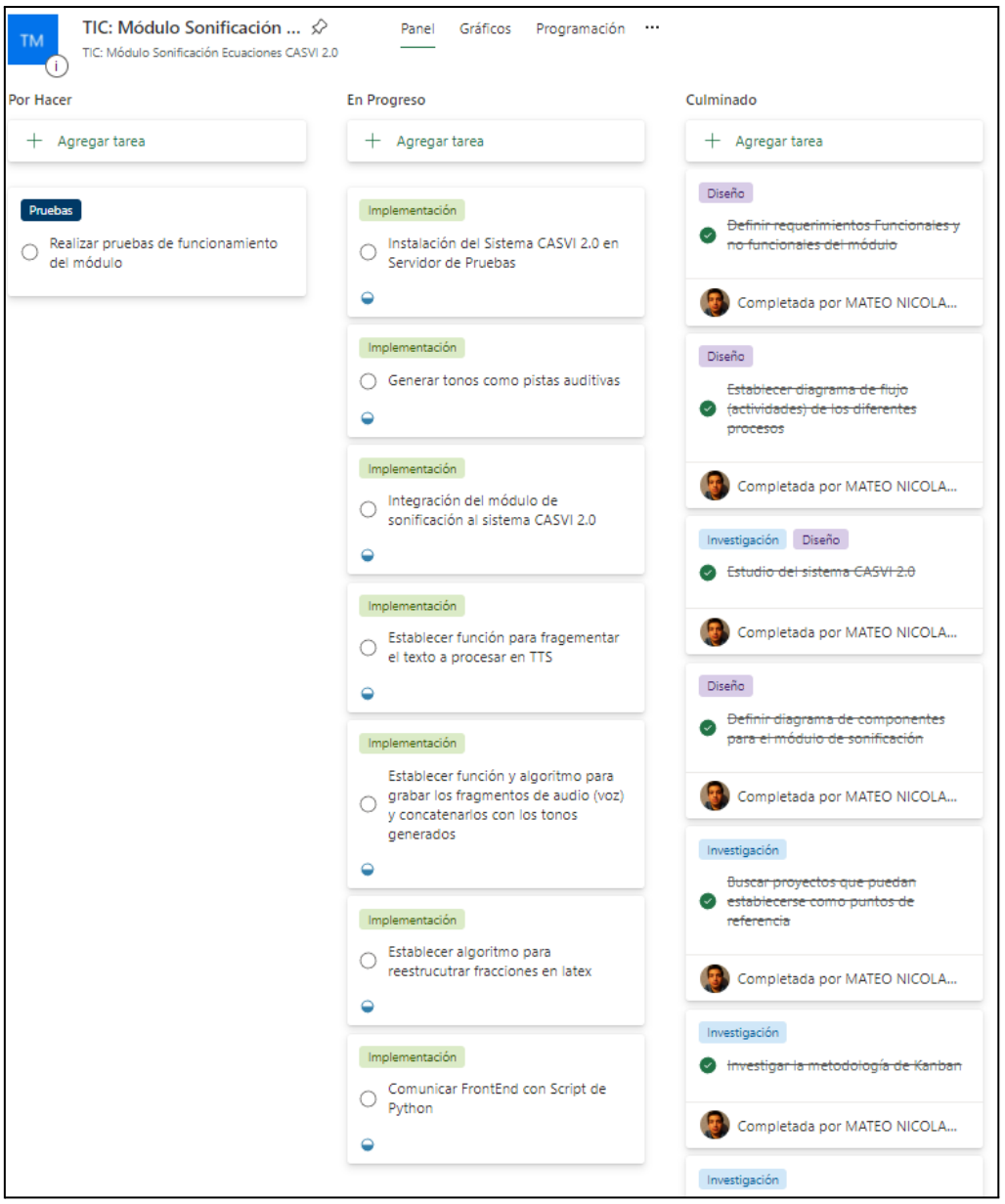


Figura 2.12 Puesta al Día del Tablero de Kanban en Fase de Implementación.

2.2.2 INSTALACIÓN DE CASVI 2.0 EN SERVIDOR DE PRUEBAS.

Para el desarrollo e incorporación del **Módulo de Sonificación** dentro del Sistema CASVI 2.0, es necesario que dentro de un servidor de pruebas que cuente con el Sistema Operativo Ubuntu 20.04 LTS, se proceda con la descarga e instalación de los diferentes proyectos de código nombrados anteriormente en el **Apartado 2.1.2 CASVI 2.0**. Cabe recalcar que se determinó el uso de un servidor de pruebas, ya que CASVI 2.0 necesita de una gran cantidad de recursos tanto en hardware como software, que no pueden ser suministrados por un computador personal.

Para el correcto funcionamiento del sistema CASVI 2.0, además de los proyectos mencionados, es necesario la descarga e instalación del lenguaje de programación de Python en sus versiones “2” y “3”, así como la instalación de *MongoDB* [34] y *Node.js* [32].

Las especificaciones mínimas del servidor y el sistema, así como el proceso detallado para la instalación del sistemas se encuentran especificados en el Manual de Instalación del Sistema CASVI 2.0 que adjunto en el **Anexo III**.

2.2.3 CODIFICACIÓN DEL MÓDULO DE SONIFICACIÓN.

Para la integración del **Módulo de Sonificación** dentro del sistema CASVI 2.0 es necesario codificar tanto en el lado del Cliente (*Front-End)* como en el lado del Servidor (*API-Master*), ya que se tomará la salida de la expresión matemática en formato LaTeX desde el lado del Cliente y se realizará un preprocesamiento que permitirá establecer la estructura óptima que facilitará la representación de la salida en lenguaje natural, para luego ser enviada al lado del Servidor para su correspondiente sonificación.

2.2.3.1 Codificación en *Front-End*

Dentro de CASVI 2.0, una vez que la expresión matemática a resolver es ingresada y enviada a procesar, la salida obtenida presenta su representación en formato LaTeX, el cual antes de ser transformado a una descripción en lenguaje natural, como se mencionó en las pautas de Diseño del **Apartado 2.1.5 DIAGRAMA FLUJO DEL MÓDULO DE SONIFICACIÓN DE ECUACIONES**, se debe hacer una reestructuración de la salida obtenida. Es por esto que para conseguir la reestructuración mencionada, en el *Front-End*, se inició por crear un nuevo componente que posea la función que se describe en el Código 2.1, la cual toma como argumento de entrada el *string* que contiene la expresión matemática en formato LaTeX. Como se muestra en la línea 10 del código se procede a entrar un bucle que permitirá encontrar todas las posibles fracciones que posea la entrada; y como se presenta en las líneas 11 y 12, mediante el uso de expresiones regulares se continúa por buscar las posiciones donde comienzan los patrones que posean las estructuras que representen una fracción o exponentes que cuenten con una fracción.

```
1.  estructurarLatex(cadena){
2.      var booleano = true;
3.      var result = cadena;
4.      var texto = "over";
5.      var index;
6.      var firstIndex;
7.      var i = 0;
8.      var matches = null;
```

```
9.      var matches2 = null;
10.         while(booleano==true){
11.             matches =
   cadena.match(/\{(.+?)\}\\over\{(.+?)\}/g);
12.             matches2 =
   cadena.match(/\{\{\{(.+?)\}\\over\{(.+?)\}\}\}/g);
13.             if(result.includes(texto) && matches2 == null){
14.                 firstIndex = result.indexOf(matches[i]);
15.                 index = firstIndex+1;
16.                 result = result.slice(0, index) + '\\frac' +
   result.slice(index);
17.                 result = result.replace('\\over', '')
18.                 i++
19.
20.             }
21.             else if(result.includes(texto) && matches2 !=
   null){
22.                 firstIndex = result.indexOf(matches2[i]);
23.                 index = firstIndex+2;
24.                 result = result.slice(0, index) + '\\frac' +
   result.slice(index);
25.                 result = result.replace('\\over', '')
26.                  i++
27.             }
28.             else {
29.               i=0
30.               booleano = false
31.               matches = null
32.               matches2 = null
33.             }
34.               }
35.         return result
36.   }
```

Código 2.1 Función para reestructurar salida de LaTeX.

Con las posiciones definidas dentro del *string* y con el patrón coincidente identificado, como se observa entre las líneas 14-17 y 22-25, se procede a ingresar la etiqueta "*\frac*" al comienzo del patrón y a su vez eliminar la etiqueta "*\over*", consiguiendo así la estructura de deseada. Por último, una vez que se ha terminado de iterar dentro del bucle como se define en la línea 35 del código se retorna una cadena de texto salida de LaTeX reestructurada. Con la nueva estructura de LaTeX obtenida se procede a utilizar la librería de *Mathlive* para obtener la descripción de la ecuación en lenguaje natural, que será almacenada en una variable, que servirá como un argumento a enviar dentro de la solicitud al **Módulo de Sonificación** en el lado del Servidor.

Para realizar la solicitud al Servidor, como se muestra en el Código 2.2 se crea una función denominada **"escucharSalida"** la cual tendrá como argumentos de entrada la descripción de la ecuación en lenguaje natural y el nivel de velocidad con la que se deberá sonificar la

ecuación (que dependerá del atajo de teclado utilizado por el usuario). Cómo es posible observar en la línea 3, si la cadena de texto con la descripción de la expresión matemática no se encuentra vacía, se procede a crear un objeto del tipo *FormData*[15] que como se mira en las líneas 8 a 11 permite adjuntar en una solicitud *HTTP* del tipo *"POST"*[16] los argumentos de entrada, para solicitar la generación del audio en el lado del Servidor.

```
1.  escucharSalida(ttsvar, velocidad)
2.      {  // Comunicacion con script de python para generar
   audio para representar salida de la ecuacion
3.      if(ttsvar!="")
4.      {
5.      var formData = new FormData();
6.      var scriptAreaTxt = ttsvar;
7.      var rate = velocidad;
8.      formData.append("scriptTxt", scriptAreaTxt);
9.      formData.append("rate", velocidad);
   fetch(process.env.VUE_APP_BACKEND+":8000/generarAudio", {
10.          method: "POST",
11.          body: formData,
12.        })
13.          .then((response) => {
14.            return response.json();
15.          })
16.          .then((myJson) => {
17.            try {
18.              var resultado = myJson
19.              console.log("ResJson")
20.              console.log(myJson);
21.              return resultado
22.            } catch (e) {
23.              console.log(e);
24.            }
25.          });
26.        }
27.      }
```

Código 2.2 Solicitud al servidor.

2.2.3.2 Codificación en *API-Master*

Para iniciar con el proceso de sonificación desde el lado del Servidor es necesario establecer la ruta con la cual se recibirán las solicitudes *HTTP* realizadas desde el lado del Cliente. Es así que como se presenta en el Código 2.3, esta ruta se encuentra definida por la función ***"generarAudio"***, la cual como se observa en la línea 3, si reconoce que la solicitud realizada es del tipo *"POST"*, continúa por obtener los valores de los parámetros enviados y añadirlos como argumentos de entrada en la llamada al **Módulo de Sonificación** (línea 8), que se encargará de todo el proceso de generación del audio que describa la expresión matemática.

[15] Tipo de objeto que permite ingresar conjuntos de datos clave-valor
[16] Solicitud HTTP que envía parámetros que puedan realizar cambios en el servidor.

Como se contempla en las líneas 9 a 16, en caso de que no se genere ningún problema se retornará una respuesta al Cliente con status 200 (“ok”).

```
1.  @app.route('/generarAudio', methods=['GET', 'POST'])
2. def generarAudio():
3.     if request.method == 'POST':
4.         scriptTxt  = request.form.get('scriptTxt')
5.         rate  = request.form.get('rate')
6.         print("***********GEN AUDIO***********")
7.         print(scriptTxt)
8.
   respuestaJson={"salidaAudio":smcasvi.sonificar(scriptTxt,rate)
   }
9.         response = make_response(
10.               jsonify(
11.                   respuestaJson
12.               ),
13.               200,
14.           )
15.           response.headers["Content-Type"] =
   "application/json"
16.           return response
17.       else:
18.           pass
```

Código 2.3 Recepción de solicitud al Módulo

Dentro de lo que corresponde al **Módulo de Sonificación** como tal, como se mostró en el **Apartado 2.1.5 DIAGRAMA FLUJO DEL MÓDULO DE SONIFICACIÓN DE ECUACIONES**, la cadena de texto en lenguaje natural generada por *Mathlive* está en inglés. Es así que lo primero que se realiza dentro del módulo es traducir esta expresión al español; para lo cual, como se presenta en el Código 2.4, se ha creado la función ***“traducir”*** para que se encargue de ello.

```
1. def traducir(auxEntrada):
2.     translator = Translator()
3.     def replacement(texto):
4.         texto = texto.lower()
5.         texto = texto.replace("paren.", "parenthesis.")
6.         texto = texto.replace("+", "plus ")
7.         texto = texto.replace("end", "end of")
8.         texto = texto.replace("pm", "plusminus")
9.         texto = texto.replace("lnopen", "Ln open")
10.          texto = texto.replace("logopen", "log open")
11.          texto = texto.replace("close", "closed")
12.          texto = texto.replace("pie", "pi ")
13.          texto = texto.replace("'x'", "equis")
14.          texto = texto.replace("'i'", "i ")
15.          texto = texto.replace("'b'", "b ")
16.          texto = texto.replace("'c'", "c ")
```

```
17.          texto = texto.replace("'d'", "d ")
18.          texto = texto.replace("'e'", "e ")
19.          texto = texto.replace("comma", ",comma")
20.          texto = texto.replace(".", "")
21.          return texto
22.      resultado = replacement(auxEntrada)
23.      result = translator.translate(resultado, src='en' ,
   dest='es')
24.      print(result.text)
25.      return result.text
```

Código 2.4 Traducción de expresión al español.

Como se observa en la línea número 2 del código se inicia por crear un objeto del tipo *Translator*, que es una clase que pertenece a la librería de *Googletrans* [45] y se encarga de realizar la traducción de textos mediante la *API* de *Google Translator*. Antes de iniciar con el proceso de traducción, como se explica en el **Apartado 2.1.5 DIAGRAMA FLUJO DEL MÓDULO DE SONIFICACIÓN DE ECUACIONES** se procede a utilizar una función interna que como se muestra entre las líneas 3 a 21, se encarga de realizar cambios en la cadena de texto que facilitarán la traducción de éste.

Con los cambios aplicados en la cadena de texto, se continúa por realizar la traducción para lo cual se utiliza la función *translate* del objeto de tipo *Translator* y como se presenta en la línea número 23 del código, se ingresarán el texto a traducir, el idioma original y el idioma la que se quiere traducir, siendo en este caso inglés (*'en'*) y español (*'es'*) respectivamente. Finalmente esta función retornará la expresión matemática en lenguaje natural ya traducida al español.

Ya que se ha conseguido que la expresión se encuentre traducida al español, se procederá a generar los fragmentos de texto que serán delimitados por los operadores anteriormente mencionados en el **Apartado 2.1.5 FLUJO DEL MÓDULO DE SONIFICACIÓN**; y para ello se ha definido la función ***"separar"***, y como se presenta en la línea número 2 del Código 2.6, se listan las diferentes formas en las que los operadores se pueden presentar dentro de la cadena de texto y como se muestra en la línea 4 del código se aplicarán expresiones regulares para encontrar patrones coincidentes con la lista de delimitadores y de manera consecuentemente generar las separaciones de la cadena de texto en fragmentos (línea 5). Por último en caso de que entre los fragmentos generados el último de estos sea un elemento vacío o sólo presente el caracter "punto" se optará por eliminar este fragmento para así evitar errores al momento de utilizar la herramienta TTS.

```
1. def separar(cadena):
2.     delimiters = ["fin de fracción.","la fracción", "raíz
```

```
   cuadrada", "Fin de la raíz cuadrada" , "Fracción final",
   "fracción final", "Paréntesis abierto.", "abierto", "Cerrar
   paréntesis.", "cerrar paréntesis.", "cerrado", "cierra
   paréntesis.", "seno", "coseno", "tangente"]
3.      example = cadena
4.      regex_pattern =
   '|'.join('(?<={})'.format(re.escape(delim)) for delim in
   delimiters)
5.      exampleResult = re.split(regex_pattern, example)
6.      if exampleResult[-1] == '' or exampleResult[-1]=='.':
7.          print("if")
8.          exampleResult.pop(-1)
9.          print(exampleResult)
10.       return exampleResult
```

Código 2.5 Función para generar fragmentos.

Antes de empezar con la grabación de los fragmentos de audio se debe contar con los tonos que funcionarán como pistas auditivas, y como lo muestra el Código 2.6 se utilizará el objeto *Mixer* de la librería *Tones* [50] con la cual se definirá una frecuencia de muestreo y el nivel de amplitud. En las líneas 2 y 3 del código se crearán los tonos base en los cuales se determinarán los parámetros como el tipo de onda con la cual se trabajará siendo en este caso ondas senoidales, el tiempo en el cual iniciará así como los efectos de vibrato y decaimiento que contribuirán en la creación de tonos diferenciados. En lo que respecta a las líneas 3 y 4 del código, a los tonos base creados se los asociará con diferentes notas musicales de manera que con estas se obtenga una melodía única y se realizará una mezcla de éstas, para que finalmente la pista auditiva creada sea guardada en archivo .WAV[17] dentro del Servidor.

```
1. mixer = Mixer(44100, 0.60)
2. mixer.create_track(0, SINE_WAVE, vibrato_frequency=0.0,
   vibrato_variance=0.0, attack=0.01, decay=0.1)
3. mixer.create_track(1, SINE_WAVE, vibrato_frequency=10.0,
   vibrato_variance=0, attack=0.01, decay=0.1)
4. mixer.add_note(0, note='b', duration=0.5 , endnote='c#')
5. mixer.add_note(1, note='c', duration=0.5, endnote='d')
6. mixer.write_wav('tone1.wav')
```

Código 2.6 Generación de pistas auditivas.

Para la transformación de los fragmentos de texto a fragmentos de audio que finalmente resultarán en un archivo final de audio que contenga a toda la expresión matemática sonificada se define la función ***"recorPlayer"*** la cual tomará como argumentos de entrada las lista de fragmentos anteriormente generados y la velocidad de referencia con la que se realizarán las grabaciones. Como se muestra en el Código 2.7, en la primera parte de la

[17] Formato de archivo de audio, que no muestra compresiones.

función, de las líneas numero 6 a 9, mediante la el objeto *AudioSegment* de la librería *Pydub* [49] se invocarán las pistas auditivas anteriormente creadas, asi como lo muestra la línea número 10 se creará un objeto de audio que en primera instancia se encuentra vacío.

```
1. def recorPlayer(exa, rate):
2.     medium = 30
3.     large = 50
4.     masv = 0.15
5.     plusrate = 20
6.     tono1 = AudioSegment.from_wav("audio/tonos/tone1.wav")
7.     tono2 = AudioSegment.from_wav("audio/tonos/tone2.wav")
8.     tono3 = AudioSegment.from_wav("audio/tonos/tone3.wav")
9.     tono4 = AudioSegment.from_wav("audio/tonos/tone4.wav")
10.       sonido = AudioSegment.empty()
```

Código 2.7 Definición de objetos de audio.

Continuando con la segunda parte de la función ***"recorPlayer"***, una vez que los objetos de audio se han definido se procederá a entrar en un bucle para iterar cada uno de los fragmentos para su respectiva grabación; como se presenta en las líneas número 4 a 15 del Código 2.8, se procederá a iniciar el motor TTS de la librería *Pyttsx3* [43] determinando en primera instancia la propiedad del idioma de la voz sintetizada que en este escenario será la correspondiente (línea 6) así como también se determinará un volumen y una velocidad de reproducción dependiendo del tamaño (longitud) de fragmento con el que se trabajará (líneas 8 a 15).

Una vez que el motor TTS se ha instanciado, como se muestran las líneas número 16 a 23 se procederá grabar el fragmento de audio y dependiendo del tipo de operador que esté funcionando como delimitador se integrará una pista auditiva al fragmento y se concatenará con el objeto de audio creado en el fragmento de código anterior, tal y como se muestra en la línea número 24 del código.

```
1. for i in exa:
2.         string = i.split(" ")
3.         if("raíz" in string or "elevado" in string or
   "potencia;" in string):
4.             record = pyttsx3.init()
5.             voices = record.getProperty('voices')
6.             record.setProperty('voice', voices[20].id)
7.             volume = record.getProperty('volume')
8.             if(len(i) >= medium and len(i) < large):
9.                 record.setProperty('rate',rate + plusrate)
10.                   record.setProperty('volume', volume + masv)
11.               elif(len(i) >= large):
12.                   record.setProperty('rate',rate +
   2*plusrate)
13.                   record.setProperty('volume', volume +
   2*masv)
```

```
14.                 else:
15.                     record.setProperty('rate',rate)
16.                 record.save_to_file(i, "audio/salidas/eq1.mp3")
17.                 record.runAndWait()
18.                 del record
19.                 time.sleep(0.1)
20.                 audSeg =
    AudioSegment.from_mp3("audio/salidas/eq1.mp3")
21.                 audSeg.fade_in(100).fade_out(100)
22.                 audSeg.export("audio/salidas/ecuacion.wav",
    format="wav")
23.                 sonido1 =
    AudioSegment.from_wav("audio/salidas/ecuacion.wav")
24.                 sonido = sonido + sonido1 + tono1
```

Código 2.8 Grabación del fragmento.

Una vez que todos los fragmentos han sido iterados se procede a salir del bucle y como lo muestra el Código 2.9, la función ***"recorPlayer"*** se procederá a tomar el objeto de audio que contiene todos los fragmentos de audio concatenados y se lo exportará a un archivo de audio con formato .WAV. Cómo es posible observar en la Figura 2.13 una vez que la función ***"recorPlayer"*** haya sido ejecutada el archivo de audio que contiene la expresión matemática sonificada se guardará dentro del Servidor.

```
1. sonido.export("audio/salidas/concat.wav", format="wav")
```

Código 2.9 Exportación de la expresión sonificada.

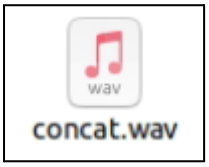


Figura 2.13 Creación de archivo de audio con la ecuación sonificada.

Con el archivo de audio creado, una vez que se realice la solicitud *HTTP* para su reproducción, el servidor procede a enviar el respectivo archivo al *Front-End* para lo cual se crea la función **función,** la cual como lo muestra el Código 2.10 se recuperá el archivo de audio con la descripción de la expresión y como lo muestra la linea 10 se enviará al *Front-End* en formato .WAV.

```
1. @app.route("/wav")
2. def streamwav():
3.     def generate():
4.         with open("audio/salidas/concat.wav", "rb") as
    fwav:
5.             data = fwav.read(1024)
6.             while data:
7.                 yield data
8.                 data = fwav.read(1024)
```

```
9.      response = Response(generate(),
   mimetype="audio/wav")
10.       return response
```

Código 2.10 Envío del archivo de audio al *Front-End.*

3. RESULTADOS Y DISCUSIÓN

En el presente Capítulo se presentan los resultados obtenidos tras haber concluido la fase de Implementación del **Módulo de Sonificación**, para lo cual se establecen las diferentes pruebas de funcionamiento y la validación de los Requerimientos Funcionales y No Funcionales, establecidos en el **Apartado 2.1.3 DEFINICIÓN DE REQUERIMIENTOS**, dentro de un escenario controlado.

3.1 ACTUALIZACIÓN DEL TABLERO DE KANBAN

En la Figura 3.1 se observa que toda las actividades relacionadas con la fase de Implementación del **Módulo de Sonificación** se han culminado, y ahora dentro de la columna de de "Actividades en Progreso" solo se encuentra la tarea de realizar las pruebas para la validación de Requerimientos Funcionales y No Funcionales. Además se puede destacar que en el Tablero de Kanban no se encuentra ninguna tarea dentro de la columna de “Actividades por Hacer”.

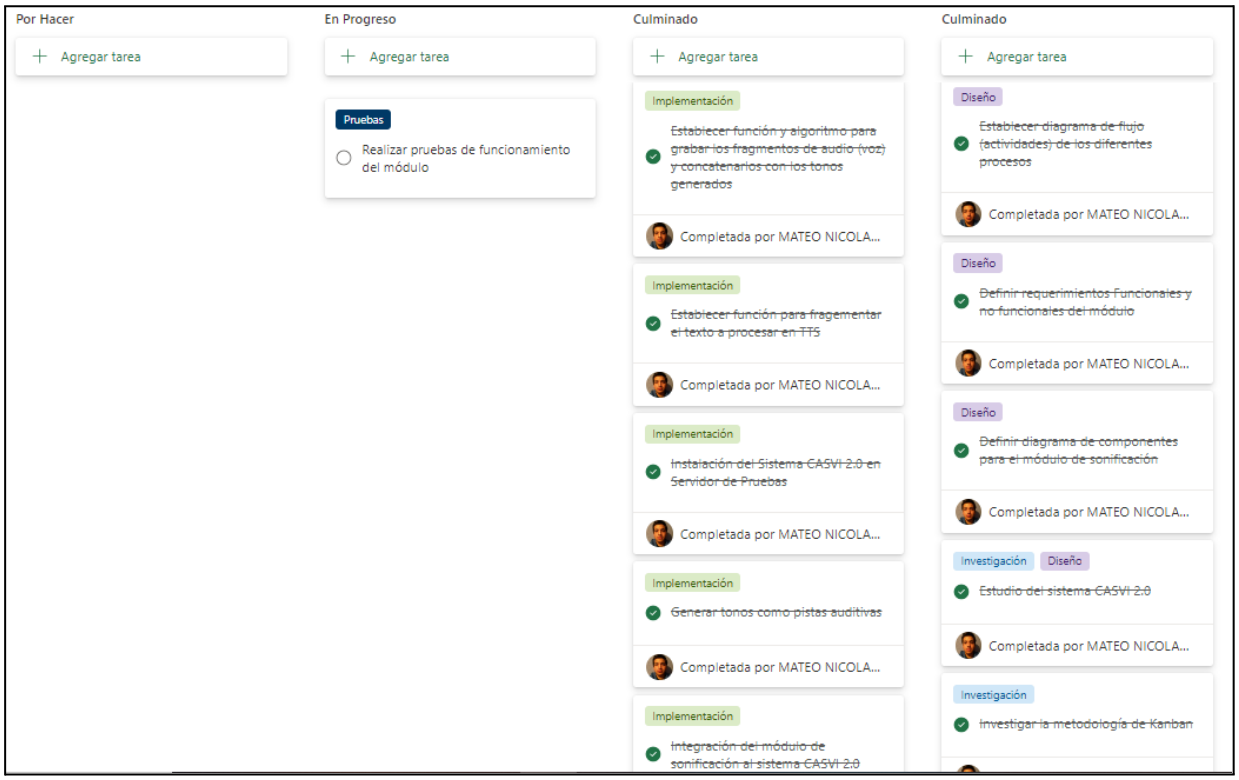


Fig 3.1 Actualización del Tablero de Kanban en la Fase de Resultados.

3.2 DEFINICIÓN DE ESCENARIO DE PRUEBAS

Para comprobar el correcto funcionamiento del **Módulo de Sonificación**, se procede a establecer un escenario de pruebas controlado, en el cual se verifique que todos los elementos que componen al módulo se encuentren operativos y cumplan con los Requerimientos Funcionales y No Funcionales de manera óptima, y mediante la simulación de un ambiente sin visión, se confirmará el correcto entendimiento de las expresiones procesadas en éste. Dentro de este escenario de pruebas se definen las siguientes consideraciones:

- Previamente se ha comprobado la correcta traducción de la descripción de la expresión obtenida de la librería de *Mathlive*.
- Las pruebas han sido realizadas de manera presencial, dentro de un laboratorio que cuente con computadoras conectadas a la red de la Escuela Politécnica Nacional.
- Los usuarios presentan completa comprensión de expresiones matemáticas de nivel medio.
- Se ha realizado la comparación entre las lecturas realizadas por el módulo de sonificación y una lectura lineal, con expresiones que fonéticamente tendrían similitud.
- Se ha revisado la reproducción de la expresión con fragmentos de distintos tamaños para corroborar los aumentos de velocidad y volumen de acuerdo a la longitud de estos.
- Se ha analizado la representación de las expresiones con las diferentes velocidades de reproducción.

3.3 VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

Para la validación de los distintos Requerimientos Funcionales y No Funcionales, se han ejecutado las distintas pruebas planteadas en el anterior Apartado y a través de la retroalimentación obtenida de encuestas aplicadas a los usuarios de pruebas, que se encuentran adjuntas en el **Anexo V**, se ha recopilado la información necesaria para determinar que todos los requerimientos planteados en la etapa de Diseño, han sido cumplidos con éxito.

3.3.1 REQUERIMIENTOS FUNCIONALES

Cómo se muestra en la Tabla 3.1 se ha comprobado la correcta traducción de las descripciones obtenidas mediante la librería *Mathlive*, en donde, como se logra observar se ha conseguido una óptima conversión del inglés al español en la cual no se presentan problemas para diferenciar los diferentes componentes de la expresión.

Tabla 3.1 Traducción de las salidas obtenidas en *Mathlive*.

Expresión	Descripción Original	Traducción
$\sqrt{x^2 + 2x - 3}$	the square root of 'X' squared plus 2 'X' minus 3. End square root	la raíz cuadrada de equis al cuadrado más 2 equis menos 3 al final de la raíz cuadrada.
$x = \frac{\sqrt{19}i+3}{2}$	‘X' equals the fraction the square root of 19. End square root 'I' plus 3, over 2. End fraction.	equis es igual a la fracción la raíz cuadrada de 19 al final de la raíz cuadrada i más 3, sobre 2 al final de la fracción.
$cos^2(x + 3) + sin^2(x)$	cosine squared Open paren. 'X' plus 3 Close paren. plus sine squared of 'X'	coseno al cuadrado paréntesis abierto equis más 3 paréntesis cerrados más seno al cuadrado de equis.
$x = 2 * (\sqrt{25\pi} + i)$	‘X' equals 2 Open paren. the square root of 25 pie . End square root plus 'I' Close paren.	equis es igual a 2 paréntesis abierto la raíz cuadrada de 25 pi final de la raíz cuadrada más i paréntesis cerrado.

Continuando con las pruebas de funcionamiento del sistema, como se mencionó en el Apartado **2.1.2 CASVI 2.0**, sin la implementación del **Módulo de Sonificación**, la lectura de las salidas de las ecuaciones se realizaba de manera lineal y con ello provocando una posible ambigüedad en su comprensión. Una vez que se ha integrado el módulo en el Sistema CASVI 2.0 ha sido posible mitigar este limitante y como se presenta en la Tabla 3.2, se ha

conseguido que expresiones matemáticas cuya lectura lineal pueden presentar similitud, ahora con la ayuda de pistas léxicas, sean descritas con un mayor nivel de especificidad (presentando los el inicio y final de los distintos componentes) y con esto sean más fáciles de diferenciar e interpretar.

Tabla 3.2 Lectura del Módulo de Sonificación.

Expresión	Lectura Lineal	Lectura Módulo de Sonificación
$\sqrt{x^2 + 2x - 3}$	raíz cuadrada de equis cuadrado más 2 equis menos 3.	la raíz cuadrada de equis al cuadrado más 2 equis menos 3 al final de la raíz cuadrada.
$\sqrt{x^2 + 2x} - 3$		la raíz cuadrada de equis al cuadrado más 2 equis final de la raíz cuadrada menos 3.
$x = \sqrt{19}\, i + \frac{3}{2}$	equis es igual a raíz cuadra de 19 i más 3 sobre 2.	equis es igual a la raíz cuadrada de 19 al final de la raíz cuadrada i más 3 sobre 2.
$x = \frac{\sqrt{19}i+3}{2}$		equis es igual a la fracción la raíz cuadrada de 19 al final de la raíz cuadrada i más 3, sobre 2 al final de la fracción.
$cos^2(x + 3) + sin^2(x)$	coseno cuadrado de equis más 3 más seno cuadrado de equis.	coseno al cuadrado paréntesis abierto equis más 3 paréntesis cerrados más seno al cuadrado de equis.
$cos^2(x) + 3 + sin^2(x)$		coseno al cuadrado de

		equis más 3 más seno al cuadrado de equis.

Como se muestra en la Tabla 3.3, se ha realizado el análisis de la representación de la expresión matemática con las diferentes velocidades definidas y se ha logrado obtener un correcto entendimiento con todas ellas. El permitir que los usuarios puedan elegir el grado de rapidez con el cual se sientan más cómodos, durante la reproducción del audio descriptivo, conllevó a que estos puedan aprovechar las diferentes pistas implementadas y se obtengan un nivel de entendimiento positivo.

Tabla 3.3 Análisis de velocidades de reproducción.

Expresión	Velocidad	Observaciones
$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$	Lenta	Se presentó una completa comprensión de los elementos que componen a la expresión matemática, así como fue posible identificar y diferenciar las diferentes pistas auditivas que representan los diferentes delimitadores definidos en la fase de Diseño. Además que permitió que los usuarios se familiaricen con las salidas generadas por el módulo.
	Normal	Se consiguió un completo entendimiento de la expresión, sin presentar ninguna inconformidad por parte del usuario.
	Rápida	La comprensión de los diferentes elementos e identificación de las pistas auditivas fue factible y presentó una mayor conformidad.

Como se presenta en la Tabla 3.4, se han comprobado los correctos cambios de volumen y velocidad (partiendo de un nivel de referencia) de acuerdo al tamaño del fragmento a leer (longitud de caracteres); y como se logra observar, se consigue que estas variaciones no afecten en la comprensión por parte del usuario y a su vez contribuyeron a que se generen archivos de audio más cortos y en fragmentos más extensos, el aumento del volumen ayude a que se mantenga el entendimiento de la voz sin importar que su velocidad aumentó con relación al nivel de referencia.

Tabla 3.4 Análisis de reproducción por fragmentos

Expresión	Fragmento	Longitud (caracteres)	Observaciones
$x = \frac{\sqrt{12231 - 32\pi}}{2759} + 22$	'equis es igual a la fracción',	28	Se presentó una reproducción del fragmento correspondiente a la velocidad reproducción elegida (lento, medio, rápido)
	' la raíz cuadrada',	17	
	' de 12231 menos 32 pi al final de la raíz cuadrada',	50	Se presentó un aumento del 20% en la velocidad y volumen audio sin afectar en la comprensión del usuario y consiguiendo que la reproducción del fragmento no sea de larga duración.
	', sobre 2759 al final de la fracción',	35	Se presentó un aumento del 10% en la velocidad y volumen audio sin afectar en la comprensión del usuario y consiguiendo que la reproducción del fragmento no sea de larga duración.
	' más 22'	7	Se presentó una reproducción del fragmento correspondiente a la

			velocidad reproducción elegida (lento, medio, rápido)

Como última parte de las pruebas, se procede a analizar la secuencia de reproducción del audio generado por la expresión matemática y, como se presenta en la Tabla 3.5, se muestra una correcta separación de los fragmentos de acuerdo a los delimitadores establecidos durante la fase de Diseño; además de presentarse una correcta inserción de la pista auditiva entre los fragmentos, obteniendo así una descripción detallada y guiada que ayude al usuario a tener un completo entendimiento de la expresión matemática procesada.

Tabla 3.5 Secuencia de reproducción de expresión matemática.

Expresión	
$x = \frac{\sqrt{19+i\alpha}+4^{\frac{\pi}{2}}}{4^{\frac{2}{i+6}}}$	equis es igual a la fracción la raíz cuadrada de 19 más i alfa final de raíz cuadrada más 4 a la pi sobre 2; , sobre 2 a la fracción 2, sobre i más 6 final de fracción; fin de fracción.
Fragmento/ Pista Auditiva	equis es igual a la fracción
	Tono de fracción
	la raíz cuadrada
	Tono de raíz
	de 19 más i alfa final de la raíz cuadrada
	Tono de raíz
	más 4 a la pi sobre 2; , sobre 2 a la fracción
	Tono de fracción
	2, sobre i más 6 final de fracción
	Tono de fracción
	fin de fracción
	Tono de fracción

Como se muestra en la Tabla 3.6, se ha validado el Requerimiento Funcional. Para una revisión profunda de los resultados generados por el **Módulo de Sonificación**, en el **Anexo VI** se adjunta un conjunto de expresiones matemáticas con sus respectivos archivos de audio.

Tabla 3.6 Validación de Requerimientos Funcionales.

Cod	Descripción	Estatus	Observación
RF-1	Se debe generar una salida de audio clara y descriptiva que permita al usuario poder comprender y reconocer toda la expresión matemática.	Validado ✅	Los usuarios presentaron conformidad con la claridad del audio reproducido, así como un óptimo nivel de satisfacción en la comprensión de la expresión matemática representada

3.3.2 REQUERIMIENTOS NO FUNCIONALES

Con la retroalimentación generada por los usuarios de prueba, como se presenta en la Tabla 3.7, se han comprobado y validado el cumplimiento de todos estos requerimientos No Funcionales.

Tabla 3.7 Validación de Requerimientos No Funcionales.

Cod	Descripción	Estatus	Observación
RNF-1	La velocidad de reproducción de la salida de audio generada debe contribuir a que el usuario no pierda la atención.	Validado ✅	Se trabajó con diferentes rangos de velocidad hasta determinar aquellos en los que no se generen audios tiempos de duración innecesariamente largos, así como no se generen audios con velocidades de reproducción tan rápidos que se pierda la comprensión de la descripción.
RNF-2	La salida de audio generada debe contar con un nivel de volumen dentro de los rangos que eviten afectaciones a la salud del usuario.	Validado ✅	Se probó con diferentes niveles de volumen tanto en la voz sintetizada como en las pistas auditivas, determinando un nivel de referencia en el cual no se presente disgusto por parte del usuario.
RNF-3	La activación del módulo de sonificación debe ser	Validado ✅	La implementación de atajos de teclado simples permitió que los usuarios se adapten fácilmente a la activación del módulo.

	fácil de invocar por parte del usuario.		
RNF-4	El sistema debe contar con una conexión a Internet.	Validado ✅	No se presentó problemas de conexión con la API de *Google Translator* la cual se encuentra en línea

3.4 FINALIZACIÓN DEL TABLERO DE KANBAN

Culminadas las pruebas de funcionamiento sobre el **Módulo de Sonificación**, corregidos todos los problemas encontrados en su ejecución y validados todos los requerimientos descritos en el **Apartado 2.1.3 DEFINICIÓN DE REQUERIMIENTOS**, como se muestra en la Figura 3.2, todas las actividades planificadas en el Trabajo de Integración Curricular han sido completadas.

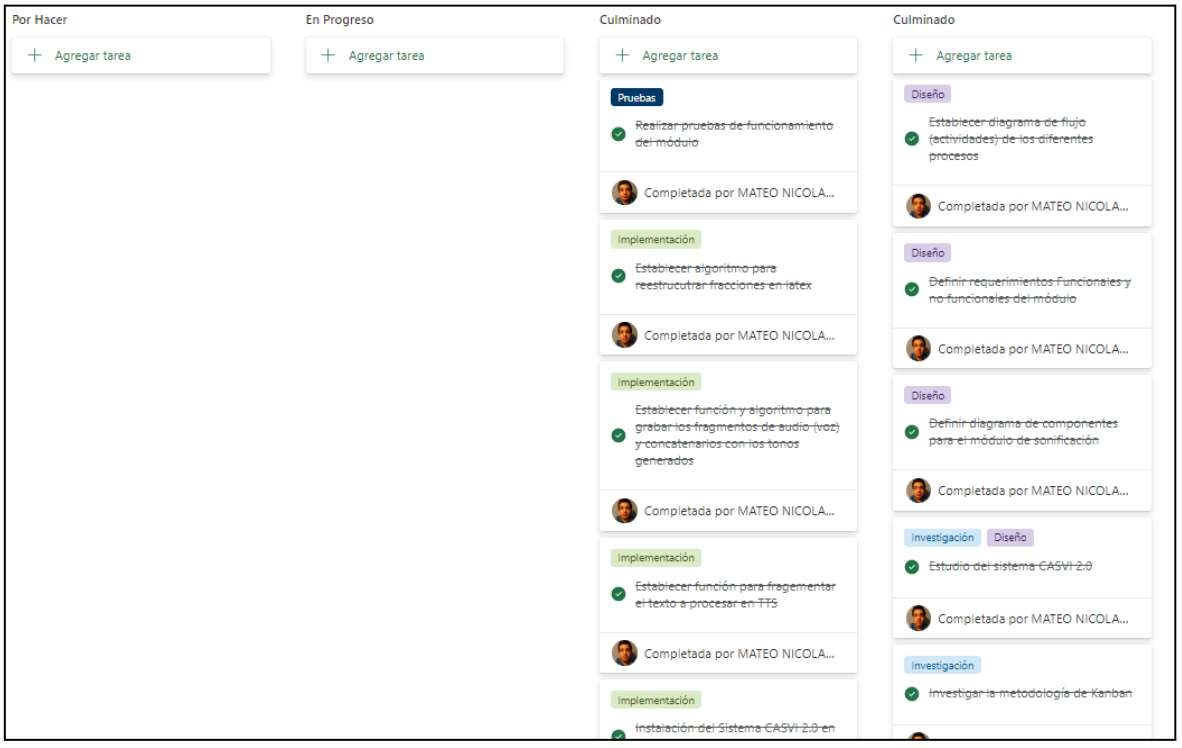


Fig 3.2 Finalización del Tablero de Kanban.

3.5 CONCLUSIONES Y RECOMENDACIONES

3.5.1 CONCLUSIONES

Este Trabajo de Integración Curricular ha sido desarrollado a través de una serie de fases, siendo estas: Teórica, Diseño, Implementación y Pruebas de Funcionamiento; de las cuales se han obtenido las siguientes conclusiones:

- La ausencia de un **Módulo de Sonificación** que describa la expresión matemática y guíe al usuario no vidente dentro de esta, limitaba la comprensión dentro del sistema web matemático CASVI 2.0, pues solo contaba con una con una lectura lineal que no permitía una completa comprensión de los diferentes elementos que comprenden la salida de la expresión matemática así como los límites de estos; generando ambigüedad y confusión en el usuario.

- La implementación de un módulo de sonorización sobre el sistema web matemático CASVI 2.0 ha permitido la generación de una descripción auditiva de las salidas obtenidas del procesamiento de una ecuación, con lo que se ha conseguido que usuarios con discapacidad visual, mediante las integración de pistas tanto léxicas como de sonido, tengan la oportunidad de tener un mayor entendimiento sobre la expresión matemática obtenida, facilitando así su entendimiento .

- La implementación de la metodología de Kanban ha permitido una mejor organización y priorización de todas las tareas que involucraron el desarrollo del Trabajo de Integración Curricular, así como, ha contribuido a un mejor control de calidad y transparencia en cada avance presentado; consiguiendo así la integración de un módulo eficiente dentro del Sistema CASVI 2.0

- El diseño de un flujo de trabajo para el proceso de sonificación de las expresiones matemáticas dentro del Sistema CASVI 2.0 es fundamental en el desarrollo de un óptimo **Módulo de Sonificación**.

- Las distintas funciones desarrolladas dentro del **Módulo de Sonificación**, han sido el resultado de la correcta aplicación de las diversas herramientas, conceptos y tecnologías investigadas durante la fase Teórica y de Investigación, consiguiendo así un módulo óptimo y consistente con lo planteado durante la fase de Diseño.

- El uso de pistas léxicas y auditivas para la descripción de la expresión de la matemática han permitido establecer un eficiente **Módulo de Sonificación**, que funcione como apoyo del usuario no vidente para la correcta interpretación y comprensión de las salidas obtenidas..

- La ejecución de pruebas de funcionamiento en escenarios controlados son una pieza muy importante en el desarrollo de un proyecto. El análisis de sus resultados posibilita tener una constante retroalimentación que permita la temprana detección y corrección de problemas o eventualidades que afecten ya sea la funcionalidad del módulo o la satisfacción del usuario final (no vidente).

3.5.2 RECOMENDACIONES

Durante el desarrollo del presente Trabajo de Integración Curricular, se han presentado un conjunto de inconvenientes cuya inmediata solución ha sido necesaria, con lo cual en base a las situaciones dadas se establecieron las siguientes recomendaciones:

- El uso de APIs *online* dentro del procesamiento de los diferentes elementos del módulo en el lado del Servidor pueden generar muchos retardos que son percibidos por los usuarios. Es por esto que se recomienda trabajar en todo lo posible con librerías de código locales.

- Los eventos relacionados con la generación del audio por parte del Servidor son más fáciles de programar mediante *Python*, puesto que este lenguaje es muy rico en herramientas relacionadas con el procesamiento de datos, así como en motores de audio, haciendo posible que el desarrollo del módulo no presente un nivel de dificultad innecesario.

- Es una buena práctica de desarrollo, el siempre probar primero las diferentes funcionalidades del módulo en un ambiente aislado del sistema, permitiendo identificar y depurar cualquier falla propia del módulo. Una vez que se haya comprobado esto, se puede proceder a realizar pruebas del módulo integrado en el sistema, para así detectar algún problema durante la comunicación con este.

- Probar la sonificación de la expresión matemática con diferentes configuraciones de audio y diferentes parámetros de sonido en las pistas auditivas, de manera que se puedan generar varias opciones de las cuales se pueda escoger la más óptima y agradable para el usuario.

3.5.3 TRABAJO FUTURO

Con la experiencia conseguida en todo el proceso de desarrollo del Trabajo de Integración Curricular se han presentado posibles consideraciones para un trabajo futuro el cual mejore o complemente al **Módulo de Sonificación:**

- Se podría definir un método que permita sonificar de manera individual los diferentes fragmentos que componen a la expresión matemática, de manera que el usuario pueda desplazarse entre los diferentes componentes y genere eventos que permitan reproducirlos. Es decir que si el usuario desea explorar a detalle solo una parte de la expresión no tenga que volver a escuchar todo el audio generado y más bien solo presente el fragmento de audio correspondiente al componente de la expresión en la cual el usuario se encuentra posicionado.

- Se podría definir una serie de métodos que permitan personalizar la experiencia del usuario, es decir que este tenga la oportunidad de modificar diferentes parámetros del audio como tono, amplitud, volumen, timbre, entre otros; de manera que el usuario pueda establecer a su gusto como quiere que la expresión matemática sea sonorizada.

4. REFERENCIAS BIBLIOGRÁFICAS

[1] K. Lee, *Python Programming Fundamentals*, 2ª ed. Decorah: Springer, 2014

[2] JavaScript | MDN . Accedido el 6 de junio de 2022. *https://developer.mozilla.org/es/docs/Web/JavaScript*.

[3] S. Mianji Johnsson, *The building of the webpages : The comparison study of MERN and MEVN* , Dissertation, 2020.

[4] M. Bermejo, El Kanban. España: Universitat Oberta de Catalunya, 2012.

[5] Mejía, Paúl, Luiz César Martini, Felipe Grijalva, y Ana María Zambrano. CASVI: Computer Algebra System Aimed at Visually Impaired People. Experiments . *IEEE Access* 9 (2021): 157021-34. *https://doi.org/10.1109/ACCESS.2021.3129106.*

[6] Menzi-Çetin, Nihal, Ecenaz Alemdağ, Hakan Tüzün, y Merve Yıldız. *Evaluation of a University Website's Usability for Visually Impaired Students . Universal Access in the Information Society* 16, n.º 1 (1 de marzo de 2017): 151-60. *https://doi.org/10.1007/s10209-015-0430-3.*

[7] A. Bhowmick y S. M. Hazarika, "*An insight into assistive technology for the visually impaired and blind people: state-of-the-art and future trends*", *Journal on Multimodal User Interfaces*, vol. 11, n.º 2, pp. 149–172, enero de 2017. Accedido el 30 de noviembre de 2022. *https://doi.org/10.1007/s12193-016-0235-6*

[8] Mejía, Paúl, Luiz César Martini, Felipe Grijalva, Julio César Larco, y Juan Carlos Rodríguez. *A Survey on Mathematical Software Tools for Visually Impaired Persons: A Practical Perspective* . *IEEE Access* 9 (2021): 66929-47. *https://doi.org/10.1109/ACCESS.2021.3076306.*

[9] Consuegra, Wilfredo Martínez, Raúl González Peña, y Mario Yuniesky Escobar Raimundo. PARTICULARIDADES DEL DISEÑO, DESARROLLO Y EVALUACIÓN DE SOFTWARE EDUCATIVO ACCESIBLE PARA PERSONAS CIEGAS . *Revista Varela* 7, n.º 18 (1 de septiembre de 2007): 1-9.

[10] Alabi, Adefunke O., y Stephen M. Mutula. *Digital inclusion for visually impaired students through assistive technologies in academic libraries* . *Library Hi Tech News* 37, n.º 2 (1 de enero de 2020): *14-17. https://doi.org/10.1108/LHTN-11-2019-0081.*

[11] Silveira, Barbara Cristina A., Thiago Silva-de-Souza, y Ana Regina C. da Rocha. *Software Accessibility for Visually Impaired People: a systematic mapping study* . En *Proceedings of the 17th Brazilian Symposium on Software Quality*, 190-99. SBQS. New York, NY, USA: Association for Computing Machinery, 2018. *https://doi.org/10.1145/3275245.3275266.*

[12] Bhardwaj, Raj Kumar, y Sanjay Kumar. *A comprehensive digital environment for visually impaired students: user's perspectives* . *Library Hi Tech* 35, n.º 4 (1 de enero de 2017): 542-57. *https://doi.org/10.1108/LHT-01-2017-0016.*

[13] *"Web Content Accessibility Guidelines (WCAG) 2.2"*. World Wide Web Consortium (W3C). https://www.w3.org/TR/WCAG22/ (accedido el 15 de septiembre de 2022).

[14] Ortega-Santin, Yadira Nathalie. Diseño y planificación para el desarrollo de un software web orientado a la enseñanza de la geometría para estudiantes no videntes , 2018. *https://reunir.unir.net/handle/123456789/7433.*

[15] *User Agent Accessibility Guidelines (UAAG) 2.0* . Accedido 10 de octubre de 2022. *https://www.w3.org/TR/UAAG20/#conceptual_overview.*

[16] Flores Rodriguez, Hector Enrique, y Jaime Hernan Sanchez Ilabaca. *APRENDIZAJE MOVIL DE CIENCIAS PARA CIEGOS*. LOM EDICIONES S.A, 2006. *http://repositorio.conicyt.cl/handle/10533/166071.*

[17] Ferreyra, José Alberto, Amalia Méndez, y María Rodrigo. El uso de las TIC en la Educación Especial: descripción de un sistema informático para niños discapacitados visuales en etapa preescolar . *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, n.º 3 (2 de enero de 2009): 55-62. *https://doi.org/10.24215/18509959.0.p.*

[18] Sánchez, Jaime. Aprendizaje de Ciencias a través de Audio en Niños Ciegos , 1-21, 2006.

[19] Orea, Alfonso Sánchez, Alma Rosa García Gaona, María Dolores Vargas Cerdán, José Rafael Rojano Cáceres, y Francisco Javier Álvarez Rodríguez. Asistente móvil basado en audio para la lectura de textos como apoyo a personas con discapacidad visual .

Tecnología Educativa Revista CONAIC 2, n.º 3 (2015): 43-47. *https://doi.org/10.32671/terc.v2i3.152.*

[20] Nazemi, Azadeh, Iain Murray, y Nazanin Mohammadi. Mathspeak: *An Audio Method for Presenting Mathematical Formulae to Blind Students* . En *2012 5th International Conference on Human System Interactions*, 48-52, 2012. *https://doi.org/10.1109/HSI.2012.17.*

[21] *Expanding Audio Access to Mathematics Expressions by Students With Visual Impairments via MathML* - Frankel - 2017 - ETS Research Report Series - Wiley Online Library . Accedido 10 de octubre de 2022. *https://onlinelibrary.wiley.com/doi/full/10.1002/ets2.12132.*

[22] Sepulveda, Jose Sebastian Fuentes. Accesibilidad en expresiones matemáticas , s. f., 56. *http://www.inf.udec.cl/~jfuentess/files/pdf/MemoriaTitulo.pdf*

[23] Aptus.org. Teoría de la carga cognitiva, un área de investigación que los profesores necesitan comprender . Accedido 11 de octubre de 2022. *https://www.aptus.org/publicacion/articulo-teoria-de-la-carga-cognitiva-un-area-de-investigacion-que-los-profesores-necesitan-comprender/.*

[24] Asinsten, Juan Carlos. *El sonido Edición de sonido en computadora, para proyectos en Clic, multimedia y otras actividades educativas*. Argentina: Educ.ar, s. f. *https://www.educ.ar/recursos/91094/el-sonido/download/inline.*

[25] Sundberg, J. Acoustic and psychoacoustic aspects of vocal vibrato . *STL-QSPR* 35, n.º 2-3 (1994): 045-068.

[26] Ahmad, Muhammad Ovais, Jouni Markkula, y Markku Oivo. *Kanban in software development: A systematic literature review* . En *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, 9-16, 2013. *https://doi.org/10.1109/SEAA.2013.28.*

[27] Mascolo, Maria Di, Yannick Frein, y Yves Dallery. *An Analytical Method for Performance Evaluation of Kanban Controlled Production Systems* . *Operations Research* 44, n.º 1 (febrero de 1996): 50-64. *https://doi.org/10.1287/opre.44.1.50.*

[28] Lei, Howard, Farnaz Ganjeizadeh, Pradeep Kumar Jayachandran, y Pinar Ozcan. *A Statistical Analysis of the Effects of Scrum and Kanban on Software Development Projects* . *Robotics and Computer-Integrated Manufacturing*, Special Issue: Extended Papers Selected from FAIM 2014, 43 (1 de febrero de 2017): 59-67. *https://doi.org/10.1016/j.rcim.2015.12.001.*

[29] Corona, Erik, y Filippo Eros Pani. *A Review of Lean-Kanban Approaches in the Software Developmen*t . *1* 10 (enero de 2013). *http://www.wseas.us/journal/pdf/information/2013/5709-110.pdf.*

[30] Raut, Laukik, Rajat Wakode, y Pravin Talmale. *Overview on Kanban Methodology and its Implementation* . *International Journal for Scientific Research & Development* 03 (1 de julio de 2015): 2518-21.

[31] *Amazon Web Services, Inc. What Is Python? - Cloud Beginner's Guide to Python - AWS* . Accedido 6 de noviembre de 2022. *https://aws.amazon.com/what-is/python/.*

[32] *Node.js. About Node.js* . Node.js. Accedido 6 de noviembre de 2022. *https://nodejs.org/en/about/.*

[33] Maxima, un sistema de álgebra computacional . Accedido 6 de noviembre de 2022. *https://maxima.sourceforge.io/es/.*

[34] MongoDB. ¿Qué Es MongoDB? Accedido 6 de noviembre de 2022. *https://www.mongodb.com/es/what-is-mongodb*.

[35] *Introduction | Vue.js* . Accedido 6 de noviembre de 2022. *https://vuejs.org/guide/introduction.html.*

[36] *The LaTeX Project Introduction to LaTeX* . Accedido 6 de noviembre de 2022. *https://www.latex-project.org/about/.*

[37] *MathML | MDN* . *MDN Web Docs* Accedido 6 de noviembre de 2022. *https://developer.mozilla.org/es/docs/Web/MathML.*

[38] *The University of Edinburgh. SnuggleTeX - Overview & Features* . Accedido 6 de noviembre de 2022. *https://www2.ph.ed.ac.uk/snuggletex/documentation/overview-and-features.html.*

[39] *Web Speech API - Referencia de la API Web | MDN* . Accedido 7 de noviembre de 2022. *https://developer.mozilla.org/es/docs/Web/API/Web_Speech_API.*

[40] Pallets. *Flask* . Accedido 7 de noviembre de 2022. *https://palletsprojects.com/p/flask/.*

[41] *Apache Maven Project. Maven – Introduction* . Accedido 7 de noviembre de 2022. *https://maven.apache.org/what-is-maven.html.*

[42] Solución Kanban de administración de tareas para Teams | Microsoft Planner . Accedido 19 de octubre de 2022. *https://www.microsoft.com/es-ww/microsoft-365/business/task-management-software.*

[43] Bhat, Natesh M. *Pyttsx3 Documentation* , 2017., 21.

[44] Gourdol A. *CortexJS* . Accedido 21 de noviembre de 2022. *https://cortexjs.io/mathlive/.*

[45] *Googletrans: Free and Unlimited Google translate API for Python — Googletrans 3.0.0 documentation* . Accedido 21 de noviembre de 2022. *https://py-googletrans.readthedocs.io/en/latest/.*

[46] *re — Regular expression operations — Python 3.11.0 documentation* . Accedido 21 de noviembre de 2022. *https://docs.python.org/3/library/re.html.*

[47] os — Interfaces misceláneas del sistema operativo — documentación de Python - 3.10.8. Accedido 21 de noviembre de 2022. *https://docs.python.org/es/3.10/library/os.html.*

[48] *time — Time access and conversions — Python 3.11.0 documentation* . Accedido 21 de noviembre de 2022. *https://docs.python.org/3/library/time.html.*

[49] Robert, James. *Pydub*. Python, 2022. *https://github.com/jiaaro/pydub.*

[50] *What is NumPy? — NumPy v1.23 Manual* . Accedido 21 de noviembre de 2022. *https://numpy.org/doc/stable/user/whatisnumpy.html.*

[51] Barraza, Paula, Douglas J. Gillan, Arthur Karshmer, y Skye Pazuchanics. «A Cognitive Analysis of Equation Reading Applied to the Development of Assistive Technology for Visually-Impaired Students». *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 48, n.º 5 (1 de septiembre de 2004): 922-26. *https://doi.org/10.1177/154193120404800538.*

[52] Kondak, Zachary. «Web Sonification Sandbox - an Easy-to-Use Web Application for Sonifying Data and Equations», 1 de enero de 2017. https://www.academia.edu/67782406/Web_Sonification_Sandbox_an_Easy_to_Use_Web_Application_for_Sonifying_Data_and_Equations.

5. ANEXOS

Los anexos adjuntados como recursos complementarios al desarrollo del actual Trabajo de Integración Curricular se presentan a continuación:

ANEXO I: Trabajo de Integración Curricular CASVI 2.0.

ANEXO II: Casos de Uso.

ANEXO III: Manual de Instalación CASVI 2.0.

ANEXO IV: Códigos del Módulo de Sonificación.

ANEXO V: Encuestas realizadas a usuarios de pruebas.

ANEXO VI: Expresiones matemáticas y sus archivos de audio.