

**HERRAMIENTA SOFTWARE PARA EL ENTRENAMIENTO DE LA LECTURA
MUSICAL DE UN INSTRUMENTO DE PERCUSIÓN**

DIANA PATRICIA ORTIZ LIZCANO



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2012

**HERRAMIENTA SOFTWARE PARA EL ENTRENAMIENTO DE LA LECTURA
MUSICAL DE UN INSTRUMENTO DE PERCUSIÓN**

DIANA PATRICIA ORTIZ LIZCANO

**Trabajo de grado presentado como requisito para optar al título de Ingeniero
de Sistemas**

Director:

FERNANDO ANTONIO ROJAS MORALES



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2012

*A Dios, primero que todo,
por ser el motor de mi vida y
mi fortaleza en momentos de debilidad.*

*A mi madre Nelly y a mi padre Heriberto por creer en mí,
por ser mi apoyo incondicional en todo momento brindándome
su cariño y motivándome día a día a salir adelante.*

*A mi hermana Sandra, por ser mi ejemplo a seguir guiándome
con sus consejos y enseñanzas durante este tiempo.*

*A Juan Sebastián por estar pendiente de mí
dándome su apoyo, cariño y ánimo a lo largo del proyecto.*

*A mis amigos con los que he compartido grandiosos momentos, aquellos
que confiaron en mí y me apoyaron desde el comienzo, en especial: Jessica, Paula,
Néstor, Ferney, Julián, Juan David y John Alexander.*

A cada uno gracias por todo.

AGRADECIMIENTOS

La autora del proyecto ofrece sus más sinceros agradecimientos a:

El profesor **Fernando Antonio Rojas Morales**, director del proyecto, porque se arriesgó a proponerme su idea y confió en mis capacidades para la realización de la misma. A lo largo del proyecto me brindó su apoyo y conocimiento permitiéndome alcanzar todos los objetivos propuestos.

La **UNIVERSIDAD INDUSTRIAL DE SANTANDER** por brindarme los medios necesarios para alcanzar una meta tan importante como la de convertirme en profesional

Y finalmente a la **Escuela de Ingeniería de Sistemas** por acogerme y dotarme de conocimiento y sabiduría durante estos años de formación como ser humano íntegro dispuesto a servirle a la sociedad.

CONTENIDO

	Pág.
INTRODUCCIÓN	14
JUSTIFICACIÓN	16
OBJETIVOS	18
OBJETIVO GENERAL	18
OBJETIVOS ESPECÍFICOS	18
1. FUNDAMENTACIÓN TEÓRICA	19
1.1 SOPORTE TEÓRICO	19
1.1.1 Software educativo.	19
1.1.2 Tipologías de Software Educativo.	20
1.1.3 Clasificación de programas didácticos.	21
1.1.4 Funciones del Software Educativo.	22
1.1.5 Ingeniería de software educativo.	23
1.1.6 Software educativo como medio de enseñanza.	23
1.1.7 Software educativo como estrategia educativa	25
1.1.8 La música por computadora.	27
1.1.9 Computadoras aplicadas a la música.	28
1.1.10 Entorno MIDI.	29
1.1.11 El problema de la interpretación.	30
1.1.12 La notación musical.	31
1.1.13 La composición por computadora.	31
1.1.14 La música por computadora en la educación.	32
1.2 SOPORTE TECNOLÓGICO	33
1.2.1 Netbeans.	33
1.2.2 Java.	34
2. METODOLOGÍA DE DESARROLLO DEL PROYECTO	37

CONCLUSIONES	87
RECOMENDACIONES	88
BIBLIOGRAFÍA	89
ANEXOS	90

LISTA DE FIGURAS

	Pág.
Figura 1. Interacción de usuarios 1	24
Figura 2. Interacción de usuarios 2	27
Figura 3. Música por computador	28
Figura 4. Metodología de ISE	38
Figura 5. Diagrama de casos de uso	47
Figura 6. Diagrama de clases	58
Figura 7. Diagrama de navegación de ventanas:	59
Figura 8. Primer prototipo:	64
Figura 9. Segundo prototipo:	65
Figura 10. Tercer prototipo:	66
Figura 11. Cuarto prototipo:	66
Figura 12. Quinto prototipo:	67
Figura 13. Funcionamiento de núcleo del sistema	71
Figura 14. Generación de sonidos	71
Figura 15. Funcionamiento operaciones musicales	72
Figura 16. Características de usabilidad	77
Figura 17. Análisis modo ritmo	79
Figura 18. Análisis modo compositor	80
Figura 19. Pregunta No. 1	81
Figura 20. Pregunta No. 2	82
Figura 21. Pregunta No. 3	83
Figura 22. Pregunta No. 4	83
Figura 23. Pregunta No. 5	84
Figura 24. Pregunta No. 6	85
Figura 25. Pregunta No. 7	85

LISTA DE TABLAS

	Pág.
Tabla 1. Características de los programas educativos	20
Tabla 2. Tipologías de Software Educativo	21
Tabla 3. Funciones del Software Educativo	22
Tabla 4. Tabla de Eventos:	46
Tabla 5. Especificación de los casos de usos:	48
Tabla 6. En ritmo:	74
Tabla 7. En compositor:	75
Tabla 8. Modo ritmo:	79
Tabla 9. Modo componer:	80
Tabla 10 Observaciones de la prueba piloto	86

LISTA DE ANEXOS

	Pág.
ANEXO A. Glosario de términos	90
ANEXO B. Descripción por pantallas de la aplicación	91
ANEXO C. Diagrama de clases generado por NETBEANS	96

RESUMEN

TÍTULO:

HERRAMIENTA SOFTWARE PARA EL ENTRENAMIENTO DE LA LECTURA MUSICAL DE UN INSTRUMENTO DE PERCUSIÓN.

AUTOR: Diana Patricia Ortiz Lizcano.**

PALABRAS CLAVES: batería, tablatura, software, usabilidad, ritmos, tecnología.

DESCRIPCIÓN

Los avances tecnológicos han sido un gran apoyo para la generación de sonidos musicales por medio de computadoras convirtiéndola en una tendencia altamente aceptada para lograr el proceso de enseñanza-aprendizaje en la lectura y escritura de partituras musicales. Estos avances han hecho más accesible la producción y ejecución de música debido a la facilidad de adquisición de equipos sistemáticos que incentivan sobre todo a la población infantil a desarrollar destrezas mentales desde edades tempranas.

Este documento muestra cómo se construyó una aplicación software como apoyo didáctico para la asimilación de conceptos musicales, ilustrando el proceso de desarrollo de software educativo para el entrenamiento de la lectura de un instrumento de percusión. Para lograrlo se muestra de manera ilustrativa la forma en la que se crean y reproducen ritmos a partir de colocar las notas en una tablatura semejando la realidad. También se describe como por medio de este trabajo de investigación de la mano de la ingeniería del software educativo, bajo la orientación del director del proyecto y guiados por un estudiante de música se logró obtener un producto estable con altos niveles de usabilidad y contenido educativo al que se le realizó una prueba piloto en la que se midió el impacto del uso de la herramienta con usuarios finales alcanzando resultados satisfactorios.

* Proyecto de Grado.

** Facultad de ingenierías físico-mecánicas. Escuela de ingeniería de sistemas e informática. Directora de Proyecto: Fernando Antonio Rojas Morales

SUMMARY

TITLE:

SOFTWARE TOOL FOR MUSICAL READING TRAINING OF A PERCUSSION INSTRUMENT.*

AUTHORS: Diana Patricia Ortiz Lizcano**

KEY WORDS: drums, tablature, software, usability, rhythms, technology.

ABSTRACT

Technological developments have been very supportive for the creation of music sounds by using computers, converting them into a new trend utterly accepted to achieve the teaching-learning process in the composition and comprehension of music tablatures. These improvements have become the production and execution of the music processes much more accessible, through the acquisition of new technological equipment, which encourages specially the younger population to improve mental skills, starting at early years.

This document states how a software application was built as a didactical support for the assimilation of musical concepts, applied to a particular percussion instrument, the drums, illustrating the development process of an educational software used to the training of musical lecture understanding. In order to be successful, the rhythms and notes produced are created and displayed on a screen tablature. This document also describes as through this research with the assistance of software education engineering, oriented by the project director and guided by a music student is able to obtain a stable product with high levels of usability and educational content. This product was tested using a pilot test that measured the tool impact in the software use with end users. This pilot test had great results.

* Draft Grade.

** Faculty physical-mechanical engineering. School systems engineering and information technology. Project Director: Fernando Antonio Morales Rojas

INTRODUCCIÓN

En este trabajo se plantea una metodología de análisis, diseño, desarrollo e implementación para software educativo. Esta metodología intenta buscar la asociación entre la ingeniería de software y el aprendizaje surgiendo de estos dos campos la creación de un software educativo.

En el desarrollo de este proyecto se utilizó la metodología de Ingeniería de Software Educativo (ISE) la cual permite detectar situaciones problemáticas, sus posibles causas y las alternativas de solución teniendo como base, teorías de aprendizaje que ayudan a elaborar productos con fines pedagógicos.

Además de tener un enfoque pedagógico con objetivos educativos, este trabajo esta guiado por pautas de calidad de software. Los principales intereses del proyecto son dos: el primero de ellos es el interés pedagógico, porque se pretende que mediante el uso de esta herramienta se adquiera un conocimiento básico de composición musical a través de la estrategia que se emplee, y el segundo es el interés social debido a que este trabajo de investigación espera convertirse en un aporte más para el área de entrenamiento musical.

En la actualidad se encuentran diversos aplicativos relacionados con la música, algunos de ellos complejos y otros más sencillos orientados únicamente al entretenimiento. Es por esto que surgió la necesidad de crear una herramienta guiada por una metodología disciplinada basada en criterios de evaluación de productos software educativos. Combat¹ se clasifica como un juego educativo que busca dar soporte al proceso de enseñanza-aprendizaje materializando teorías musicales existentes en un producto informático que simula el funcionamiento de una batería de forma didáctica e interactiva. El núcleo del proyecto es el API de

¹ Abreviatura usada para mencionar el proyecto: combat – compositor de batería

Java encargado de reproducir los sonidos de los instrumentos de percusión a determinada frecuencia y velocidad. El usuario del sistema tiene la posibilidad de crear tablaturas² de batería en una interfaz sencilla y amigable sin tener en cuenta el grado de conocimiento musical, simplemente basado en la creatividad y el talento innato de los niños.

La línea de desarrollo que siguió este proyecto es la que corresponde a productos que nacen con la enseñanza asistida por computadora, estos productos se clasifican dentro del software educativo y se pueden trabajar de dos formas: con acompañamiento o sin acompañamiento de un tutor. Para este proyecto se contemplan las dos posibilidades, lo importante es que el usuario orientado por un tutor o de forma individual logre realizar los ejercicios básicos que la herramienta ofrece.

La regla principal en la construcción de este software fue la usabilidad, teniendo en cuenta que la población para la que se desarrolló son niños, y que para ellos es muy importante contar con una interfaz gráfica atractiva e intuitiva que mantenga estándares de los entornos de desarrollo que utilizan constantemente.

Para comprobar si la aplicación cumple con la funcionalidad requerida se realizó una prueba piloto en una población de niños de diferentes edades, midiendo el impacto cognoscitivo que la herramienta genera en ellos. Este tipo de pruebas permiten evaluar la eficacia del software, su eficiencia, y el desempeño de los usuarios.

² Término que se utiliza para definir formas de escritura musical especiales para ciertos instrumentos. Para mayor información revisar <http://es.wikipedia.org/wiki/Tablatura>

JUSTIFICACIÓN

En el ámbito musical los campos de apoyo que el computador ofrece son múltiples. Es por esto que se considera al computador como un instrumento de apoyo que le facilita a los seres humanos la forma de elaborar una actividad.

El computador tiene un atractivo muy importante y es la facultad de reproducir sonidos. En el software elaborado para este proyecto el usuario tiene la posibilidad de componer y reproducir ritmos, de tal forma que se establezca una relación entre lo que hizo con la realidad. Después de que se ha creado algún ritmo o tema el usuario lo puede escuchar varias veces, y hacer correcciones si lo considera necesario. Además de esto puede guardar su composición y manipularla de la forma que desee construyendo temas a partir de ritmos creados con anterioridad.

Pese a la existencia de numerosas aplicaciones orientadas a la música, la aplicación aquí descrita tiene fines educativos que combinados con didáctica resultan siendo también entretenidos. No obstante, existen algunos programas con un enfoque específico hacia el aprendizaje de la música, algunos comprenden el desarrollo auditivo, contrapunto básico y otros hasta composición elemental, pero la gran diferencia radica en que ninguno de ellos cumple con los requerimientos que debería tener un software dirigido a la población infantil mostrándole de forma didáctica contenidos musicales que contribuyan al entendimiento de conceptos básicos que se van adquiriendo con el uso constante de la aplicación.

Más específicamente, lo que se esperaba lograr era motivar a los niños a entrenarse en la lectura musical de un instrumento de percusión creando un software que los ejercitara de forma pedagógica y ante todo permitiera la interacción del niño con la máquina por medio de una interfaz sencilla y práctica

de manejar con las herramientas ubicadas de forma visible. La tablatura se diseñó pensando en que la población principal del software es infantil en promedio, con pocos o muy bajos conocimientos en lectura musical; así que se realizó muy sencilla pero sin perder su objetivo principal que es enseñar a leer el ritmo de una canción por medio de las figuras musicales como si estuvieran leyendo cualquier texto.

El programa ofrece varios servicios, entre ellos la selección y cancelación de instrumentos, la reproducción sonora de lo que se ha escrito en la tablatura, herramientas usuales como cortar, pegar, guardar, limpiar, reproducir, pausar, parar, tempo y volumen. Permite la entrada de datos, en este caso notas musicales a través de cada uno de los instrumentos, esto se hace por medio de la MIDI (Musical instruments digital interface o Interface Digital para Instrumentos Musicales. Es una norma que surge en el año 1983 debido a la necesidad de unificar criterios sobre la construcción de instrumentos musicales electrónicos con tecnología digital)³ que se encarga de reproducir lo que le corresponde a cada instrumento en tiempo real, ya sea por fragmentos o secuencias completas. Para el usuario representa una herramienta muy útil porque se logra la realización de una tablatura a partir de la ejecución musical de los instrumentos.

Por otro lado, el programa interpreta adecuadamente lo que el usuario realiza, controla errores que se puedan presentar al intentar colocar notas que excedan el tiempo de duración para un compás 4/4. La reproducción final es exacta, permitiendo que se hagan ajustes de parámetros como velocidad de reproducción, variación de la intensidad con la que suena, entre otros.

³ Referencia tomada de prodmusical.unsl.edu.ar/apuntes/Norma%20MIDI.pdf

OBJETIVOS

OBJETIVO GENERAL

Desarrollar una herramienta de software que provea un medio para entrenar niños en la lectura musical de un instrumento de percusión.

OBJETIVOS ESPECÍFICOS

- Definir en forma detallada los requerimientos del software educativo por medio de un modelo de casos de uso.
- Diseñar un software con una interfaz comunicacional agradable en la cual el usuario interactúe con la máquina de forma sencilla.
- Emplear un lenguaje de programación útil para desarrollar aplicaciones que permita incorporar contenidos multimedia.
- Realizar una prueba piloto con el fin de medir el impacto del uso de la herramienta musical a niños diferentes edades.

1. FUNDAMENTACIÓN TEÓRICA

La información que aparece en esta fase del proyecto se recopiló por medio de algunos sitios web y referencias bibliográficas. De acuerdo a lo que se encontró se clasificó aquella información útil durante el desarrollo del proyecto para ser mencionada.

1.1 SOPORTE TEÓRICO

1.1.1 Software educativo. El software educativo, consiste en la elaboración de programas que permiten cumplir o apoyar labores educativas, aquí se encuentran tanto los que apoyan la administración de procesos educacionales o de investigación, como los que dan soporte al proceso mismo de enseñanza-aprendizaje.

Es por esto que el software educativo también se denomina Material Educativo Computarizado (MEC), el cual es un ambiente informático que se encarga de que el usuario para el que se preparó, viva el tipo de experiencias educativas que se consideran importantes frente a una necesidad educativa dada. La calidad del MEC depende de lo que se espera de él, ubicándolo en el contexto en el que se dan las necesidades, así como los recursos y limitaciones.

Tabla 1. Características de los programas educativos

Características	Descripción
Facilidad de uso	En lo posible autoexplicativos y con ayudas.
Capacidad de motivación	Mantener el interés de los alumnos
Relevancia curricular	Relacionados con las necesidades del docente
Versatilidad	Adaptables al recurso informático disponible
Enfoque pedagógico	Que sea actual: constructivista o cognitivista
Orientación hacia los alumnos	Con control del contenido del aprendizaje
Evaluación	Incluirán módulos de evaluación y seguimiento

En la Tabla No. 1 se observan algunas características principales de los programas educativos. Se da por entendido que dichos programas deben servir de recurso para incentivar procesos de enseñanza y aprendizaje con características particulares de otros materiales educativos y con uso intensivo de los recursos informáticos de los que se dispone.

1.1.2 Tipologías de Software Educativo. Los programas educativos se pueden clasificar según su tipo. En la Tabla No. 2 se muestran algunos de acuerdo a diferentes criterios. Un aspecto clave de todo buen diseño son las características de la interface de comunicación. Dicha interface deberá estar de acuerdo con la teoría comunicacional aplicada y con las diferentes estrategias para el desarrollo de determinados procesos mentales. Por otra parte, cuando el software se desarrolla a partir de un lenguaje de programación, ya sea estructurado u orientado a objetos, se tiene que considerar que se fundamenta en algún algoritmo que lo soporta, cuyo diseño deberá reunir algunas características esenciales como la modularidad. Gran parte de los programas educativos pertenecen a un sub-grupo denominado hipermediales⁴, y en ellos las bases de datos de imágenes fijas o en movimiento, vídeo clips y sonidos juegan un papel fundamental a la hora de diseñar el programa.

⁴ Resultado de la combinación de hipertexto y multimedia, es decir no incluye solo texto sino también imágenes, audio, etc. Referencia tomada de <http://www.hipertexto.info/documentos/hipermedia.htm>

Tabla 2. Tipologías de Software Educativo

Tipologías según:		
Los contenidos	Temas, áreas curriculares	
Los destinatarios	Por niveles educativos, edad, conocimientos previos	
Su estructura	Tutorial, base de datos, simulador constructor, herramienta	
Sus bases de datos	Cerrados o abiertos	
Los medios que integra	Convencional hipermedia, realidad virtual	
Los objetivos educativos que pretende facilitar	Conceptuales, actitudinales, procedimentales	
Los procesos cognitivos que activa	Observación, identificación, construcción memorización, clasificación, análisis, síntesis, deducción, valoración, expresión, creación, etc.	
El tipo de interacción que propicia	Recognitiva, reconstructiva, intuitiva, constructiva	(Kemmis, 1970)
Su función en el aprendizaje	Instructivo, revelador, conjetural.	
Su comportamiento	Tutor, herramienta, aprendiz	Taylor(1990)
Sus bases psicopedagógicas sobre el aprendizaje	Conductista, constructivista, cognitivista	(Gros Begoña, 1997)
Su función en la estrategia didáctica	Informar, motivar, orientar, ayudar, proveer recursos, facilitar prácticas, evaluar	
Su diseño	Centrado en el aprendizaje, centrado en la enseñanza, proveedor de recursos	

1.1.3 Clasificación de programas didácticos. Una clasificación factible de los programas puede ser: tutoriales, simuladores, entornos de programación y herramientas de autor.

- Tutoriales: son programas que dirigen el aprendizaje por medio de una teoría subyacente conductista de la enseñanza, guían los aprendizajes y comparan los resultados de los alumnos contra patrones, generando muchas veces nuevas ejercitaciones de refuerzo, si en la evaluación no se superaron los objetivos de aprendizaje.

- Simuladores: ejercitan los aprendizajes inductivo y deductivo de los alumnos mediante la toma de decisiones y adquisición de experiencia en situaciones imposibles de lograr desde la realidad, facilitando el aprendizaje por descubrimiento.
- Herramientas de apoyo: tales como los procesadores de textos, planillas de cálculo, sistemas de gestión de bases de datos, programas de comunicación, que no están dentro de la clasificación de educativos, pero muchas veces son necesarios para la redacción final de trabajos, informes y monografías.

1.1.4 Funciones del Software Educativo. Las funciones del software educativo serán definidas según Marqués (1995) en la siguiente tabla⁵:

Tabla 3. Funciones del Software Educativo

Función	Descripción
Informativa	Presentan contenidos que proporcionan una información estructurada de la realidad. Representa la realidad y la ordenan. Tales como bases de datos, simuladores, tutoriales.
Instructiva	Promueven actuaciones de los estudiantes encaminadas a facilitar el logro de los objetivos educativos, el ejemplo son los programas tutoriales.
Motivadora	Suelen incluir elementos para captar el interés de los alumnos y enfocarlo hacia los aspectos más importantes de las actividades.
Evaluadora	Al evaluar implícita o explícitamente, el trabajo de los alumnos.
Investigadora	Los más comunes son: las bases de datos, los simuladores y los entornos de programación.
Expresiva	Por la precisión en los lenguajes de programación, ya que el entorno informático, no permite ambigüedad expresiva.
Metalingüística	Al aprender lenguajes propios de la informática.
Lúdica	A veces, algunos programas refuerzan su uso, mediante la inclusión de elementos lúdicos.
Innovadora	Cuando utilizan la tecnología más reciente.

⁵ La información de las tablas fue tomada de <http://laboratorios.fi.uba.ar/lsi/cataldi-tesisdemagistereninformatica.pdf>

1.1.5 Ingeniería de software educativo. Para el desarrollo de un software educativo se deben tener en cuenta dos elementos esenciales: la estructura y el modelo de aprendizaje. La estructura, puede ser tan formal o informal como el desarrollador decida respecto a lo que quiera hacer, debido a que ésta se puede modelar. En cuanto al modelo de aprendizaje, por ser software educativo requiere de un objetivo de aprendizaje relacionado con la serie de conocimientos que se quieren enseñar. Este modelo a diferencia de la estructura es implícito, pocas veces el desarrollador conoce de él, sin embargo debe buscar una forma para hacer llegar los conocimientos al usuario final.

Para desarrollar un Material Educativo Computarizado (MEC) de calidad, es necesario tener presentes cuales son los aspectos que se consideran importantes para su desarrollo. Los aspectos se miran desde tres enfoques: su entorno, tratamiento y metodología de desarrollo. Desde su entorno se considera a la población objetivo, cuál será el área de conocimiento, que necesidades educativas se van a resolver, cuales son las limitaciones y recursos para los usuarios del MEC, que equipo se requiere, otro software requerido como soporte (además del MEC), documentación (manuales del MEC), medios de transferencia. También se considera importante la estrategia didáctica escogida (tratamiento), examinando la forma como se guiará al estudiante hacia el aprendizaje, el sistema de motivación y de refuerzo que se usará, así como el sistema de evaluación que se utilizará.

1.1.6 Software educativo como medio de enseñanza. Una aplicación informática soportada en una estrategia pedagógica bien definida que apoya el proceso de enseñanza-aprendizaje se denomina un software educativo. Este tipo de software constituye un efectivo elemento en el desarrollo de la educación.

El desarrollo de software educativo en los últimos tiempos, se ha sido considerado como un elemento didáctico interactivo que se elabora a partir de la representación de conocimiento facilitándole al usuario la realización de

actividades por medio de elementos que le permiten solucionar problemas impactando su estructura cognitiva. En la figura 1 se muestran usuarios realizando actividades específicas requeridas apoyadas en procesos cognitivos tales como: observación, identificación, construcción y creación.

Los simuladores, textos electrónicos e hipertextos son algunos de los elementos específicos que se consideran como software educativo, es decir, programas elaborados en una plataforma informática que buscan apoyar el desarrollo de temáticas específicas incluidas en el sistema educativo y que poseen una clara intención pedagógica.

El papel de la informática dentro de la educación se caracteriza por ser un componente de apoyo al proceso de enseñanza y aprendizaje. El software educativo se define como un elemento didáctico que diseña ambientes basados en los requerimientos cognitivos de los estudiantes.

Figura 1. Interacción de usuarios 1



El software educativo es el que se dedica a facilitar diferentes procesos de los sistemas de educación, principalmente procesos tales como: enseñanza-aprendizaje, vinculación con la práctica laboral, investigación estudiantil, gestión académica, etc. Permitiendo incluir sistemas computacionales como medios

auxiliares en procesos didácticos que abarcan objetivos, contenidos, medios, métodos y evaluación, sobre temáticas específicas.

El software educativo como medio de enseñanza resulta eficiente auxiliar para docentes en cuanto a la preparación de las clases ya que contribuyen a una mayor ganancia metodológica haciendo más sencillo el proceso de enseñanza en las actividades del profesor y el alumno. Además proporcionan beneficios pedagógicos pues motivan a los alumnos en la realización de actividades, estimulan el pensamiento abstracto, permiten la interactividad evaluando lo aprendido, desarrollan habilidades, simulan procesos complejos, facilitan el trabajo independiente e introducen al estudiante en el uso de las técnicas más avanzadas, por lo que el uso de estos novedosos medios de enseñanza se hacen hoy prácticamente imprescindibles.

La tendencia actual en los ambientes de aprendizaje apoyados por computadora es hacia sistemas de soporte, enfocados al entrenamiento más que a la tutoría.

Es importante resaltar que el software educativo debe estar diseñado con un propósito educativo. Sus contenidos, metodología y evaluación deben ayudar a alcanzar los objetivos asignados en cada etapa del sistema educativo permitiendo mejorar el proceso de enseñanza-aprendizaje.

1.1.7 Software educativo como estrategia educativa. Aprender por sí solos debe ser una práctica que se empieza a realizar desde los primeros años de vida. Esto significa que se deben seguir procesos de superación continua con una metodología orientada hacia el estudio independiente complementándola con el trabajo en equipo.

Con respecto al trabajo independiente, el Dr. Carlos Álvarez en su libro⁶ ofrece la siguiente definición: “El trabajo independiente es el modo de organización del proceso docente dirigido a la formación de la independencia, como característica de la personalidad del estudiante.”

El trabajo independiente es uno de los medios más efectivos de actividad cognoscitiva del estudiante. Ya que por su nivel de independencia y concientización del proceso, logra un mayor nivel de profundización científica y desarrollo de habilidades generales que la docencia sola no puede alcanzar. En la figura 2 se observa a uno de los usuarios realizando las actividades de forma independiente mostrando un alto grado de concentración.

El aprendizaje adquiere el rango de autónomo cuando el proceso educativo de enseñanza-aprendizaje toma a cada estudiante como el principal protagonista de su propia realidad personal, comprometiéndolos a alcanzar niveles de auto desarrollo que aún no han explotado para dar soluciones a problemas propuestos en diversas áreas escolares.

La función de las herramientas diseñadas para software educativo es servir de guía de la conducta humana en las actividades que se estén desarrollando. El uso específico de estas aplicaciones permite elaborar soluciones concretas de una tarea, ya que le hace llegar al estudiante formas, métodos y prácticas usuales para mejorar el entorno de aprendizaje y contribuir a la adquisición de habilidades en su proceso de formación⁷.

⁶ Libro *La Escuela en la vida* (Álvarez de Zayas, p. 51)

⁷ PASCUAL FERNÁNDEZ MAYRA, Archivo Histórico Provincial de Villa Clara .Cuba

Figura 2. Interacción de usuarios 2



1.1.8 La música por computadora. Desde sus inicios el hombre se ha esforzado por crear y ejecutar música de forma automática. La forma natural es la voz, no obstante se construyeron instrumentos musicales tales como las liras, arpas, trompetas, que al ser combinados ofrecían nuevas posibilidades de generar música variando parámetros importantes como intensidad, timbre, entre otros.

En este siglo se dieron dos grandes avances técnicos que ocasionaron un giro definitivo a la automatización y generación de sonidos y por consiguiente a la composición y ejecución de la misma: los sintetizadores electrónicos y las computadoras digitales. Hacia los años 70, comenzaron a utilizarse los sintetizadores digitales, existían computadoras dedicadas a la producción de sonido. A comienzos de la década de los 80 se hizo evidente la necesidad de una norma que permitiera la comunicación autónoma de los instrumentos musicales entre sí con una computadora. Esto condujo a la creación de la norma MIDI hoy día universalmente adoptada.

Con los avances tecnológicos y las posibilidades de adquirir equipos sistemáticos debido a la disminución en los precios de los productos electrónicos, se hizo cada vez más accesible la posibilidad de producir y ejecutar música propia. Así, con la

ayuda de secuenciadores se hizo posible ejecutar comandos a una computadora para generar sonidos, agregando algunas configuraciones extras con el objetivo de lograr una interpretación con características “humanas”. En la figura 3 se observa la conexión de un dispositivo externo; un teclado, a un computador para generar sonido de un piano de forma similar a la realidad.

Figura 3. Música por computador



1.1.9 Computadoras aplicadas a la música. Para este tipo de computadoras existen dos enfoques. El primero consiste en dotar a la computadora de un periférico (plaqueta o tarjeta de sonido) que realice la conversión digital/analógica, es decir, que convierta los números que la computadora va generando en señales eléctricas analógicas, que se aplican a un altavoz.

El segundo enfoque se basa en utilizar un sintetizador y controlarlo desde la computadora por medio de mensajes. El proceso es similar al primer caso, los mensajes MIDI se generan en tiempo real por medio de algún algoritmo, o en tiempo diferido. Este archivo MIDI debe ser creado con anticipación a su uso, y se puede generar mediante un algoritmo, o memorizando los mensajes MIDI obtenidos durante la ejecución en tiempo real por medio de un controlador MIDI.

En conclusión, en el primer enfoque, la computadora construye el sonido, mientras que en el segundo, sólo transmite órdenes a un dispositivo externo; el sintetizador,

y es éste quien tiene la labor de generar el sonido, según las instrucciones que reciba por medio del algoritmo.

1.1.10 Entorno MIDI. Las siglas MIDI representan no sólo la norma de comunicación entre instrumentos y computadoras, también se habla de instrumentos MIDI, interfaces MIDI, controladores MIDI, cables MIDI, archivos MIDI, programación MIDI, etc. Hasta el punto que podría hablarse de una filosofía MIDI.

La función principal de MIDI es intercambiar una serie de mensajes. Esto se puede hacer desde las siguientes rutas:

- Desde un controlador MIDI hacía un sintetizador.
- Desde un controlador MIDI hacía una computadora.
- Desde una computadora hacia un sintetizador.

Cada instrumento tiene una salida conocida como: MIDI Out y por supuesto una entrada MIDI In. Al conectar la salida de un equipo con la entrada de otro equipo, se logra dirigir los mensajes desde un equipo hacia otro, la ruta se selecciona dependiendo la conexión que se haya hecho. La primera ruta se utiliza cuando se quiere ejecutar música con el sintetizador en forma manual, básicamente como si lo hiciera un instrumento acústico. La segunda ruta se emplea para grabar una ejecución manual. Es importante aclarar que no se graba la música como tal sino las instrucciones para reproducirla; específicamente parámetros como timbre, duración, intensidad, etc. La tercera ruta se refiere a la ejecución automática, la computadora envía órdenes al sintetizador. Estas órdenes pueden haber sido grabadas anteriormente a partir de una ejecución manual, o pueden haber sido ingresadas desde el teclado alfanumérico de la computadora con la codificación que use el programa.

Las rutas se pueden combinar entre sí obteniendo resultados favorables. Por ejemplo sería posible ejecutar manualmente un instrumento, al mismo tiempo que la computadora ejecuta uno o más instrumentos automáticamente. Así mismo una persona podría tocar uno a uno todos los instrumentos de un conjunto instrumental. Lógicamente, todos los instrumentos se podrán ejecutar desde un mismo controlador, para facilitar el trabajo del músico. Esto es posible porque a la computadora se le puede solicitar que ejecute los mensajes MIDI con un timbre de los que hay disponibles, por ejemplo con un teclado se pueden generar sonidos de una flauta.

Los programas que utilizan secuenciadores permiten ingresar las órdenes de diferentes maneras, una de ellas, es con un intérprete humano en tiempo real. Otra es ingresar las notas despacio facilitando la ejecución. Y otra es aquella que no requiere el manejo técnico de ningún instrumento, se ingresan nota por nota desde el teclado los distintos parámetros asociados: duración, timbre, intensidad, etc. Existen algunos programas que diseñan una interfaz amigable permitiendo ingresar parámetros por medio de íconos de notación musical tradicional, mientras que en otros casos estos se ingresan por el teclado.

MIDI tiene una serie estándar de mensajes que son comunes a todos los instrumentos MIDI y fabricantes entre los cuales están la activación y desactivación de notas, timbre, la afinación, el volumen, etc. Por otro lado existe una serie de mensajes no estandarizados, que los define cada fabricante. Con estos mensajes MIDI especiales se pueden controlar las diversas funciones específicas de cada instrumento.

1.1.11 El problema de la interpretación. Existe un inconveniente cuando se ejecuta música por medio de una computadora y es lograr que la interpretación sea musicalmente convincente. Hay dos hechos que prueban que hacer esto teóricamente posible. El primero es utilizar un sintetizador de buena calidad, así

los intérpretes humanos podrán generar interpretaciones similares a las de un instrumento acústico. Estas interpretaciones serán grabadas en forma de mensajes MIDI para ser reproducidas finalmente por la computadora. El segundo es que la computadora controle con precisión superior a la de un humano los dos parámetros básicos que gobiernan la interpretación que son: la intensidad del sonido y el tiempo.

1.1.12 La notación musical. La notación musical se basa en dos principios. El primero es que las duraciones de los sonidos y los silencios son múltiplos o submúltiplos de la unidad de tiempo correspondiente y esta es constante. La segunda es que el sonido musical se puede dividir en pequeños fragmentos denominados compases. Cada compás tiene un número de unidades de tiempo.

La ventaja de estas notaciones es que hay una única manera posible de representar cada pieza musical. La desventaja es que la notación que se obtiene no proporciona recursos para identificar qué es lo que se está interpretando. Esta desventaja se reduce cuando es un humano experimentado el que la ejecuta, ya que existen convenciones que el músico utiliza consciente o inconscientemente para poder interpretarlo.

1.1.13 La composición por computadora. La computadora se puede utilizar para varios fines de la composición. En primer lugar, funcionando como secuenciador para permitir la verificación inmediata de lo que se está componiendo. En segundo lugar, puede contribuir realizando cálculos que se requieren para reproducir una forma musical que sea compleja. En tercer lugar, puede componer automáticamente sonidos musicales según las pautas que se introduzcan.

Existen programas elaborados por institutos de investigación de diversas partes del mundo, que debido a la diversidad de conocimiento ofrecen posibilidades más avanzadas en cuanto a los procesos creativos. Por ejemplo, es posible trabajar en

forma aleatoria con varios sonidos controlando parámetros globales, tales como la frecuencia, duración del sonido, el timbre, el tempo, etc. También es posible implantar reglas de composición que controlen que tipos de acordes son permitidos.

1.1.14 La música por computadora en la educación. Existen grandes posibilidades para la música por computadora en la educación musical. Una de ellas es que permite realizar un estudio minucioso de las partituras existentes para determinado conjunto instrumental, ejecutando las partes de forma individual, en grupos, o en su totalidad, comparando las distintas versiones. También se utiliza para enseñar diversas técnicas de composición y para el estudio de la instrumentación. Otra forma de uso es emplearla como patrón para el aprendizaje de un instrumento, detectando y corrigiendo errores. En algunas ocasiones se puede utilizar para enseñar los procesos interpretativos, analizando interpretaciones hechas sobre un controlador y almacenadas como datos MIDI por intérpretes expertos, así como las propias de los alumnos. La lista de posibilidades de uso depende exclusivamente de las necesidades que se presenten y de la imaginación de los interesados en hacer esta clase de recursos que poco se han explotado.

Lo anterior hace referencia a que la computadora se debe utilizar como accesorio pedagógico para enseñar de forma eficiente el conocimiento que se ha estado transmitiendo de forma tradicional.

Una gran ventaja de aprender música por computadora es que convertirse en instrumentista requiere además de capacidades motrices fuertes disciplinas durante 10 ó 12 años aproximadamente. Mientras que adquirir conocimiento y técnica para controlar un sintetizador de forma sistemática podría tardar solo un par de años que complementados con algunas prácticas y rutinas de entrenamiento causarían el mismo efecto. Los músicos que aprenden por medio

de computadoras tienen una visión mucho más amplia, ya que no se limitan solo a un instrumento, sino que pueden realizar interpretaciones con conjuntos completos, como por ejemplo sonidos electrónicos.⁸

No se trata de reemplazar con un software educativo procesos que están probados por otros medios con calidad sino de aprovechar los recursos que ofrece un computador para apoyar los procesos de enseñanza-aprendizaje. El software educativo busca complementar aquello que con otros medios es más difícil de lograr.

1.2 SOPORTE TECNOLÓGICO

1.2.1 Netbeans. Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. Es un producto libre y gratuito sin restricciones de uso, de código abierto y gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

Esta plataforma permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en esta plataforma pueden ser extendidas fácilmente por otros desarrolladores de software.

⁸ MIYARA FEDERICO, Secretaria de cultura de la provincia de Santa Fe, La música por computadora.

Durante el desarrollo ocurrió una cosa interesante. La gente empezó a construir aplicaciones usando el NetBeans core runtime con sus propios plug-ins, de hecho, esto se convirtió en un mercado bastante grande.

NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.⁹

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas)
- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato)
- Administración de ventanas
- Framework basado en asistentes (diálogos paso a paso)

1.2.2 Java. Java surgió en 1991 cuando un grupo de ingenieros de Sun Microsystems trataron de diseñar un nuevo lenguaje de programación destinado a electrodomésticos. La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño muy reducido.

⁹ Definición referenciada: Wikipedia.org

Debido a la existencia de distintos tipos de CPUs y a los continuos cambios, era importante conseguir una herramienta independiente del tipo de CPU utilizada. Desarrollaron un código “neutro” que no dependía del tipo de electrodoméstico, el cual se ejecutaba sobre una “máquina hipotética o virtual” denominada Java Virtual Machine (JVM). Era la JVM quien interpretaba el código neutro convirtiéndolo a código particular de la CPU utilizada. Esto permitía lo que luego se ha convertido en el principal lema del lenguaje: “Write Once, Run Everywhere”. A pesar de los esfuerzos realizados por sus creadores, ninguna empresa de electrodomésticos se interesó por el nuevo lenguaje.

Como lenguaje de programación para computadores, Java se introdujo a finales de 1995. La clave fue la incorporación de un intérprete Java en la versión 2.0 del programa Netscape Navigator, produciendo una verdadera revolución en Internet. Java 1.1 apareció a principios de 1997, mejorando sustancialmente la primera versión del lenguaje. Java 1.2, más tarde rebautizado como Java 2, nació a finales de 1998.

Al programar en Java no se parte de cero. Cualquier aplicación que se desarrolle se apoya en un gran número de clases preexistentes. Algunas de ellas las ha podido hacer el propio usuario, otras pueden ser comerciales, pero siempre hay un número muy importante de clases que forman parte del propio lenguaje (el API o Application Programming Interface de Java). Java incorpora en el propio lenguaje muchos aspectos que en cualquier otro lenguaje son extensiones propiedad de empresas de software o fabricantes de ordenadores (threads, ejecución remota, componentes, seguridad, acceso a bases de datos, etc.). Por eso muchos expertos opinan que Java es el lenguaje ideal para aprender la informática moderna, porque incorpora todos estos conceptos de un modo estándar, mucho más sencillo y claro que con las citadas extensiones de otros lenguajes. Esto es consecuencia de haber sido diseñado más recientemente y por un único equipo.

La versatilidad y eficiencia de la tecnología Java, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en la tecnología ideal para su aplicación a redes. De portátiles a centros de datos, de consolas de juegos a súper equipos científicos, de teléfonos móviles a Internet, Java está en todas partes.

Java ha sido mejorado, ampliado y probado por una comunidad especializada de más de 6,5 millones de desarrolladores, la mayor y más activa del mundo. Esta tecnología le permite a los desarrolladores:

- Desarrollar software en una plataforma y ejecutarlo en prácticamente cualquier otra plataforma.
- Crear programas para que funcionen en un navegador web y en servicios web.
- Desarrollar aplicaciones para servidores como foros en línea, tiendas, encuestas, procesamiento de formularios HTML, etc.
- Combinar aplicaciones o servicios que usan el lenguaje Java para crear servicios o aplicaciones totalmente personalizados.
- Desarrollar potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo coste y prácticamente cualquier tipo de dispositivo digital.

El principal objetivo del lenguaje Java es llegar a ser el “nexo universal” que conecte a los usuarios con la información, situada en el ordenador local, en un servidor de Web, en una base de datos o en cualquier otro lugar.

2. METODOLOGÍA DE DESARROLLO DEL PROYECTO

Desarrollar un software para apoyo de procesos de aprendizaje es una tarea difícil porque para ser aprobada por instituciones educativas se requiere que cumpla condiciones de usabilidad de alta calidad. Para lograr que se ajuste a esas condiciones se deben tener en cuenta aspectos educativos y formativos que garanticen el cumplimiento de las necesidades planteadas.

Este proyecto no está dirigido hacia alguien en particular sino a una población específica: la población infantil, sin embargo para identificar las necesidades fue importante tener contacto frecuente con los usuarios finales porque son ellos los que manifiestan las necesidades para que se pueda dar pronta solución estableciendo mecanismos adecuados que se basen en fundamentos educativos.

Para dar inicio se hizo una identificación de cuáles debían ser la características del software que se realizó teniendo como referencia algunas aplicaciones ya existentes similares a la que se desarrolló.

La metodología que se utilizó fue la propuesta por Álvaro Galvís¹⁰ en esta se conservan las etapas de un proceso sistemático (análisis, diseño, desarrollo, prueba, ajuste e implementación) pero en este caso existe un énfasis en el diseño de ambientes educativos computarizados que se evalúa permanentemente bajo algunos criterios ya establecidos durante las etapas del proceso ayudando a un perfeccionamiento continuo del software.

En la siguiente figura se muestra el flujo de acción de la metodología y se señala que hay dos maneras de ejecutarse; una podría ser en función de los resultados

¹⁰ Formador en el uso de pedagogías activas apoyadas en ambientes educativos. Revisar <https://sites.google.com/site/galvaro50/>

- Problema o necesidad a atender: El principal problema que surge es la necesidad de crear un software interactivo y didáctico que tenga como objetivo crear una rutina que le facilite el aprendizaje a los niños al momento de leer una tablatura musical en un instrumento de percusión. Esto debido a que el software existente sobresale por estar dedicado netamente al entretenimiento, los que hay con fines educativos no tienen una interfaz apropiada para que la utilicen niños o personas sin conocimientos musicales.
- Análisis de alternativas de solución: La solución fue la creación del software Combat bajo parámetros descritos en la metodología teniendo en cuenta el cumplimiento del objetivo planteado y pensando en los usuarios a los que va dirigido, contemplando la usabilidad como requisito fundamental en el desarrollo de este proyecto. Según la clasificación hecha por el Dr. Álvaro Galvís lo que se desarrolló es un simulador que puede usarse para que el aprendiz llegue al conocimiento por medio de un trabajo exploratorio, conjetural y a través de aprendizaje por descubrimiento dentro de un micromundo que se acerca razonablemente, en su comportamiento a la realidad o aquello que se intenta modelar.
- Establecimiento de la herramienta en la que se va a desarrollar: En un principio se había pensado en desarrollar la aplicación en flash ya que este organiza las imágenes y sonidos para crear animaciones 2D que se utilizan en sitios Web con contenido multimedia mostrándose como una buena opción de herramienta de desarrollo. Después se pensó en java debido a que los aplicativos desarrollados en este lenguaje pueden ejecutarse en cualquier dispositivo, permite desarrollar herramientas como juegos, interfaces web, software para dispositivos móviles, entre otros. La licencia de java es GNU GPL lo cual permite su libre uso, modificación y distribución, convirtiéndose java en la mejor opción para desarrollar e implementar el software.

De acuerdo a la etapa de análisis se formulan los requerimientos que debe tener el software que se va a desarrollar. Los requerimientos para esta aplicación son los siguientes:

Requerimientos funcionales:

- Brindar una interfaz sencilla de manejar y agradable para los usuarios.
- Proporcionar un ambiente de aprendizaje que mantenga motivado y activo al usuario.
- Garantizar que el sonido de los instrumentos de percusión sea claro y distinguible al oído del usuario.
- Permitir al usuario componer ritmos insertando las figuras musicales.
- Permitir al usuario reproducir ritmos creados por él y ritmos incluidos por defecto en el sistema.
- Garantizar que el trabajo que el usuario haya realizado no se pierda.
- Realizar una prueba piloto para medir el impacto del software.

Requerimientos no funcionales:

- El software debe ocupar toda la pantalla del computador en el que se ejecute.
- La aplicación debe poder ejecutarse en la mayoría de sistemas operativos.
- Las herramientas de ayuda como botones deben estar ubicadas en un lugar visible.
- Los iconos, dibujos y gráficos utilizados en el software deben ser intuitivos.
- El software debe tener sensibilidad ante actividades que realice el usuario como forma de respuesta.

La aplicación como ya se ha mencionado en algunas ocasiones está dirigida a niños, pero no se limita a que sólo ellos hagan uso de esta. También puede ser utilizada por usuarios de otras edades que estén interesados.

El modo de uso de la aplicación es individual pero no está de más el apoyo de alguien mayor que pueda supervisar el rendimiento del niño en caso de que este pueda presentar confusiones o dudas respecto al uso del software.

Antes de empezar a usar el software no se requiere ningún tipo de experiencia, va a poder usarse como la mayoría de aplicaciones y juegos que se conocen. Para facilitar el ingreso no va a haber una pantalla de registro de usuario porque se corre el riesgo de que las contraseñas sean olvidadas y esto impida el buen uso de la aplicación. Por tanto el software consta de una pantalla con el logo de la aplicación que desaparece por sí sola para mostrar enseguida el escenario de interacción que corresponde a la pantalla en la que se crean y reproducen los ritmos. En cuanto a conocimientos en música no serán necesarios porque todo está elaborado de forma intuitiva precisamente pensando en que el principal elemento que usen los niños sea su creatividad a la hora de crear ritmos y su lógica para empezar a descubrir el funcionamiento de la tablatura asociándolo a sonidos y tiempos manejados en el software.

Etapa 2: Diseño de MECs

En el diseño se tomó como punto inicial la necesidad que había que por resolver estableciendo el objetivo principal del software que era crear una rutina de entrenamiento. Por tratarse de software educativo se adquiere un compromiso mínimo de aprendizaje, en este caso se espera que todo aquel que lo use aprenda conceptos básicos cómo: cuál es el tiempo de duración de las notas, qué es el tempo, qué es el volumen y qué instrumentos conforman el set de batería.

El diseño está en función directa de los resultados de la etapa de análisis. La orientación y contenido de la aplicación se deriva de la necesidad educativa que justifica su elaboración, así como el interés del usuario por aprender sobre el tema en caso de que este no tenga ningún conocimiento previo.

Es necesario atender a tres tipos de diseño: *Educativo* (para resolver los interrogantes que se refieren al alcance, contenido y tratamiento que debe apoyar el SE), *comunicacional* (es donde se maneja la interacción entre usuario y maquina se denomina interfaz), y *computacional* (con base a las necesidades se estable qué funciones es deseable cumpla el SE en apoyo de sus usuarios, el docente y los estudiantes).

Diseño Educativo:

En esta fase del diseño se resuelven interrogantes relacionados con el alcance, el contenido y el tratamiento que debe apoyar la aplicación. Este software busca convertirse en una herramienta de apoyo que cree una rutina de entrenamiento facilitando la comprensión de conceptos musicales ilustrados de forma sencilla. Buscando generar aprendizaje por asociación, es decir cuando ocurren dos fenómenos al mismo tiempo como es colocar una nota en la tablatura y que se genere un sonido.

¿Qué aprender con el MEC?

Al hacer un estudio de mercado de herramientas musicales diseñadas para niños se observó que la mayoría de ellas carecen de contenido apropiado al usuario para comprender temas relacionados con la composición de música por medio de computadora y solo ofrecen entretenimiento y diversión. Con esta aplicación se busca que los usuarios especialmente niños por la interfaz llamativa, herramientas básicas representadas por imágenes, botones grandes sencillos y fáciles de

entender puedan crear música por sí mismos haciendo uso de figuras musicales, controles y una variedad de instrumentos. En el micromundo que se realizó para esta aplicación se materializó un modelo matemático de cómo componer música para facilitar y ejercitar a los usuarios combinando la simulación y el juego con el fin de mostrar la teoría y la práctica. Como es un micromundo el usuario tiene la capacidad de variar parámetros como tempo, volumen y hacer repeticiones de fragmentos de ritmos con el botón loop conduciéndolo a crear sus propios ambientes o personalizar ambientes existentes que le ayuden a comprender y crear nuevos conceptos. Se presupone la construcción de conocimiento y aprendizaje por el descubrimiento y la activa participación del usuario teniendo en cuenta que tendrá que pensar y construir conocimiento a través de los mecanismos disponibles para la composición de ritmos en donde haciendo uso de la creatividad y la invención creara sus propias canciones cumpliendo de esta forma el principal objetivo del software. El aprendizaje será constructivista debido a que el eje fundamental es la actividad propia del usuario acerca del objeto de estudio según sus intereses que se fundamenta en la exploración, descubrimiento, cooperación, reflexión y la elaboración de productos significativos.

¿En qué ambiente o micromundo aprenderlo?

En un micromundo dedicado a la creación de ritmos para batería que ofrece una serie de recursos para que esto se pueda realizar. La forma en la que se componen ritmos es sencilla e interactiva, el usuario ubica notas desde un panel hasta la cuadrícula de la tablatura, cuando las notas se dibujan emiten sonidos correspondientes a un instrumento, esto debido a que a cada fila de la tablatura le corresponde uno de los instrumentos del set de batería. Los elementos principales que hacen parte del micromundo son: las notas, los instrumentos y la tablatura porque son los que permiten la construcción de partituras musicales. En los casos de uso se especifica la interacción de los objetos y la secuencia de los mensajes entre ellos.

Diseño Comunicacional:

En esta fase del proyecto se define la zona de comunicación e interacción entre el usuario y el programa, esto comúnmente se conoce como *interfaz*. Las interfaces gráficas de usuario (IGU), son una representación visual de la forma en la que el usuario tiene contacto con la aplicación. Para el diseño de la pantalla se tuvo en cuenta que los principales usuarios del software son niños, por esta razón tiene colores llamativos, gráficos ilustrativos, sonidos claros generando alto grado de usabilidad construyendo una interfaz amigable e intuitiva.

El modelo computacional de la interfaz consta de:

- Una pantalla de inicio que contiene el logo de la aplicación con el fin de que el usuario identifique que se está ejecutando. Esta pantalla desaparece de forma automática.

- La pantalla de creación y reproducción de ritmos es la pantalla principal del sistema, está dividida en 5 paneles:
 - ✓ Panel superior: este contiene herramientas como: abrir, guardar, seleccionar, eliminar y limpiar. También muestra el nombre del ritmo que se está reproduciendo y en la esquina superior derecha muestra el logo de la aplicación.
 - ✓ Panel central izquierdo: muestra los instrumentos, estos tienen actividades específicas determinadas por la acción que se ejecute.
 - ✓ Panel central: en este se encuentra la tablatura, la cual se muestra como una cuadrícula de 16x4. La forma en la que se maneja es por medio de compases.
 - ✓ Panel central derecho: aquí están las notas en forma de botones.

- ✓ Panel inferior: es en donde aparecen los siguientes controles: play, stop, pause, loop, tempo y el volumen.
- La interfaz requiere de algunas características especiales al realizar operaciones tales como:
 - Para cambiar de lugar una nota de la tablatura existen dos formas: haciendo clic izquierdo y cambiando el cursor o también oprimiendo el botón de seleccionar y ubicar el cursor del mouse sobre la imagen de la nota. En los dos casos el cursor adoptará la forma de la nota y se dibujará en la nueva posición que el usuario escoja.
 - Para eliminar una nota de la tablatura se selecciona el icono eliminar y se ubica sobre la imagen de la nota haciendo clic, inmediatamente la nota desaparecerá.
 - Para que no suene un instrumento de los del set se debe hacer clic izquierdo sobre el instrumento que se desea cancelar y este cambiara el color mostrando que está desactivado. Para activarlo se hace la misma operación, clic izquierdo sobre el instrumento y este retomara su estado original.
 - Para cambiar alguno de los instrumentos se hace clic derecho sobre el instrumento y este mostrará una lista desplegable, al ubicarse sobre alguno de los nombres de la lista cambiara automáticamente y se mostrará uno nuevo.
 - Para agregar un nuevo compás se debe oprimir el botón lateral que muestra el ícono (+).

Diseño Computacional:

En la fase de diseño comunicacional se logó definir como era la interacción de los objetos con la interfaz y en esta fase se mostrará cómo se desarrolló esta labor.

Al llegar a esta etapa ya se han definido cuales son los objetos, atributos e incluso métodos que se van a tener en cuenta en cada una de las clases que se realicen debido a que se ha hecho un trabajo de ingeniería del software que permite recolectar esta información por medio de tablas de eventos, identificación de objetos de entidad, de borde y de control, casos de uso y diagramas de clase. Todo esto con el fin de facilitar la implementación en la herramienta de desarrollo que se ha seleccionado.

Tabla 4. Tabla de Eventos:

Evento:	Entrada al Sistema:	Actor que provee la entrada al Sistema	Salida al Sistema:	Actor que recibe la entrada:
Ingresar a COMBAT	Datos de la Configuración.	Usuario	COMBAT configurado.	Usuario
El usuario compone un ritmo.	Notas de la partitura.	Usuario	Tablatura con notas.	Usuario
El usuario reproduce el ritmo.	Algoritmo para reproducir notas.	Usuario	Reproducción del compás.	Usuario
El Usuario guarda la composición.	Nombre del archivo.	Usuario	Archivo.	Usuario

Diagrama de Casos de Uso:

Figura 5. Diagrama de casos de uso

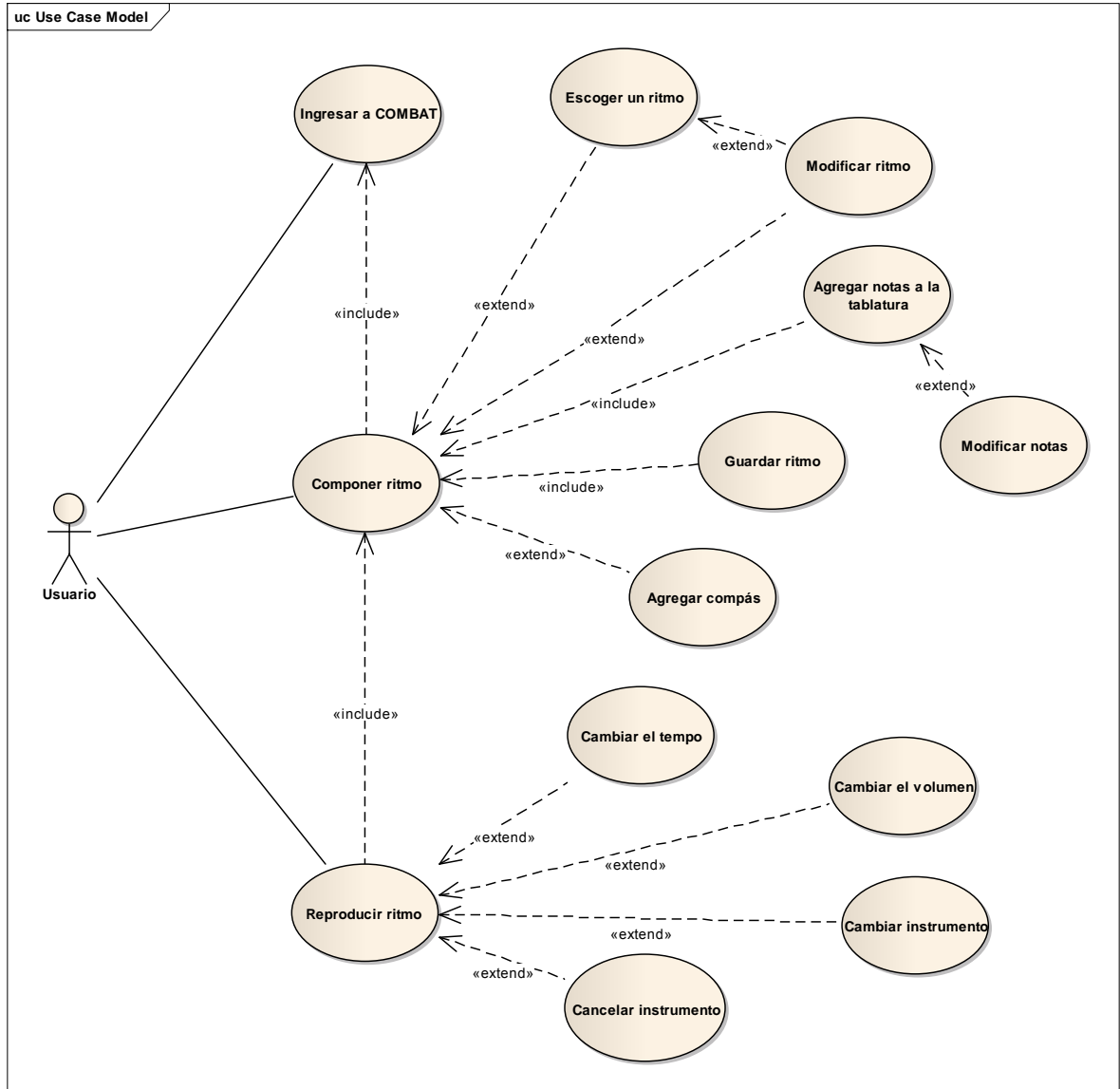


Tabla 5. Especificación de los casos de usos:

CASOS DE USO 1		
Caso de Uso	Ingresar a COMBAT	
Actor(es)	Usuario	
Propósito	Que el usuario pueda ingresar satisfactoriamente al sistema.	
Descripción	Este caso de uso inicia cuando el usuario ha ingresado al sistema.	
Precondición(es)	Que la aplicación este instalada y permita que se pueda ejecutar sin problema.	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	<ol style="list-style-type: none"> 1. El usuario ejecuta la aplicación. 1. El usuario hace clic para ingresar al sistema. 	<ol style="list-style-type: none"> 2. Se muestra una pantalla de bienvenida. 2. Se muestra la pantalla principal del software.
Sub-flujos	Paso 1: Si el usuario no puede ingresar al sistema.	
Poscondición(es)	Abre la pantalla principal del sistema.	

CASOS DE USO 2		
Caso de Uso	Componer ritmo.	
Actor(es)	Usuario	
Propósito	Que el usuario comprenda el funcionamiento del sistema ubicando notas en la tablatura que completen los tiempos determinados para un compás 4/4.	
Descripción	Este caso de uso inicia cuando el usuario coloca notas en la tablatura que serán interpretadas por el software como un arreglo de sonidos con diferentes tiempos de duración.	
Precondición(es)	Que el usuario coloque 1 o más notas en la tablatura.	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	<ol style="list-style-type: none"> 1. El usuario arrastra una nota a la tablatura. 4. Coloca más notas en la tablatura. 	<ol style="list-style-type: none"> 2. Deshabilita la nota que se ha seleccionado. 3. Dibuja la nota seleccionada por el usuario y emite un

		sonido.
Sub-flujos	Paso 1: Si el usuario no dibuja ninguna nota en la tablatura.	
Poscondición(es)	Se muestra la tablatura con notas lo cual indica que se ha creado un ritmo.	

CASOS DE USO 3		
Caso de Uso	Reproducir ritmo	
Actor(es)	Usuario	
Propósito	Que el usuario después de crear o cargar un ritmo oprima la opción "play" para activar la reproducción de lo que hay en la tablatura en ese momento.	
Descripción	Este caso de uso es el que le permite al usuario escuchar el sonido de las notas en conjunto que hay puestas en la tablatura.	
Precondición(es)	Que se haya creado algún ritmo o se cargue un ritmo de los que hay por defecto en el sistema.	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	1. El usuario oprime el botón "play".	2. Reproduce cada una de las notas y muestra una línea guía para indicarle al usuario en que parte de la reproducción esta en un instante de tiempo.
Sub-flujos	Paso 1: Si el usuario no carga uno de los ritmos por defecto o no componga ningún ritmo y la tablatura este vacía.	
Poscondición(es)	Reproducir ritmos nuevos o precargados en el sistema.	

CASOS DE USO 4		
Caso de Uso	Escoger un ritmo	
Actor(es)	Usuario	
Propósito	Que el usuario pueda seleccionar algunas de las composiciones que se encuentran en el sistema con el fin de tener un modelo a seguir para elaborar su propia composición.	
Descripción	Este caso de uso es el que le permite al usuario cargar en la tablatura ritmos que tiene el sistema por defecto incluidos para facilitar el manejo de las notas.	
Precondición(es)	Que existan ritmos por defecto creados por el desarrollador en el sistema.	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	1. El usuario despliega los ritmos que se encuentran en el menú y selecciona uno.	2. En la tablatura se muestran las notas que conforman el ritmo que ha seleccionado el usuario.
Sub-flujos	Paso 1: Si el usuario no carga alguno de los ritmos que existen en el sistema.	
Poscondición(es)	La tablatura se muestra llena de notas.	

CASOS DE USO 5		
Caso de Uso	Modificar un ritmo	
Actor(es)	Usuario	
Propósito	Que el usuario pueda hacer uso de ritmos creados anteriormente como base para la creación de otros nuevos.	
Descripción	Este caso de uso es el que le permite al usuario cambiar ritmos que han sido creados por él o que se encuentran por defecto en el sistema.	
Precondición(es)	Que haya un ritmo en la tablatura.	

Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	1. El usuario puede colocar o quitar notas de un ritmo para crear uno nuevo.	2. Se aplican algoritmos para eliminar o agregar notas.
Sub-flujos	Paso 1: Si el usuario no agrega o elimina notas no se hacen modificaciones.	
Poscondición(es)	Se muestra el nuevo ritmo que ha creado el usuario por medio de las modificaciones.	

CASOS DE USO 6		
Caso de Uso	Agregar notas a la tablatura	
Actor(es)	Usuario	
Propósito	Que el usuario haga uso de las notas que ofrece el sistema y las coloque de forma correcta en la tablatura respetando el tiempo de duración.	
Descripción	Este caso de uso inicia cuando el usuario agrega por lo menos una nota a la tablatura.	
Precondición(es)	Que el usuario coloque 1 o más notas en el pentagrama.	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	1. El usuario hace clic sobre una de las notas. 3. Coloca mas notas en el pentagrama.	2 .Deshabilita la nota que se ha seleccionado y las dibuja en la tablatura. 4. Dibuja las notas seleccionadas por el usuario y emite un sonido.
Sub-flujos	Paso 1: Si el usuario no agrega ninguna nota a la tablatura.	
Poscondición(es)	Se muestra la tablatura con las notas que se han seleccionado.	

CASOS DE USO 7		
Caso de Uso	Modificar notas	
Actor(es)	Usuario	
Propósito	Que el usuario pueda quitar o reemplazar notas que haya colocado.	
Descripción	Este caso de uso es el que le permite al usuario cambiar de posición o eliminar notas de la tablatura.	
Precondición(es)	Que haya colocado notas en la tablatura	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	1. El usuario puede colocar o quitar notas de un ritmo haciendo los cambios que desee.	2. Se aplican algoritmos para eliminar o agregar notas.
Sub-flujos	Paso 1: Si el usuario no agrega o elimina notas no se hacen modificaciones.	
Poscondición(es)	Se muestra la tablatura con las nuevas notas modificadas por el usuario.	

CASOS DE USO 8		
Caso de Uso	Guardar un ritmo	
Actor(es)	Usuario	
Propósito	Que el usuario tenga la posibilidad de archivar composiciones que ha hecho con el fin de crear sus propios ritmos.	
Descripción	Este caso de uso permite que el usuario almacene los datos con una extensión y ubicación determinada por el sistema.	
Precondición(es)	Que el usuario haya creado o modificado algún ritmo.	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	1. El usuario crea una composición o carga una de las que aparecen por defecto.	3. Se muestra una ventana que solicita el nombre del

	2. El usuario oprime el botón "guardar"	ritmo y la ubicación
Sub-flujos	Paso 1: Si el usuario no realiza modificaciones o crea un nuevo ritmo.	
Poscondición(es)	Se actualiza el sistema con los nuevos ritmos que se han creado.	

CASOS DE USO 9		
Caso de Uso	Agregar nuevo compás	
Actor(es)	Usuario	
Propósito	Que el usuario pueda crear no sólo un ritmo sino también un tema completo añadiendo compases.	
Descripción	Este caso de uso inicia cuando el usuario después de haber creado un compás quiere agregar otro más para formar temas completos.	
Precondición(es)	Que se haya creado el primer compás.	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	1. El usuario completa un compás y oprime el botón lateral derecho "agregar compás".	2. Aparece un nuevo compás totalmente vacío.
Sub-flujos	Paso 1: Si el usuario no oprime el botón de agregar un nuevo compás.	
Poscondición(es)	Indica que se ha creado un tema.	

CASOS DE USO 10	
Caso de Uso	Cambiar el tempo
Actor(es)	Usuario
Propósito	Que el usuario pueda aumentar o disminuir la velocidad de reproducción de un ritmo.

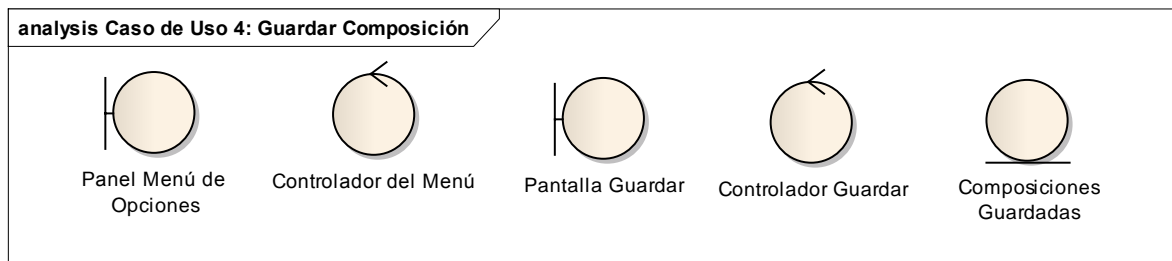
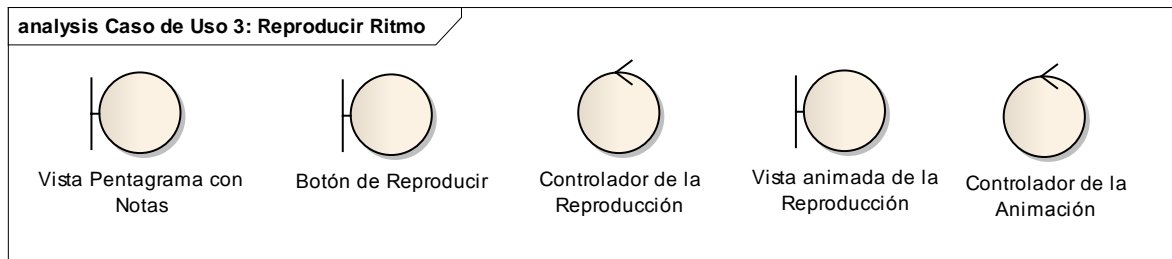
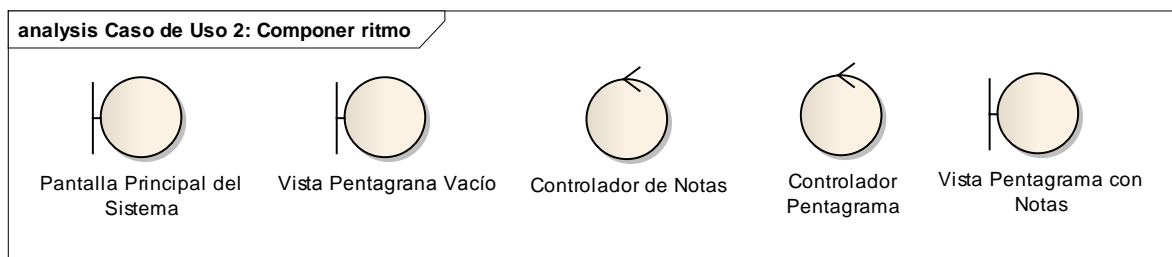
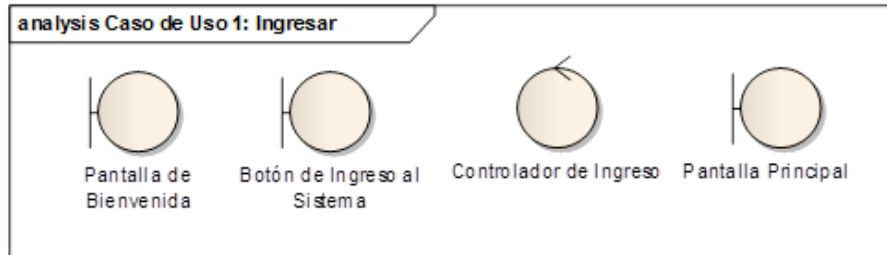
Descripción	Este caso de uso le permite al usuario escuchar el ritmo más rápido o más lento según el tempo que seleccione.	
Precondición(es)	Que se cambie el tempo que trae el sistema en su configuración.	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	1. El usuario seleccione un nuevo valor para el tempo.	2. Se va a reproducir el ritmo a una velocidad diferente, puede ser más rápido o más despacio según el tempo que se ha escogido
Sub-flujos	Paso 1: Si el usuario no cambia la configuración inicial del sistema.	
Poscondición(es)	Reproducir un ritmo con velocidad diferente.	

CASOS DE USO 11		
Caso de Uso	Cambiar el volumen	
Actor(es)	Usuario	
Propósito	Que el usuario pueda aumentar o disminuir la intensidad de reproducción de un ritmo.	
Descripción	Este caso de uso le permite al usuario escuchar el ritmo a una intensidad mayor o menor según el grado de volumen que seleccione.	
Precondición(es)	Que se cambie el valor del volumen que trae el sistema en su configuración.	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	1. El usuario seleccione un nuevo valor para el volumen.	2. Se va a reproducir el ritmo a con una intensidad diferente, puede ser más alta o más baja según el que se ha escogido.

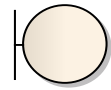
Sub-flujos	Paso 1: Si el usuario no cambia la configuración inicial del sistema.
Poscondición(es)	Reproducir un ritmo con a un volumen más alto o más bajo.

CASOS DE USO 12		
Caso de Uso	Cancelar/Cambiar algún instrumento	
Actor(es)	Usuario	
Propósito	Que el usuario elija si quiere que suenen los 4 instrumentos que conforman el sistema o si quiere cambiar alguno de los que están ahí por otro.	
Descripción	Este caso de uso es el que le permite al usuario armar el set de batería de la forma que él quiera, ya sea desactivando alguno de los instrumentos o sustituyendo unos por otros.	
Precondición(es)	Que para la reproducción este seleccionado por lo menos uno de los instrumentos.	
Flujo Principal	Acciones de Actor(es)	Respuestas del Sistema
	<ol style="list-style-type: none"> 1. El usuario se ubica sobre cualquier instrumento y hace clic izquierdo. 3. El usuario hace clic derecho sobre algún instrumento. 5. El usuario selecciona alguno de los instrumentos de la lista. 	<ol style="list-style-type: none"> 2. El instrumento cambia de color mostrando que en ese momento esta desactivado, además esa línea de la tablatura también se desactiva. 4. Aparece un menú con los nombres de todos los instrumentos disponibles. 6. Se cambia el instrumento que estaba por el nuevo que escogió el usuario.
Sub-flujos	Paso 1: Si el usuario no hace ningún cambio de instrumento y deja activados los 4 que muestra el sistema.	
Poscondición(es)	Personalizar el set de instrumentos de acuerdo a la selección del usuario.	

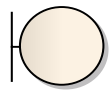
Clases (Entidad+Borde+Control) para cada caso de uso:



analysis Caso de Uso 5: Escoger Ritmo



Panel Controles



Menú Ritmos Por Defecto



Controlador Ritmos



Ritmos

analysis Caso de Uso 3: Agregar un nuevo compás



Pantalla Principal del Sistema



Vista Pentagrama Compás 1



Controlador Pentagrama



Botón Agregar Nuevo Compás

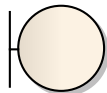


Controlador Agregar Compás

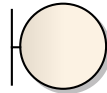


Vista con Nuevo Compás

analysis Caso de Uso 6: Seleccionar Instrumento



Panel Instrumentos



Vista Menú Instrumentos



Controlador Instrumentos

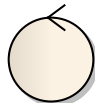


Instrumentos

analysis Caso de Uso 7: Cambiar el Tempo



Panel Controles



Controlador Controles



Vista Menú Tempo



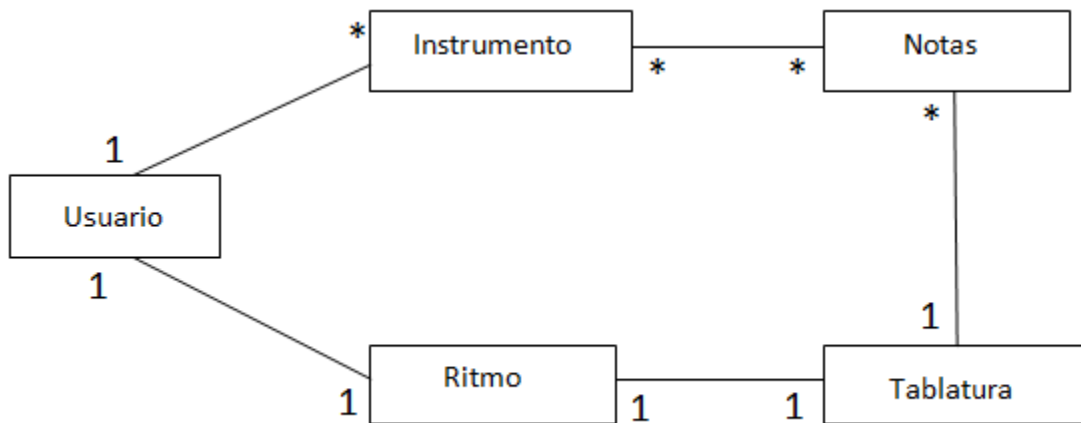
Tempos Guardados

Para identificar los objetos principales del sistema fue necesario hacer un proceso de abstracción en el que además de identificarlos y definir sus propiedades se dedujo la relación de estos con otros elementos.

Identificación de las clases y sus atributos: Para esto se hace una descripción del sistema resaltando los nombres o sustantivos que aparecen. Estos sustantivos se convierten en clases más adelante.

COMBAT es un software que tiene como objetivo motivar a los niños a entrenarse en la lectura musical de un instrumento de percusión. Busca enseñar de forma didáctica la forma en la que se ubican las notas musicales en una tablatura para que se puedan generar ritmos y de esta forma aprendan a leer canciones por medio de figuras musicales. El software permite que se agreguen más compases para que se pueda crear no sólo un ritmo sino también un tema completo. Está diseñado para que lo use el usuario solo o con apoyo de un tutor.

Figura 6. Diagrama de clases



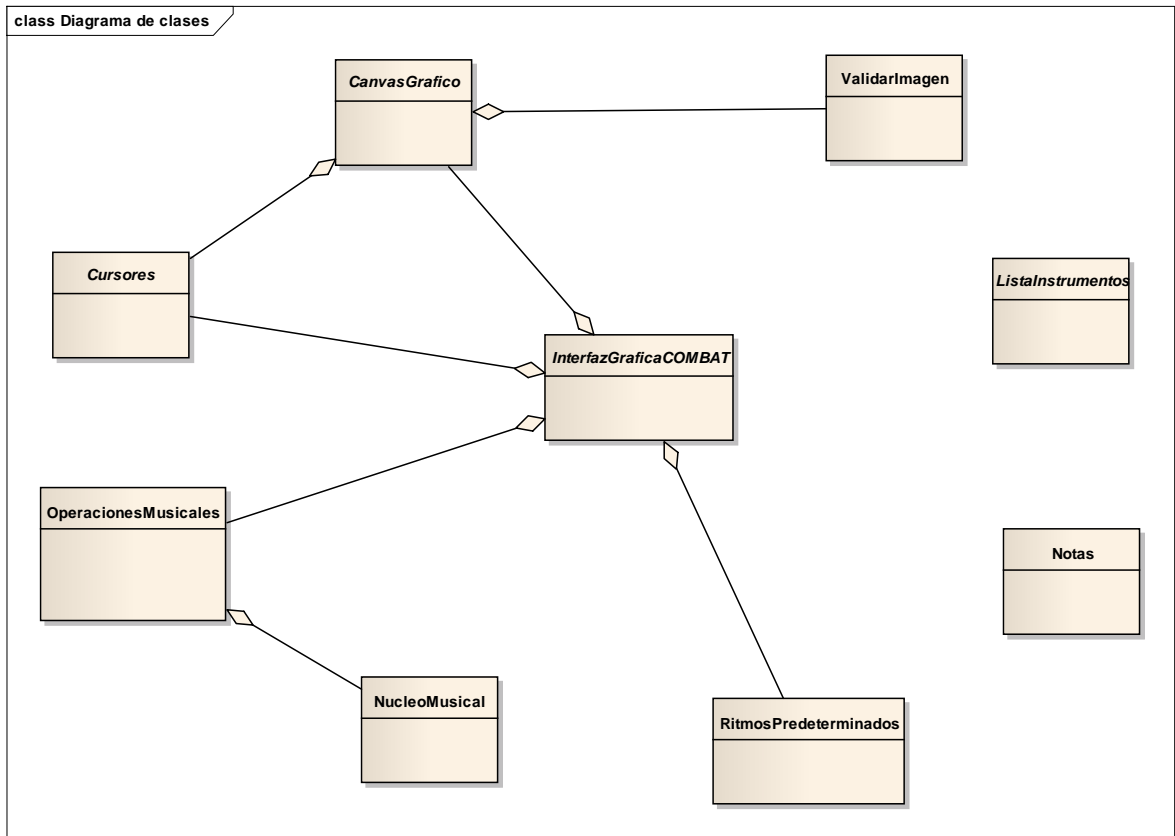
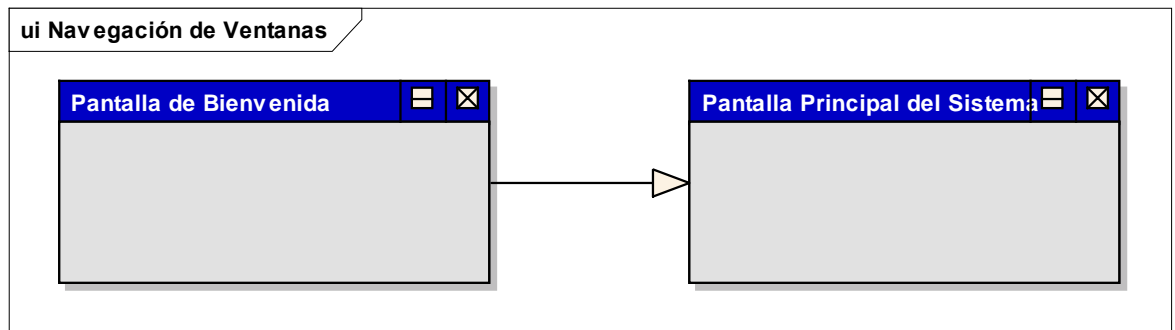


Figura 7. Diagrama de navegación de ventanas:



Etapa 3: Desarrollo de MECs

Esta es la fase de implementación, en esta fase se reúne toda la información de las fases anteriores para implementarla en la herramienta de desarrollo que se ha elegido; en este caso Java. Desde que se empezó a hacer el análisis se tuvieron en cuenta recursos humanos, computacionales y temporales para que una vez

iniciada esta fase de implementación se hiciera uso de esos recursos con el fin de cumplir los objetivos del proyecto en el tiempo determinado y satisfaciendo las necesidades planteadas.

Estrategia para el desarrollo de MECs:

Para el buen desarrollo del proyecto se contó con la asistencia de un estudiante de música de la UNAB de 9° semestre con conocimiento en instrumentos de percusión especialmente de batería. Él explicó el fundamento teórico de cómo tocar batería, y de esta forma mediante procesos matemáticos se llegó a una forma pedagógica para poder plasmar ese fundamento de una manera más sencilla. Después de tener comprendida la parte teórica se procedió al desarrollo; esta parte muy ligada a reuniones periódicas con el director del proyecto para realizar consultas acerca de detalles que iban surgiendo a medida que se desarrollaba la aplicación. A medida que se desarrollaba el módulo el director lo revisaba para ajustar detalles por si se requerían.

Con esta estrategia de desarrollo se logró que el proyecto mantuviera siempre su esencia, y no sufriera cambios que lo alejarán del objetivo principal. Los cambios que surgían en cada reunión contaban con la aprobación del especialista en música para garantizar que dichos cambios fueran positivos para el proyecto y por supuesto de ayuda para los usuarios finales del sistema.

Desarrollo y Documentación del MECs:

Documentar el software es en algunas ocasiones una labor tediosa comparada con la creatividad que existe al desarrollarlo. Algunos ven la documentación como algo que no aporta ningún beneficio.

Gracias a la aparición de las metodologías ágiles se hizo que documentar fuera uno de los principios del manifiesto ágil indicando que el cumplimiento de esos principios haría que se valorará más el software. Debido a esto algunas organizaciones han adoptado estos principios a la hora de desarrollar software con el fin de tener control sobre el sistema que se está desarrollando¹².

Por estas razones en este proyecto se hizo documentación no exhaustiva pero que permite entender el fragmento de código que se está analizando sirviendo de ayuda en caso de que haya que hacer modificaciones. También se programó en forma estructurada y legible usando el estándar lowerCamelCase que dice que la primera letra es en minúscula y cuando haya cambio de palabra se coloca en mayúscula, esta se usó para nombres de objetos y de las variables, mientras que para nombrar las clases se utilizó UpperCamelCase que dice que la primera letra de cada una de las palabras es mayúscula y cuando haya cambio de palabra se coloca también en mayúscula.

Pensando un poco en racionalizar el esfuerzo de la programación, fue conveniente antes de iniciar la codificación identificar procedimientos de utilidad común y mirar si existían librerías aplicables. En este proyecto en particular se tuvieron en cuenta métodos para hacer animaciones, crear sonidos, simular el funcionamiento de una tablatura, entre otros.

Una aplicación multimedia es un sistema que utiliza múltiples medios de expresión físicos o digitales para presentar y comunicar información. Además se habla de multimedia interactiva cuando el usuario tiene libre control sobre la presentación de los contenidos, acerca de qué es lo que desea ver y en qué momento verlo. Esta aplicación es considerada como una aplicación multimedia interactiva. Para la realización de este tipo de aplicaciones se sugiere contar con un equipo interdisciplinar en el que participen al menos tres profesionales: el experto en el

¹²Referenciado de <http://www.fabricasdesoftware.es/importancia-documentacion.html>

contenido, el experto en el diseño y el técnico programador. Fue por esta razón que para la elaboración de los prototipos se conformó un equipo en el que participó una persona con buenos conocimientos musicales, el director del proyecto como diseñador y el desarrollador del proyecto encargado de la implementación.

Pensando en que se iba a realizar era un software educativo y didáctico, se debía construir una interfaz que le facilitara el trabajo al usuario. Para esto el factor más importante a tener en cuenta era la usabilidad. La usabilidad es una cualidad demasiado abstracta como para ser medida directamente, así que para poder ilustrarla se estudian una serie de factores que miden el grado de usabilidad de la herramienta. Los factores son los siguientes:

- **Facilidad de aprendizaje:** Cuán fácil es aprender la funcionalidad básica del sistema, como para ser capaz de realizar correctamente la tarea que desea realizar el usuario.
- **Eficiencia:** El número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema.
- **Recuerdo en el tiempo:** Para usuarios intermitentes (que no utilizan el sistema regularmente) es vital ser capaces de usar el sistema sin tener que aprender cómo funciona partiendo de cero cada vez.
- **Tasa de errores:** Este atributo contribuye de forma negativa a la usabilidad de un sistema. Se refiere al número de errores cometidos por el usuario mientras realiza una determinada tarea
- **Satisfacción:** Éste es el atributo más subjetivo. Muestra la impresión subjetiva que el usuario obtiene del sistema.

Estos cinco atributos pueden descomponerse a su vez para conseguir una mayor precisión en los aspectos de usabilidad en los que se quiera poner mayor énfasis.

La usabilidad es un concepto muy importante en los sistemas interactivos. Es una cualidad centrada en el usuario, busca ayudarlo a lograr los objetivos específicos con eficiencia y satisfacción. No solo tiene que ver con aspectos de diseño, sino también con la forma en la que se haga la interacción con el usuario, estas interacciones se aplican en el código en el que se implementa la aplicación.

Es importante invertir tiempo antes de iniciar el desarrollo de la aplicación pensando en cómo hacer para que tenga buena usabilidad, porque se pueden presentar errores graves en un momento cercano al finalizar el proyecto ocasionando un aumento prolongado del tiempo de desarrollo para poder corregir dichos problemas.

Para lograr que la interfaz sea intuitiva se debe diferenciar entre el diseño funcional y el diseño visual. El diseño funcional es la forma como se interpretan los requerimientos del cliente¹³ y el diseño visual es el que transforma los datos en estructuras visuales mediante el conocimiento de procesos perceptivos y cognitivos¹⁴. Cuando el diseño visual tiene claras cuáles son sus funciones la interfaz se vuelve intuitiva. A pesar de que lograr la funcionalidad de este software fue una tarea compleja, siempre se le dio prioridad a la usabilidad manteniendo la simplicidad de la pantalla.

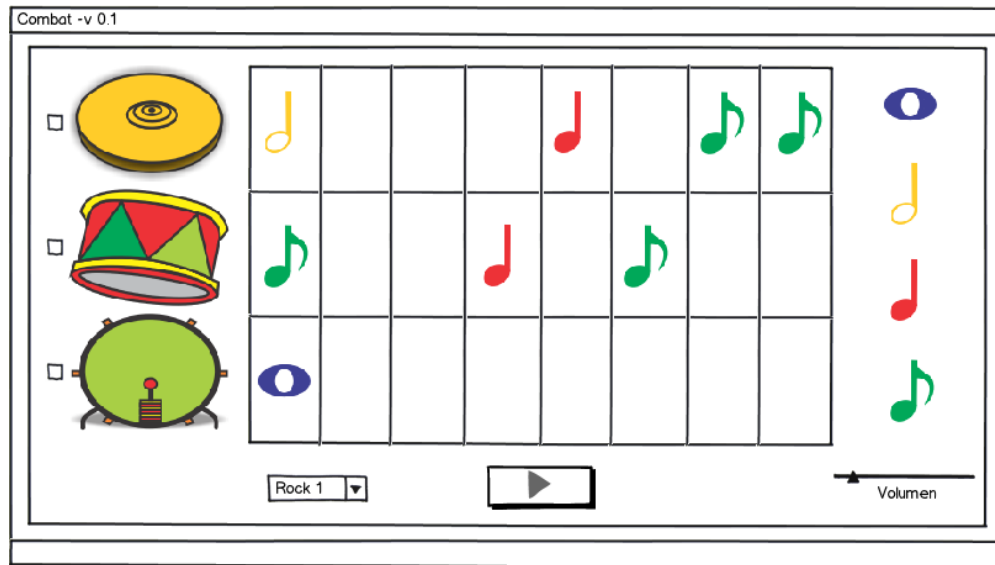
Desarrollo evolutivo por prototipos:

El uso de prototipos se basa en comprender los requerimientos que se deben tener en cuenta al desarrollar software. Generalmente se hace una primera versión con funcionalidad reducida que va evolucionando hasta llegar al sistema final que reúne todas las especificaciones del sistema.

¹³ Para mayor información revisar http://www.docirs.cl/que_es_diseno_funcional.htm

¹⁴ Para mayor información revisar http://www.disenovisual.com/static/disenovisual_que_es.htm

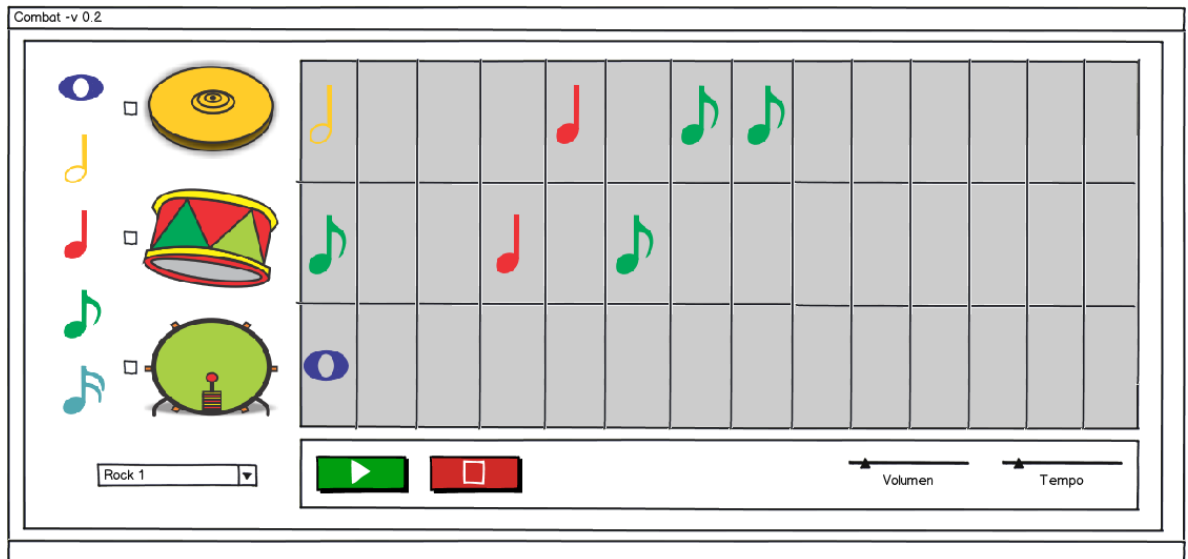
Figura 8. Primer prototipo:



Esta primera versión a pesar de ser muy sencilla busca ser eficiente, manteniendo la simplicidad de la interfaz logrando que el número de transacciones para realizar una actividad sea mínimo.

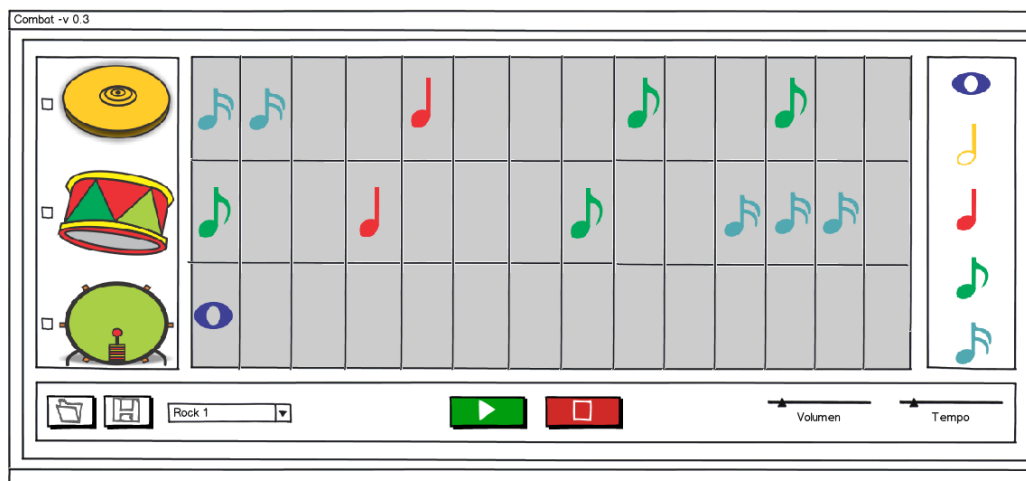
La aplicación contempla una tablatura como la parte fundamental del proyecto, tres instrumentos en el set de la batería, controles como: tempo, play/pause, volumen y algunos ritmos predeterminados.

Figura 9. Segundo prototipo:



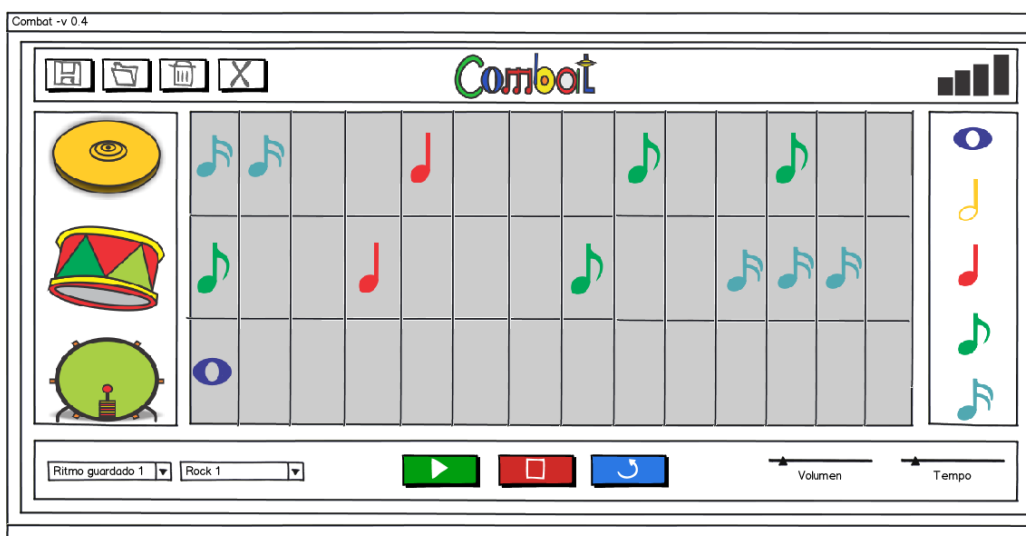
Las notas musicales que existen son: redonda, blanca, negra, corchea, semicorchea, fusa, semifusa, garrapatea y semigarrapatea, se consideran las cinco primeras como las más conocidas y las dos últimas quedaron obsoletas por falta de uso. En un principio el software iba a tener las figuras musicales hasta la corchea para que la velocidad de los ritmos fuera mayor. Se decidió incluir hasta la semicorchea para mantener el estándar de las figuras de uso más frecuente. La fusa y semifusa no se agregaron para evitar una cantidad exagerada de notas. Para mantener la velocidad de los ritmos se agregó el tempo, de modo que si el usuario quiere, puede reemplazar figuras de menor valor por otras solo aumentándolo.

Figura 10. Tercer prototipo:



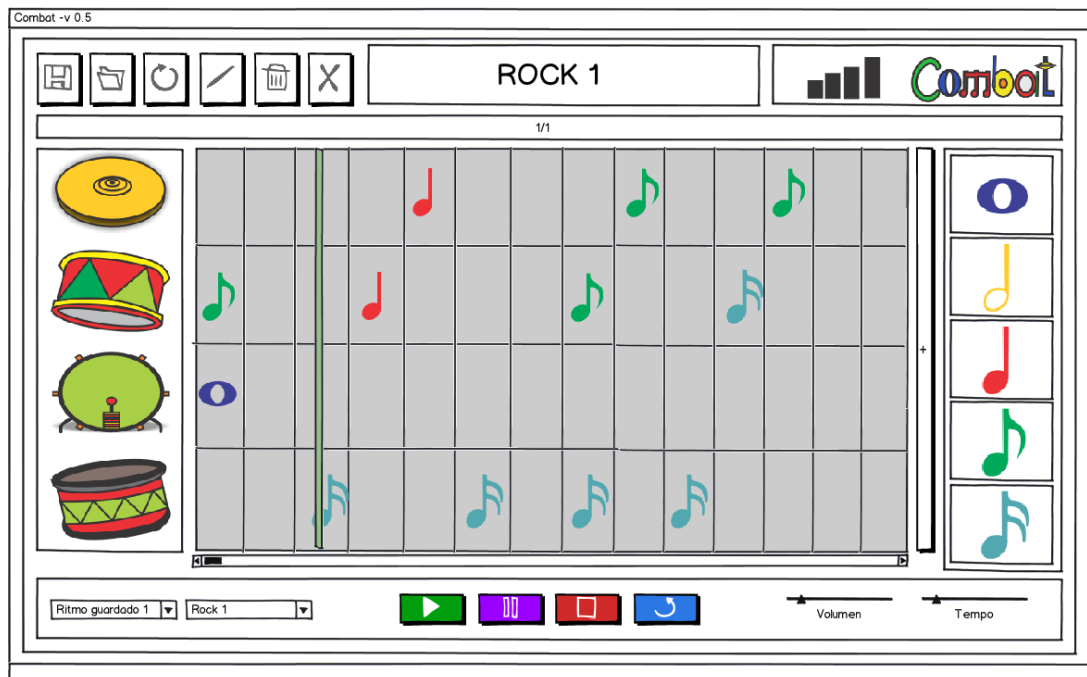
La alineación de los objetos de una pantalla ayuda a conseguir equilibrio, armonía y modularidad, una buena alineación contribuye con la estética de la interfaz. La distribución del espacio es un factor importante porque evita que la información se concentre en una sola parte de la pantalla. Teniendo en cuenta estas técnicas se decidió cambiar de lugar las notas, para que la tablatura quedara ubicada en el centro de la pantalla y se alineara con el panel de las notas y el panel de los instrumentos. Se agregaron dos funciones importantes: el abrir y el guardar, para darle la opción al usuario de archivar composiciones que haya creado.

Figura 11. Cuarto prototipo:



La cancelación de los instrumentos se iba a realizar por medio de “checkbox”, pero para hacerlo de forma más intuitiva y agradable se optó por usar eventos que ofrece el mouse en java. Así que cuando el usuario haga un clic izquierdo sobre algún instrumento este se cancelará y a su vez cancelará toda la fila impidiendo que se coloquen notas mientras se encuentre en ese estado. Es bueno tener separados los paneles de acuerdo a las funciones que tenga cada uno, por ejemplo en el panel de controles (volumen, play, stop y tempo) no deberían aparecer funciones como abrir y guardar. Por esto se creó un panel superior al que además de agregarle otras funciones como eliminar, limpiar tablatura y cerrar también se le agregó una gráfica que mide el volumen junto con el logo para darle identidad a la aplicación. Surgieron nuevos detalles importantes como crear un nuevo “combobox” en el panel de los controles con el fin de guardar en uno de ellos los ritmos y en el otro los temas, ya sean los predeterminados o los creados por el usuario. Además aparece una nueva función y es el “loop”, esta permite repetir un ritmo o tema las veces que el usuario lo desee.

Figura 12. Quinto prototipo:



Pensando en que el usuario tenga más posibilidades a la hora de componer un ritmo se agregó un nuevo instrumento. Dentro de las funciones de la aplicación ya se encontraba abrir, guardar, eliminar y cerrar. Pero se implementaron dos nuevas; limpiar o refrescar y un botón para hacer cambio de cursor. En el panel de los controles se adicionó un nuevo botón para pausar un ritmo y poder reanudarlo fácilmente. En este último prototipo surgió un aporte importante y fue la posibilidad de construir no solo un ritmo sino todos los que el usuario quiera agregar convirtiendo la unión de ritmos (elaborados en un solo compás) a un tema (elaborados en varios compases). Este valor agregado convirtió al software en una herramienta completa con la que el usuario puede componer canciones y guardarlas para después modificarlas y construir otras nuevas. Como es un software musical no podía faltar un identificador llamado línea de progreso que indica en que parte va la reproducción en determinado momento. A partir de estos nuevos cambios se llegó a la quinta y última versión del software.

Después de elaborar varios prototipos se había obtenido uno que cumplía con el objetivo del proyecto que era diseñar un software con una interfaz comunicacional agradable en la cual el usuario interactuara con la máquina de forma sencilla.

Hasta ahora los prototipos se habían diseñado solo en blanco y negro, faltaba decidir que combinación de colores utilizar para hacerla más atractiva a los niños.

El uso efectivo del color es un tema amplio con muchas características importantes para tener en cuenta. Expertos en cromoterapia recomiendan el color amarillo en tonos pasteles y alternado con otros colores porque favorece la concentración y el desarrollo intelectual. Mencionan que en ambientes de estudio es importante tener en cuenta colores frescos (azul, verde, o combinación entre ellos), poco saturados porque ayudan a fijar la concentración debido a que transmiten tranquilidad. Los colores usados para el diseño de la pantalla son los siguientes:

Rojo: Este color estimula la acción, se recomienda usarlo en juguetes que busquen impulsar la acción. Genera atracción visual.

Azul: Es un color que se utiliza para calmar a las persona, debido a que produce paz.

Amarillo: Este color estimula la actividad mental, en algunos casos se usa en ambientes en donde trabajan niños con poca concentración o dificultades de aprendizaje. Inspira energía y optimismo.

Verde: Es un color que hace que todo sea fluido. Produce armonía y es también llamado el color de la nueva creación.

Estos cuatro colores además de combinarse muy bien dan un buen contraste. También se usan para vincular significados prácticos e intuitivos, que son fáciles de aprender y recordar¹⁵.

El color es un componente principal de las GUI (Interfaz Gráfica de Usuario) porque mejora la estética de la interfaz, aumentando la efectividad del procesamiento de la información y el rendimiento de la memoria. Se recomienda asignarle un color diferente a cada concepto que se quiera introducir para mantener la consistencia de la pantalla. Con base en esto se le asignó un color a cada panel ya que cada uno contiene información diferente. El panel central izquierdo es el que más colores tiene debido a la variedad de instrumentos, algunos de estos tienen sombra para dar un efecto más real a la imagen. Los colores de los instrumentos son combinaciones de los 4 colores base de la pantalla teniendo en cuenta el contraste con el fondo amarillo que los contiene. El panel central derecho o panel de las notas no tiene un fondo establecido, se le asignó a cada nota un color diferente para lograr diferenciarlas y que fueran

¹⁵ Para ampliar información visitar <http://www.cosasdelainfancia.com/biblioteca-compor12.htm>

llamativas para el usuario. El panel inferior es el de los controles, para resaltar los botones se usó de fondo el color rojo para estimular la acción en los niños. Y finalmente el panel superior que muestra el título del ritmo que se está reproduciendo. La combinación fondo azul y letra en color amarillo se considera atractiva.

Además de los colores los íconos también son de gran importancia en la elaboración de interfaces. Un ícono es la representación de cualquier cosa (comando, proceso o algún indicador) que los usuarios quieran ver¹⁶. Se utilizan porque las personas reaccionan intuitivamente a las imágenes y además por el espacio limitado de la pantalla. En esta aplicación se utilizaron los íconos similares según la clasificación representativa que estos tienen. Los íconos similares son aquellos que presentan el concepto de algo a través de una imagen semejante. Es bueno hacer uso de íconos que ya son conocidos y le puedan facilitar el reconocimiento de acciones al usuario. Los íconos que se usaron son sencillos y consistentes pero al mismo tiempo le dan un estilo particular a la aplicación.

Implementación del sistema:

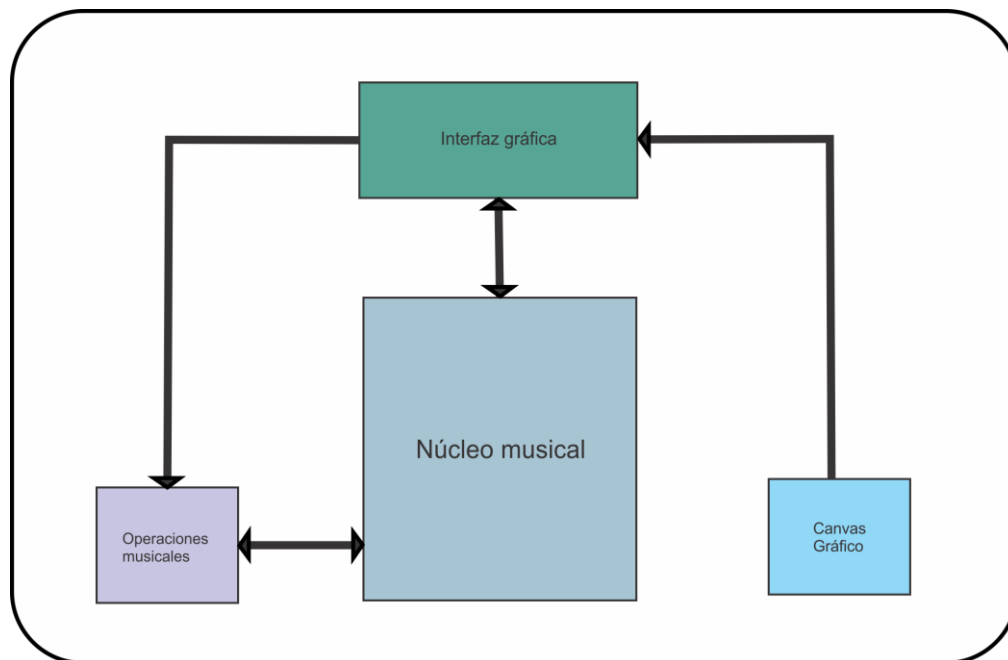
El sistema está conformado por 4 componentes principales:

1. Núcleo musical.
2. Operaciones musicales.
3. Canvas gráfico.
4. Interfaz gráfica.

El núcleo musical es el encargado de hacer todas las operaciones de bajo nivel para que se emitan los sonidos de los instrumentos con las condiciones que el usuario determine. A continuación se muestra un gráfico de su funcionamiento:

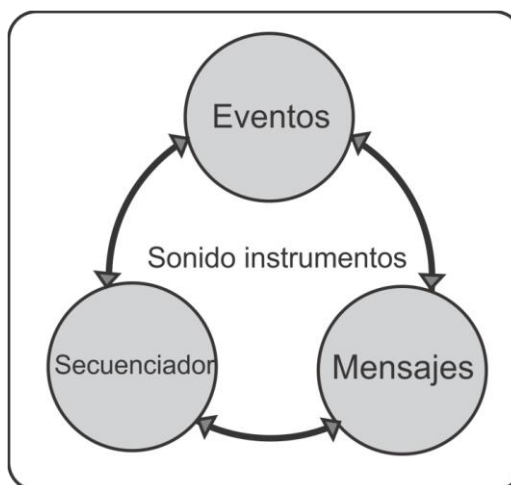
¹⁶ Tomado de http://es.wikipedia.org/wiki/Icono_%28inform%C3%A1tica%29

Figura 13. Funcionamiento de núcleo del sistema



El núcleo de la aplicación combat está basado en el API MIDI de java que contiene una completa colección de instrumentos. Este a su vez está formado de tres partes fundamentales como se muestra en la figura:

Figura 14. Generación de sonidos



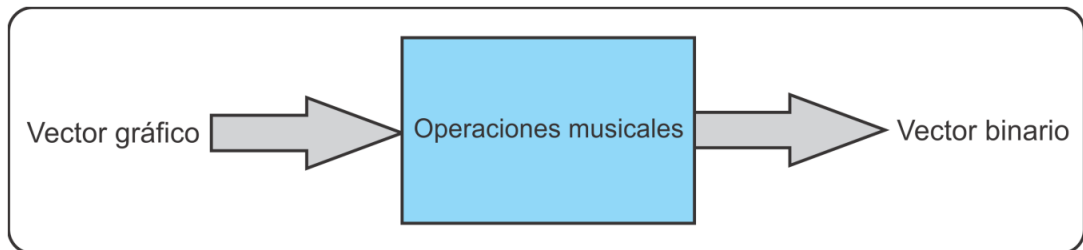
- **Eventos:** son los encargados de ejecutar las órdenes con determinadas condiciones e interacciones del usuario. La asignación de la tablatura (matriz

que contiene las notas a reproducir), el canal y la posición se hace mediante el evento crear.

- Mensajes: Asignan los parámetros de ejecución al secuenciador. Mediante instrucciones de primer nivel como: el indicador de la nota, el canal de sonido y el volumen se va a crear un evento MIDI.
- Secuenciador: Actúa como un traductor; transforma los datos numéricos a sonidos de instrumentos. Al secuenciador se le asigna parámetros como el tempo y la posición inicial.

La clase operaciones musicales contiene parte de la lógica matemática empleada en la aplicación. Para que el núcleo musical pueda entender la tablatura creada en la interfaz gráfica es necesaria una clase que haga las veces de traductor entre estos dos objetos. Utilizando conceptos matemáticos tales como: suma de matrices, formulas recursivas y análisis numérico se hicieron los cálculos que hicieron posible la transformación de la matriz grafica (con información de la posición y el tipo de nota) a una matriz con datos binarios comprensibles para el núcleo musical. Funciona de la siguiente manera:

Figura 15. Funcionamiento operaciones musicales



El canvas gráfico es el que se encarga de realizar todas las funciones que se hacen en la interfaz. En él se encuentran funciones importantes como: los eventos clic del mouse, la forma en la que se dibuja la tablatura en el panel, métodos con los que se dibujan, eliminan y reemplazan notas en la tablatura, la validación para las posiciones en las que se puede colocar una nota según el tiempo de duración y la cancelación de instrumentos.

La interfaz gráfica es la clase que integra las tres clases principales de la aplicación: núcleo musical, operaciones musicales y canvas gráfico. En esta se realiza el diseño de la aplicación por medio del paquete "swing" que provee facilidades para la construcción de interfaces gráficas de usuario. Para que se puedan realizar funciones como abrir y guardar fue necesario implementar estructuras de datos con propiedades especiales que ayudaran a interpretar los archivos guardados por el usuario. Además de esto se agregaron componentes visuales que contribuyen con el cumplimiento de requerimientos no funcionales tales como: el autoajuste de tamaño y posición de todos los elementos que hacen parte de la interfaz gráfica, una línea que muestra el avance de la reproducción en un instante de tiempo dado. Para el autoajuste se utilizó una medida de proporción dada en porcentaje/unidad en lugar de una medida fija en píxeles. Por otro lado en la línea de progreso se aplicó la fórmula física $x = x_0 + vt + \frac{1}{2}at^2$ donde, $a = 0$ ya que v es constante.

Etapa 4: Prueba de MECs

La metodología de desarrollo que se utilizó permitía ir ejecutando cada uno de los componentes del sistema generado haciendo evaluación y prueba a medida que estos iban funcionando. La evaluación fue hecha por el director del proyecto quien fue el encargado del diseño de los prototipos para verificar que se estuviera desarrollando correctamente y también por el estudiante de música que comparaba la funcionalidad del sistema con situaciones reales para saber si forma en la que se generaba el sonido era la indicada.

Después de que estos aspectos fueron aprobados, se liberó una primera versión del sistema interactivo para evaluarlo con una muestra de la población. Se pretendía a través de dicha prueba verificar que se cumplían las necesidades educativas planteadas y la funcionalidad requerida calificándola por medio de principios de usabilidad.

PRUEBA PILOTO:

Al finalizar el software, se realizó un experimento con un grupo de 10 niños con edades que oscilan entre 8 y 9 años para medir características importantes relacionadas con el uso de la herramienta. Estos criterios a evaluar también permiten medir el funcionamiento del software. Primero se hizo una breve descripción de la aplicación en la que se explicó de forma rápida como desarrollar las siguientes actividades:

Tabla 6. En ritmo:

Identificar el cursor.	Si ☺ No ☹ Por qué?
Abrir un ritmo.	Si ☺ No ☹ Por qué?
Reproducir un ritmo.	Si ☺ No ☹ Por qué?
Detener ritmo.	Si ☺ No ☹ Por qué?
Modificar ritmo.	Si ☺ No ☹ Por qué?
Reproducir nuevo ritmo.	Si ☺ No ☹ Por qué?
Guardar un ritmo.	Si ☺ No ☹ Por qué?
Limpiar partitura.	Si ☺ No ☹ Por qué?

Tabla 7. En compositor:

Ir al panel notas y seleccionar una nota.	Si 😊 No ☹️ Por qué?
Colocar una nota en la tablatura.	Si 😊 No ☹️ Por qué?
Reproducir la nota que se colocó.	Si 😊 No ☹️ Por qué?
Crear un ritmo.	Si 😊 No ☹️ Por qué?
Reproducir un ritmo.	Si 😊 No ☹️ Por qué?
Pausar un ritmo.	Si 😊 No ☹️ Por qué?
Reiniciar ritmo.	Si 😊 No ☹️ Por qué?
Cambiar tempo/volumen	Si 😊 No ☹️ Por qué?
Hacer un loop.	Si 😊 No ☹️ Por qué?
Cambiar/cancelar Instrumento	Si 😊 No ☹️ Por qué?

Estas actividades estuvieron supervisadas para resolver dudas respecto a la aplicación. Al finalizar los usuarios contestaron una encuesta formada por 9 preguntas de tipo abierto y cerrado.

**ENCUESTA ACERCA DEL PRODUCTO COMBAT - COMPOSITOR DE
BATERÍA**

1. ¿Ha utilizado antes algún programa de música?
Sí ☺ No ☹
2. ¿Con la información que ofrece la pantalla es posible realizar las actividades propuestas?
Sí ☺ No ☹
3. ¿Considera que el programa es fácil de usar?
Sí ☺ No ☹
4. ¿Le parece atractivo el diseño (íconos, imágenes y colores) del programa?
Sí ☺ No ☹
5. ¿Las funcionalidades del sistema son fáciles de ubicar (abrir, guardar, limpiar, eliminar)?
Sí ☺ No ☹
6. ¿El sonido de los instrumentos le pareció claro?
Sí ☺ No ☹
7. ¿Qué elementos gráficos le ayudaron a entender que es una herramienta para crear música? Mencíónelos

8. ¿Ocurrió alguna falla mientras estaba realizando las actividades? Mencíónela

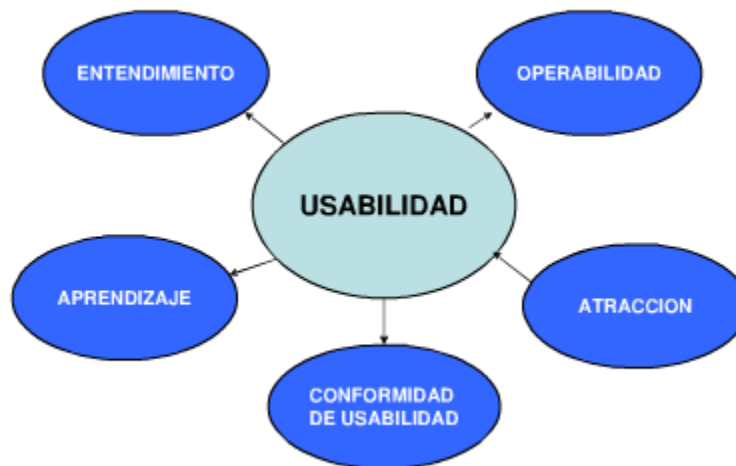
9. ¿Considera que le hace falta algo al programa?
Si ☺ No ☹
Porque? _____

ANÁLISIS DE RESULTADOS

La norma ISO/IEC 9126 es la que se utiliza para la evaluación de calidad de software educativo. Este estándar describe 6 características generales que son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. Para la evaluación de software elaborada se tuvo en cuenta únicamente el criterio de usabilidad debido a que lo que se buscaba medir era el impacto del uso de la herramienta.

La usabilidad es la capacidad del software de ser entendido, aprendido y usado en forma fácil y atractiva. Este criterio se determina por los usuarios finales e indirectos del software, por el uso que ellos le den y los resultados obtenidos.

Figura 16. Características de usabilidad



La usabilidad se define en términos de 5 sub características que se pueden definir así:

- Entendimiento: Es la capacidad del software que le permitirle al usuario comprender como usarlo para realizar alguna actividad.

- Aprendizaje: Es la forma como el software le permite al usuario aprender de su utilización.
- Operabilidad: Es la manera como el software permite que el usuario lo controle.
- Atracción: Es la capacidad que mide que tan atractivo y agradable es el software para el usuario.
- Conformidad de uso: Es la capacidad del software de cumplir estándares relacionados con usabilidad.

Para probar la usabilidad del sistema los niños usaron el programa durante media hora e iban contestando cada una de las preguntas planteadas. Para la parte de la encuesta se hicieron preguntas cerradas con dos posibles respuestas (si o no); y algunas preguntas abiertas con el fin de que los usuarios no solo se limitaran a marcar una opción sin justificar el porque de su respuesta.

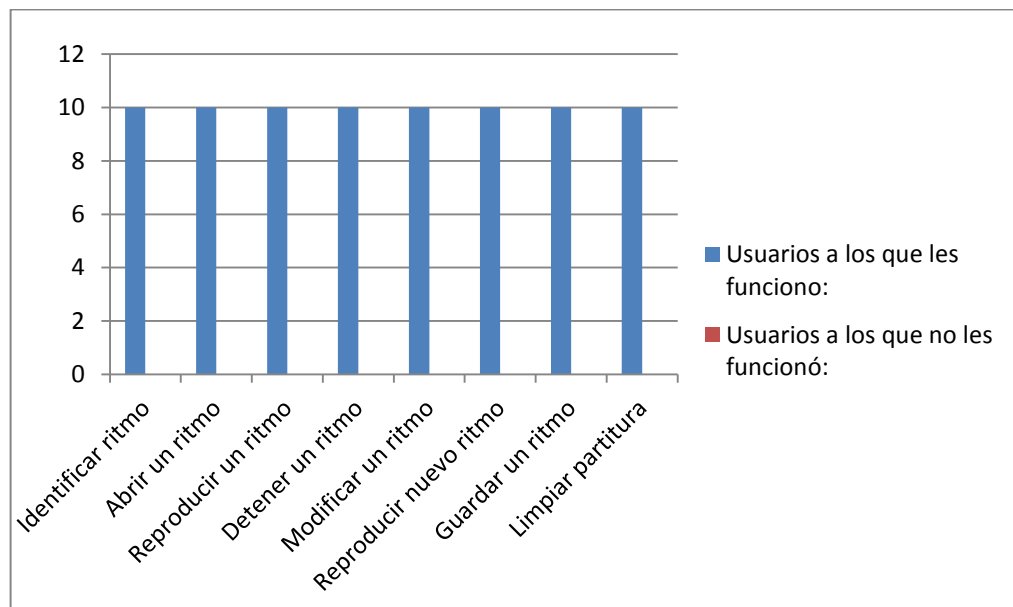
Las pruebas se realizaron en dos grupos de 5 integrantes. Los computadores en los que se instaló la prueba tenían diferentes especificaciones de hardware, lo cual ocasionó que se produjeran errores inesperados al ejecutar la aplicación y por tanto algunos de los usuarios no terminaran las actividades satisfactoriamente. Sin embargo, todos completaron la prueba.

El análisis de los resultados es el siguiente:

Tabla 8. Modo ritmo:

Actividades:	Usuarios a los que les funcionó:	Usuarios a los que no les funcionó:
Identificar ritmo	10	0
Abrir un ritmo	10	0
Reproducir un ritmo	10	0
Detener un ritmo	10	0
Modificar un ritmo	10	0
Reproducir nuevo ritmo	10	0
Guardar un ritmo	10	0
Limpiar partitura	10	0

Figura 17. Análisis modo ritmo



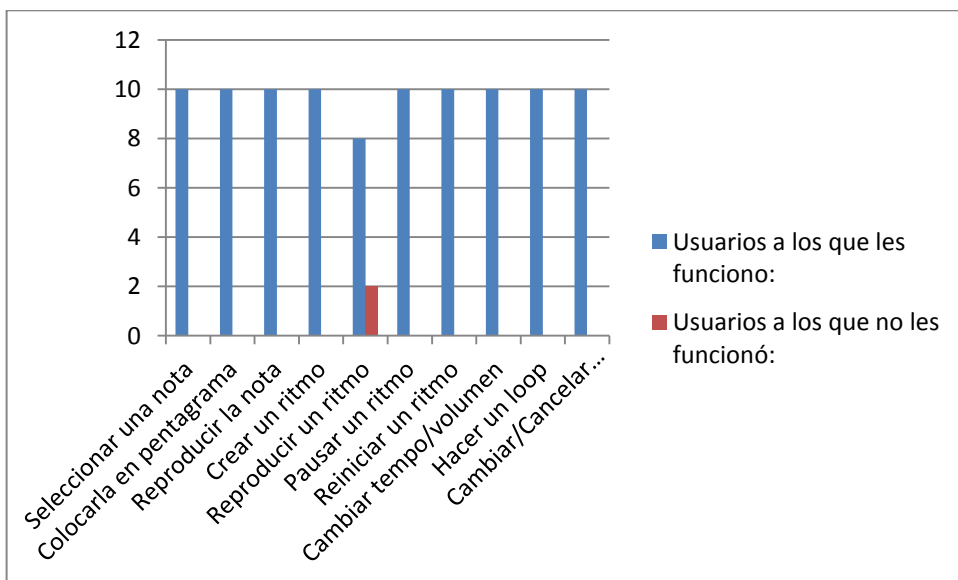
Los resultados obtenidos en esta primera parte de la prueba fueron satisfactorios. Después de la explicación que recibieron los usuarios, demostraron haber comprendido la mecánica de la aplicación realizando cada una de las actividades propuestas. Manifestaron diversas emociones durante el desarrollo de las actividades, algunos se mostraron sorprendidos porque era la primera vez que tenían contacto con una herramienta musical, otros se mostraron interesados en

adquirir la aplicación para poder usarla desde su casa. En general todos estaban satisfechos con las funciones que ofrece la aplicación.

Tabla 9. Modo componer:

Actividades:	Usuarios a los que les funcionó:	Usuarios a los que no les funcionó:
Seleccionar una nota	10	0
Colocarla en la tablatura	10	0
Reproducir la nota	10	0
Crear un ritmo	10	0
Reproducir un ritmo	8	2
Pausar un ritmo	10	0
Reiniciar un ritmo	10	0
Cambiar tempo/volumen	10	0
Hacer un loop	10	0
Cambiar/Cancelar instrumento	10	0

Figura 18. Análisis modo compositor



En esta segunda parte de la prueba los usuarios debían demostrar una vez más que tenían claro el funcionamiento de la aplicación. Existía gran expectativa por el resultado que se podía obtener en la actividad número 4: crear un ritmo. En esta actividad se pedía a los usuarios crear un ritmo haciendo uso de las notas musicales. Al finalizar la prueba y basados en los resultados que muestra la

gráfica se puede concluir que los usuarios comprendieron el tema, realizaron satisfactoriamente las actividades y crearon ritmos respetando los tiempos de duración de cada una de las notas. En la actividad número 5: reproducir un ritmo, se presentó la siguiente situación: en 2 de los equipos en los que se instaló la aplicación no se escuchaba el sonido del ritmo que se había creado. Esta falla se presentó debido a que esos equipos no cumplían con los requisitos que necesita la aplicación para ejecutarse correctamente, porque tenían instalado un sistema operativo desactualizado.

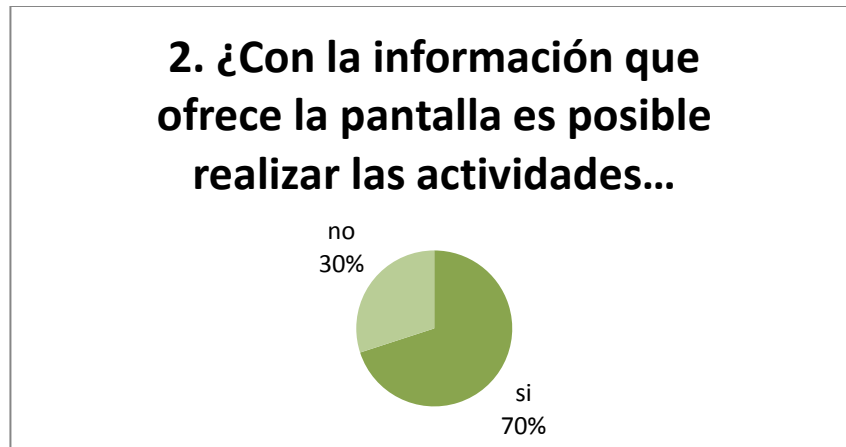
Análisis de los resultados de la encuesta:

Figura No. 19. Pregunta No. 1



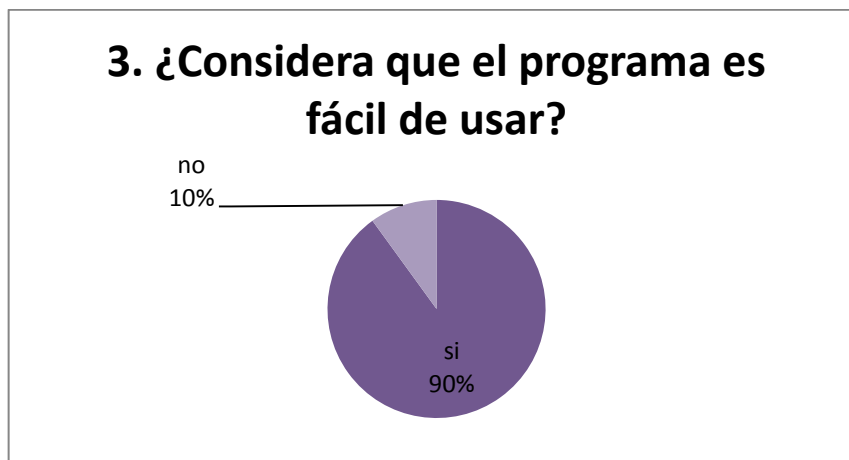
Dentro de la población que se estudió se encontró como patrón común la no experiencia en el uso de software musical. Este hecho se contempló desde la etapa de análisis del proyecto y se esperaba contar con diferentes tipos de usuario para que cualquier persona lo pueda utilizar. Los resultados obtenidos son favorables ya que los usuarios pudieron resolver todas las actividades propuestas sin conocimiento previo de cómo usar herramientas software.

Figura No. 20. Pregunta No. 2



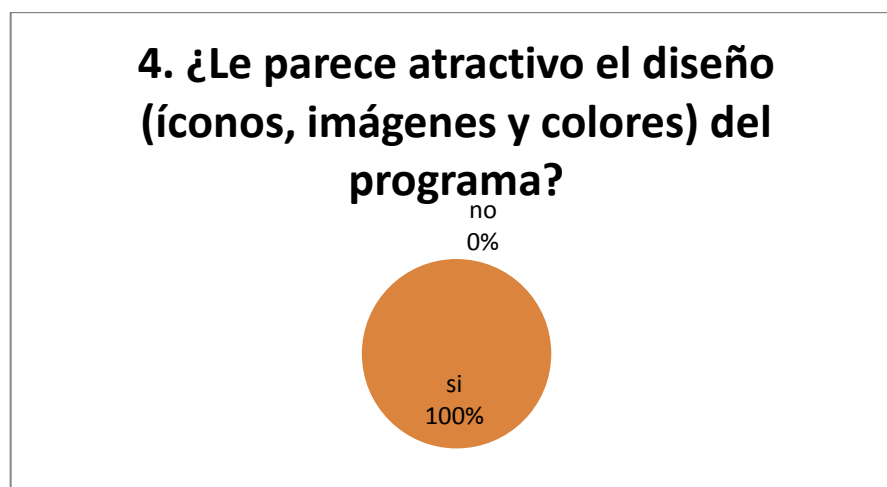
En su mayoría los usuarios a los que se les aplicó la prueba no habían utilizado antes una herramienta con las características de Combat para realizar ejercicios musicales. Sin embargo, un 70% manifestó comprensión y entendimiento de lo que tenía que hacer únicamente con la información que le brindó la pantalla. Si se les hiciera otra prueba muy seguramente recordarían el funcionamiento. Esto indica que la aplicación cumple con una de las características de usabilidad llamada: recuerdo en el tiempo, esta característica dice que los usuarios así no utilicen el sistema regularmente deben ser capaces de usarlo cada vez que lo hagan sin partir de cero.

Figura No. 21. Pregunta No. 3



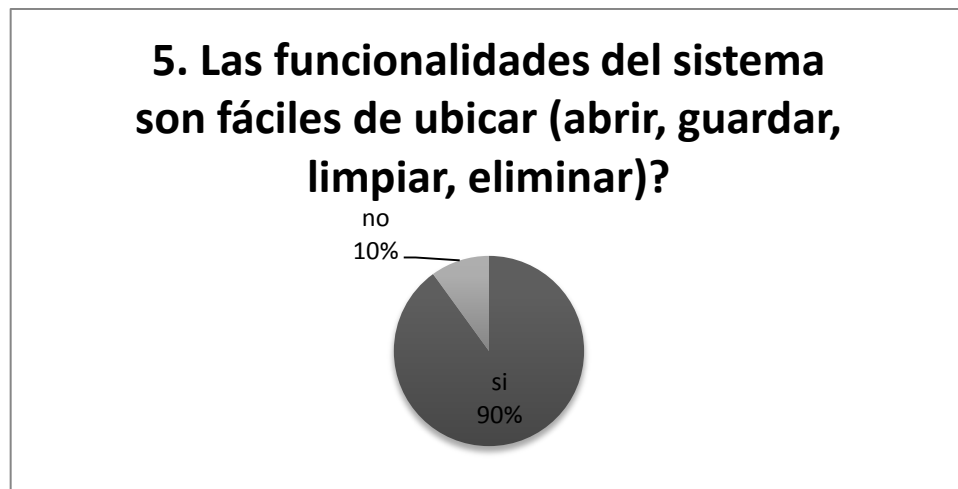
Para un 90% de la población la aplicación fue fácil de usar siendo la primera vez que utilizaban un programa en el que haciendo uso de su creatividad hicieran composiciones musicales. Es por esto que se considera que el software cumple el criterio llamado: facilidad de aprendizaje, porque los usuarios aprendieron la funcionalidad básica del sistema y realizaron correctamente las actividades que se les solicitó.

Figura No. 22. Pregunta No. 4



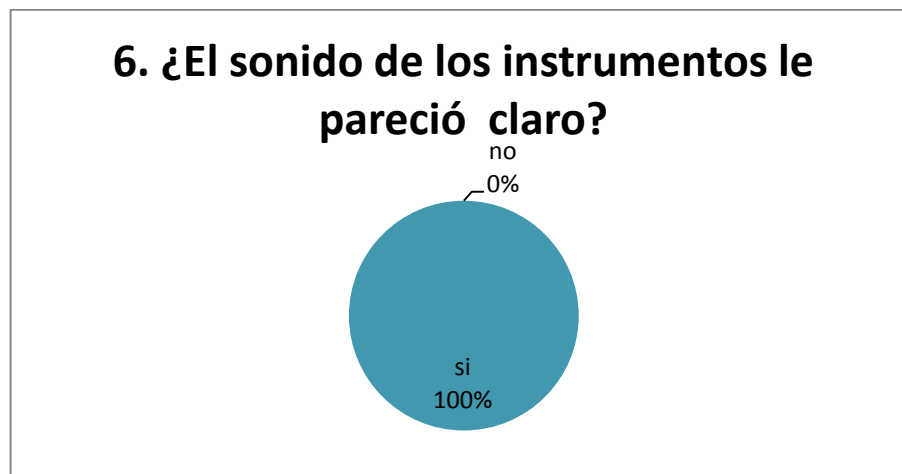
Para todos los usuarios el diseño de la aplicación fue atractivo. Les gustaron las imágenes, los íconos que se utilizaron no generaron confusiones y los colores hicieron que la aplicación fuera llamativa causando en los niños efectos positivos desde que vieron la aplicación por primera vez. Esto demuestra que se elaboró un software con un ambiente agradable para los usuarios a los que va dirigido.

Figura No. 23. Pregunta No. 5



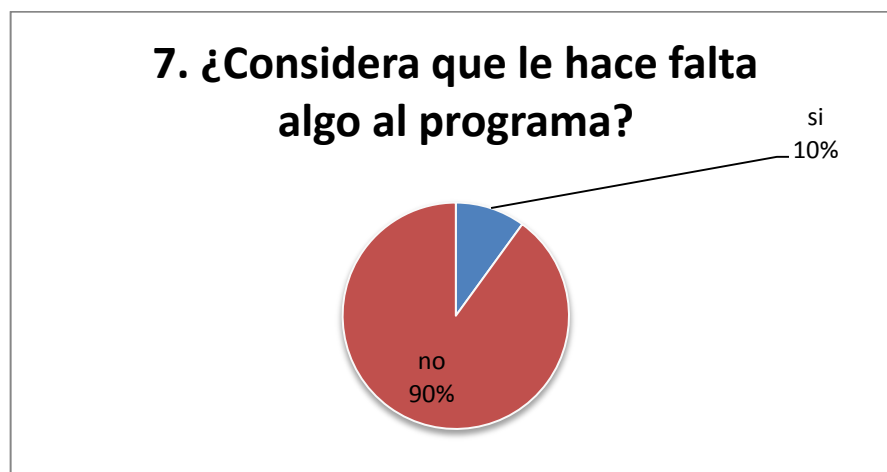
Los íconos fueron ubicado en la esquina superior izquierda para mantener la posición en la que se encuentran en algunas aplicaciones conocidas para ellos tales como: word. La imagen utilizada para cada ícono refleja el significado de la acción que se quiere realizar, por esto a los usuarios se les facilitó encontrar cada herramienta y ubicarla fácilmente.

Figura No. 24. Pregunta No. 6



Uno de los riesgos que se corre al generar sonido por computadora es que estos podrían perder propiedades importantes como la intensidad¹⁷ y el timbre¹⁸ creando confusión a los usuarios respecto al instrumento que se encuentre sonando. Pero todos los usuarios estuvieron de acuerdo con que el sonido de los instrumentos era claro y les permitía diferenciar que instrumentos se estaban reproduciendo en determinado ritmo.

Figura No. 25. Pregunta No. 7



¹⁷ Es la fuerza con la que se produce un sonido. Referenciado de <http://es.wikipedia.org/wiki/M%C3%BAsica>

¹⁸ Es la cualidad que permite distinguir los diferentes instrumentos. Referenciado de <http://es.wikipedia.org/wiki/M%C3%BAsica>

Un 90% de los usuarios considera que al software no le hace falta nada. Esto indica que se logró satisfacer al usuario y que el software cumple con la característica de usabilidad conocida como satisfacción. Los usuarios se mostraron sorprendidos y contentos realizando las actividades.

Tabla No. 10 Observaciones de la prueba piloto

Ítem	Observaciones	Pregunta asociada
Entendimiento	El entorno de la herramienta cuenta con íconos de acceso rápido a funciones y con una buena distribución de paneles que mantiene una organización en la pantalla lo que facilita que el usuario se adapte fácilmente. Un 70% de los usuarios entendió lo que tenía que hacer solo con la información de la pantalla.	¿Considera que el programa es fácil de usar?
Aprendizaje	El entorno es bastante intuitivo y gráfico, esto le permitio al usuario aprender la funcionalidad básica del sistema y realizar las actividades.	¿Considera que el programa es fácil de usar?
Operabilidad	El software cuenta con dos modos: reproducir y componer. En ambos casos el usuario tiene la posibilidad de modificar parámetros guardando dichos cambios como una nueva composición. Todo esta controlado por el usuario.	¿Las funcionalidades del sistema son fáciles de ubicar (abrir, guardar, limpiar, eliminar)?
Atracción	El entorno gráfico es llamativo, los colores, íconos e imágenes hacen que se convierta en una ambiente agradable y a su vez fácil de usar para el usuario. La aplicación ocupa toda la pantalla haciendo que el área de trabajo se vea mas amplia y le genere comodidad a los usuarios.	¿Le parece atractivo el diseño (íconos, imágenes y colores) del programa?

CONCLUSIONES

- La metodología de ingeniería de software educativo (ISE) empleada en el proyecto es ideal para el desarrollo de software con contenido educativo ya que contempla teorías de aprendizaje que ayudan a elaborar productos con fines pedagógicos.
- El software generó en los usuarios un impacto positivo que promueve en ellos aspectos importantes como la motivación y la atención facilitando la creación de una rutina de entrenamiento como parte fundamental en el proceso de aprendizaje de cualquier área.
- En aplicaciones diseñadas para poblaciones infantiles la usabilidad debe tener prioridad sobre la funcionalidad para poder ofrecerle al usuario una herramienta con una interfaz agradable y didáctica que le permita entender su funcionamiento con facilidad.
- La prueba piloto es un importante mecanismo para poder evaluar el desempeño de la aplicación, la usabilidad, la funcionalidad y el cumplimiento de los objetivos pedagógicos del proyecto.

RECOMENDACIONES

- Este proyecto hace parte de un grupo de aplicaciones educativas desarrolladas con el fin de contribuir con el entrenamiento en la lectura musical de un instrumento de percusión. Combat representa la aplicabilidad de conceptos teóricos que combinados con la práctica generan resultados positivos. Se recomienda que este tipo de proyectos de grado deberían ser una opción más frecuente para los estudiantes de ingeniería de sistemas que buscan desempeñarse como profesionales.
- Sobre la presente versión de Combat se puede construir una nueva versión orientada a la web que le permita al usuario compartir con otros usuarios del sistema las composiciones que han creado haciendo que la aplicación sea interactiva.
- Un campo de ampliación del presente proyecto es la adición de instrumentos musicales que no son de percusión convirtiendo al software en una herramienta más completa en lo que a enseñanza musical se refiere.

BIBLIOGRAFÍA

1. GALVIS, A.H (1.997). Micromundos lúdicos interactivos: Aspectos críticos en su diseño y desarrollo. *Informática Educativa*, **10** (2), pp. 191-204.
2. GALVIS, A.H (1992). *Ingeniería de Software Educativo*. Santafé de Bogotá: Ediciones Uniandes.
3. Pressman, Roger S. Ingeniería del software un enfoque práctico. Madrid: MCGraw-Hill, 5ta edición, 2002.
4. FIGUEROA, P. (1997). Especificación de Software. Uniandes-LIDIE: Proyecto Ludomática (*documento de trabajo*).
5. MOCKUS, A. (1988). *Pedagogías, escritura e informática. Educadores e Informática: promesas, dilemas y realidades*. Bogotá: Colciencias.
6. ATANASOVA-VUKOVA, A. (1995) : *Para la educación musical en el jardín de infancia, Universidad de Blagoevgrad*.
7. KRASTEVA, S. (1987): Discursos sobre metodología de la escucha de música. Narodna prosveta, Sofia.

ANEXOS

ANEXO A. Glosario de términos

MODULARIDAD: Es la capacidad que tiene un sistema de ser estudiado, visto o entendido como la unión de varias partes que interactúan entre sí y que trabajan para alcanzar un objetivo común, realizando cada una de ellas una tarea necesaria para la consecución de dicho objetivo.

ATRIBUTO: Es cada una de las características de un objeto: identificador, descripción, etc.

CASO DE USO: Corresponde a cada cosa que puede hacer un usuario dentro del modelo de datos. La identificación de los casos de uso se hace con base en los requerimientos de la aplicación que se va a desarrollar.

CLASE: Definición de atributos y métodos para un conjunto de objetos.

DEPURACIÓN: Refinamiento de métodos para optimizar el código del programa que se está desarrollando.

JAVA: Es un lenguaje de programación orientada a objetos desarrollado por Sun Microsystems.

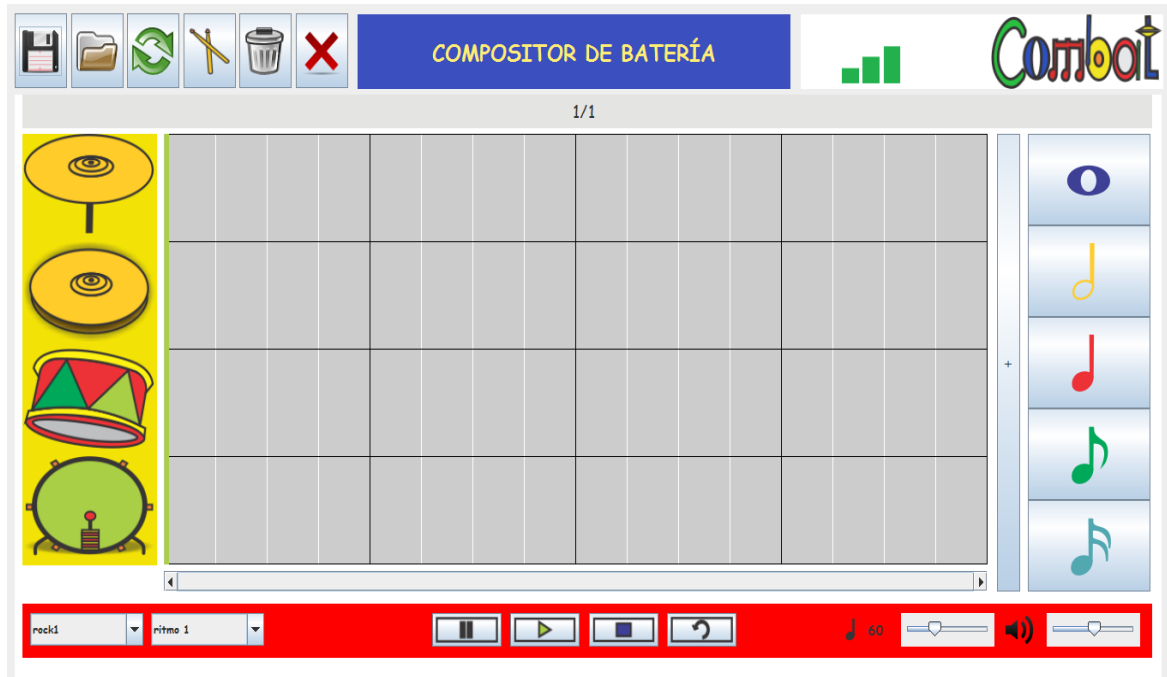
MÉTODO: Corresponde a cada una de las funciones que puede llevar a cabo un objeto, por ejemplo: crearse, eliminarse, etc.

OBJETO: es cualquier cosa que puede ser identificada en el sistema, es decir que tiene unas características y comportamiento particulares.

INSTRUMENTO DE PERCUSIÓN: es un tipo de instrumento musical cuyo sonido se origina al ser golpeado o agitado.

ANEXO B. Descripción por pantallas de la aplicación

Esta es la pantalla principal del sistema, a continuación se describe detalladamente cada uno de sus componentes:



El panel superior contiene herramientas como:

- Abrir: sirve para leer y cargar un archivo desde una ubicación definida por el usuario.
- Guardar: es el encargado de que los usuarios almacenen los datos en un fichero con extensión .bav que está definida por defecto.
- Seleccionar: es una opción que le permite al usuario cambiar el cursor con el fin de realizar actividades como mover una nota desde una posición de la tablatura a otra.
- Eliminar: funciona similar al seleccionar, debido a que cambia el cursor mientras se realiza esta acción, pero esta permite borrar notas de la tablatura.
- Limpiar: borra todo lo que hay en la tablatura en determinado momento.
- Cerrar: para salir de la aplicación.

También muestra el nombre del ritmo que se está reproduciendo y en la esquina derecha muestra el logo de la aplicación para darle identificación al software.



Panel central izquierdo: contiene los instrumentos, estos tienen actividades específicas determinadas por la acción que se ejecute, si se hace clic derecho permitirá hacer un cambio de instrumento y si se hace clic izquierdo desactivará ese instrumento cancelando toda la línea correspondiente en el compás.

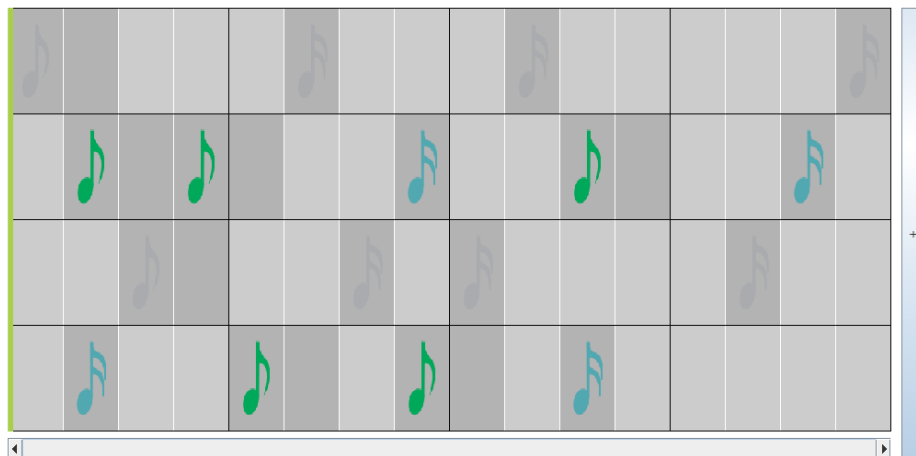
En el panel inferior aparecen los siguientes controles:

- Play: reproduce las notas que están puestas en la tablatura.
- Stop: detiene la reproducción que se esté ejecutando en ese momento.
- Pause: a diferencia del stop detiene la reproducción pero se puede reanudar desde la parte en la que se detuvo.
- Loop: hace que se repitan continuamente secciones cortas de ritmos que se hayan puesto en la tablatura.
- Tempo: dependiendo el valor que se seleccione aumenta o disminuye la velocidad con la que se está reproduciendo el ritmo.
- Volumen: es el encargado de aumentar o disminuir la intensidad con la que suena el ritmo.



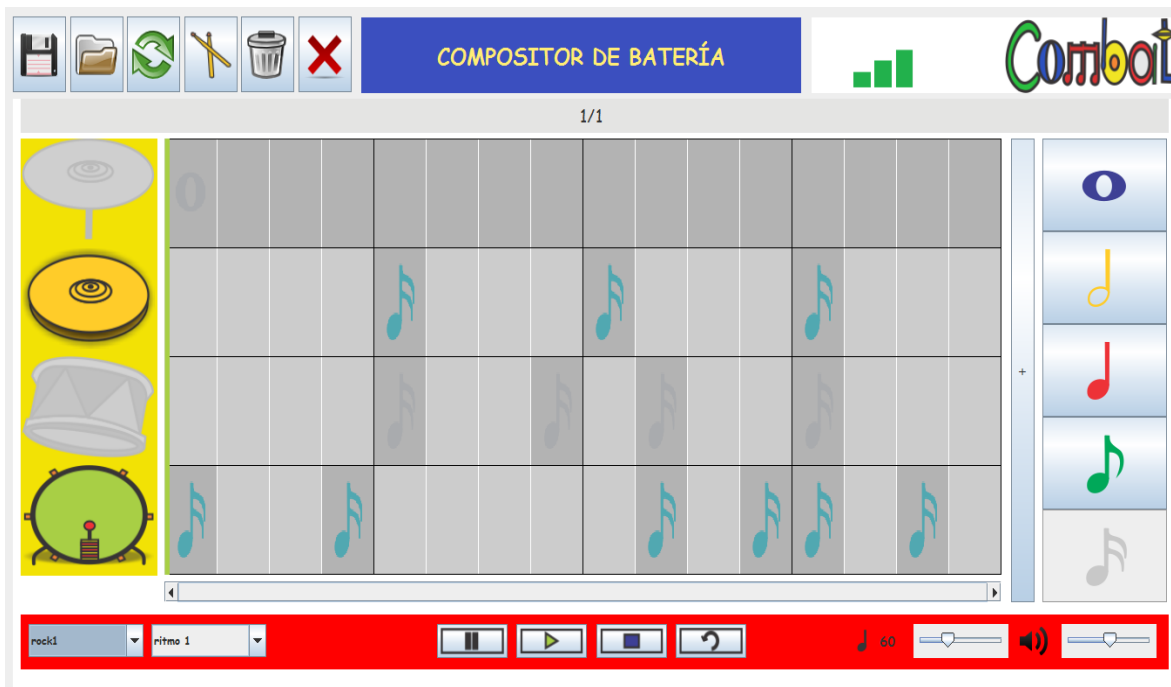
En el panel central se encuentra la tablatura, se muestra como una cuadrícula de 16x4. La forma en la que se maneja es por medio de compases. Un compás se compone de varias unidades de tiempo (redonda, blanca, negra, corchea, semicorchea). Para este caso se maneja un compás 4/4 lo cual significa que el compás podrá tener 1 redonda, 2 blancas, 4 negras, 8 corcheas ó 16 semicorcheas; es por esto que se muestran 16 divisiones en la cuadrícula, porque la máxima cantidad de notas que pueden haber son 16 semicorcheas. En el lado derecho del panel se encuentra un botón que sirve para agregar un nuevo

compás, la forma en la que se podrán visualizar los compases es por medio de un scroll ubicado en la parte inferior.



Y en el panel derecho se encuentran las notas en forma de botones, cuando se está colocando una nota en la tablatura el botón cambia de color para mostrar que esa nota se está usando simulando una activación y desactivación momentánea.

En la siguiente figura se pueden observar algunos instrumentos desactivados, para activar dicha función se debe hacer clic izquierdo sobre el instrumento:



La línea de progreso indica el avance de reproducción de la tablatura:



Con clic derecho sobre algún instrumento se puede hacer un cambio por alguno de los que muestra la lista del menú desplegable:

COMPOSITOR DE BATERÍA

1/2

Crash Cymbal 1
 Closed Hi-hat
 Snare Drum 1
 Bass Drum 2
 Bass Drum 1
 Open Hi-hat
 Snare Drum 2

rock1 ritmo 1

60

The screenshot shows a digital drum composition software interface. At the top, there is a toolbar with icons for saving, opening, undo, redo, deleting, and a red 'X' button. The main title bar is blue and contains the text 'COMPOSITOR DE BATERÍA'. Below the title bar, a progress indicator shows '1/2'. The central area is a 4x8 grid where drum patterns are composed. The first row is empty, while the other three rows contain blue musical notes. A legend on the left lists seven drum types: Crash Cymbal 1, Closed Hi-hat, Snare Drum 1, Bass Drum 2, Bass Drum 1, Open Hi-hat, and Snare Drum 2. To the right of the grid is a control panel with five buttons, each containing a different musical note icon. At the bottom, a red control bar contains dropdown menus for 'rock1' and 'ritmo 1', playback controls (stop, play, record, undo), a tempo display of '60', and two volume sliders.

