

TABDIG: VISUALIZADOR DINÁMICO DE TABLATURAS PARA GUITARRA

CARLOS ANDRÉS ROJAS BELTRÁN

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2016

TABDIG: VISUALIZADOR DE TABLATURAS DINÁMICO PARA GUITARRA

CARLOS ANDRÉS ROJAS BELTRÁN

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

Director:

**MSc. FERNANDO ANTONIO ROJAS MORALES,
MAGÍSTER EN INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2016

CONTENIDO

	Pág.
INTRODUCCIÓN.....	11
1. JUSTIFICACIÓN	13
2. OBJETIVOS	14
2.1. OBJETIVO GENERAL	14
2.2. OBJETIVOS ESPECÍFICOS.....	14
3. FUNDAMENTACIÓN TEÓRICA.....	15
3.1 SOPORTE TEÓRICO	15
3.1.1 Tablatura..	15
3.1.2 Interpretación de tablaturas.	17
3.2 SOPORTE TECNOLÓGICO	18
3.2.1 Netbeans.	18
3.2.2 Java. Su.....	20

4. METODOLOGÍA DE DESARROLLO DEL PROYECTO.	22
4.1 INCREMENTO 1.....	25
4.2 INCREMENTO 2.....	27
4.3 INCREMENTO 3.....	29
4.4 INCREMENTO 4.....	31
4.5 TABLA DE REQUERIMIENTOS	33
4.6 DIAGRAMA DE CLASES.....	35
4.7 DIAGRAMA DE CASO DE USO GENERAL	36
4.8 ESPECIFICACIÓN DE CASOS DE USO.....	37
4.9 FORMATO DE PRUEBA DE USABILIDAD Y UTILIDAD	43
4.10 RESULTADO DE PRUEBA DE USABILIDAD Y UTILIDAD	44
5. CONCLUSIONES	47
REFERENCIAS:	48

LISTA DE TABLAS

	Pág.
Tabla 1. Requerimientos	33
Tabla 2. CASO DE USO 1: Cargar tablatura	37
Tabla 3. CASO DE USO 2: Reproducción de Tablatura	38
Tabla 4. CASO DE USO 3: Desplazamiento a través de los tramos de reproducción	39
Tabla 5. CASO DE USO 4: Cambiar la velocidad de la Simulación.....	40
Tabla 6. CASO DE USO 5: Cambiar el Volumen de los sonidos	41
Tabla 7. CASO DE USO 6: Creación de bucles de repetición	42
Tabla 8. Formato de prueba de usabilidad y utilidad	43

LISTA DE FIGURAS

	Pág.
Figura 1. Metodología de trabajo	23
Figura 2. Diagrama de clases	35
Figura 3. Diagrama caso de uso general	36
Figura 4. Gráfico de los errores más frecuentes de los usuarios.	44
Figura 5. Gráfico con los errores de mayor confusión para los usuarios.	44
Figura 6. Grafica con los tiempos de prueba de cada uno de los usuarios.....	45
Figura 7. Grafico del impacto de memoria rápida de los usuarios.	45
Figura 8. Gráfico con la valoración de utilidad por parte de los usuarios.....	46

RESUMEN

TÍTULO: TABDIG: VISUALIZADOR DINÁMICO DE TABLATURAS PARA GUITARRA*

AUTORES: CARLOS ANDRÉS ROJAS BELTRÁN *

PALABRAS CLAVES: MÚSICA TABLATURA, SIMULACIÓN, GUITARRA

Este proyecto busca desarrollar una herramienta tecnológica que analice la información contenida en archivos de tablaturas y la transforme en una representación visual y dinámica que propicie la interpretación de las mismas. Para el desarrollo de este proyecto se aplicó una metodología de trabajo iterativo e incremental dividida en 4 incrementos, cada uno de ellos asociado a un ciclo de desarrollo de 4 etapas. Cada uno de estos incrementos llevó consigo una realimentación del proceso actual para aplicar en el siguiente incremento.

En este trabajo se utilizó la API de Java para implementar la recepción de los archivos, la reproducción de los sonidos, la visualización de las imágenes y componentes gráficos de la GUI, entre otros. Para la fase final de los incrementos se diseñó un pequeño sistema de pruebas orientado hacia los usuarios finales donde se presentaba el estado actual de la aplicación y ellos retroalimentan el ciclo con sus aportes.

Durante el desarrollo de esta tecnología se buscó garantizar el cumplimiento de sus 3 áreas de interés: algoritmia, práctica y música.

El desarrollo de aplicaciones altamente intuitivas y de gráficos de alta calidad, permite ampliar la población de usuarios incluyendo aquellos que sus conocimientos de informática y tecnología sean muy limitados.

El planteamiento de proyectos que involucre la investigación de otras áreas del conocimiento como la música y el diseño intuitivo favorece la interacción de los estudiantes con comunidades ajenas a su campo de acción permitiendo elaborar proyectos interdisciplinarios y con múltiples áreas de interés.

* Trabajo de Grado

* Facultad de Ingeniería Físico Mecánicas, Escuela de Sistemas. Director Fernando Antonio Rojas Morales

ABSTRACTS

TITLE: TABDIG: DYNAMIC DISPLAY OF GUITAR TABS*

AUTHORS: CARLOS ANDRES ROJAS BELTRAN**

KEY WORDS: SIMULATION, TABLATURE, GUITAR MUSIC

This project seeks to develop a tool technology that analyze the information contained in files of tablature and the transform in a representation visual and dynamic that conducive to the interpretation of them same. For the development of this project are applied a methodology of work iterative and incremental divided in 4 increments, each one of them associated to a cycle of development of 4 stages. Each one these increments led get a feedback from the process current to apply in the next increase.

In this work is used the API of Java to implement it reception of them files, the reproduction of them sounds, the display of them images and components graphics of the GUI *, between others. For the phase end of them increments is designed a small system of tests oriented towards them users late where is presented the State current of the application and they feed back the cycle with their contributions.

During the development of this technology are sought to ensure the compliance of its 3 areas of interest: algorithms, practice and music.

The development of applications highly intuitive and of graphics of high quality, allows expand the population of users including those that their knowledge of computer and technology are very limited.

The approaches to projects involving the investigation of other areas of knowledge as the music and the intuitive design promotes the interaction of students with communities outside their field of action allowing to develop interdisciplinary projects with multiple areas of interest.

* Degree project

**Faculty of engineering physical mechanical, school of systems. Director Fernando Antonio Rojas Morales

INTRODUCCIÓN

Este proyecto busca desarrollar una herramienta tecnológica que analice la información contenida en archivos de tablaturas y la transforme en una representación visual y dinámica que propicie la interpretación de las mismas. Para el desarrollo de este proyecto se aplicó una metodología de trabajo iterativo e incremental dividida en 4 incrementos, cada uno de ellos asociado a un ciclo de desarrollo de 4 etapas. Cada uno de estos incrementos llevó consigo una realimentación del proceso actual para aplicar en el siguiente incremento.

Durante el desarrollo de esta tecnología se buscó garantizar el cumplimiento de sus 3 áreas de interés: algoritmia, práctica y música.

La música porque la temática principal del software son las tablaturas y todo el contenido temático que esta tiene. La práctica de la guitarra porque los usuarios podrán tener completo control en la reproducción de la simulación, identificando la correcta ubicación de las notas en el diapasón de la guitarra. La algoritmia porque el objetivo principal de este proyecto es el de analizar, estandarizar, transformar y reproducir en forma visual y dinámica el contenido de las tablaturas.

La tablatura es un formato de escritura musical informal que carece de un formato estándar rígido, esto significa que se pueden encontrar variantes en su escritura. Por este motivo es de suma importancia la implementación de algoritmos que permitan el análisis, la estandarización y la simulación.

Para lograr una buena aceptación entre los usuarios finales, durante las 4 fases de diseño se plasmaron conceptos y técnicas de la interacción hombre computador

(HCI)*. Para el diseño del software se utilizó el Lenguaje UML* donde se hizo la abstracción de las relaciones y entidades del mundo real, para posteriormente plasmarlas en la estructura lógica.

En este trabajo se utilizó la API de Java para implementar la recepción de los archivos, la reproducción de los sonidos, la visualización de las imágenes y componentes gráficos de la GUI*, entre otros. Para la fase final de los incrementos se diseñó un pequeño sistema de pruebas orientado hacia los usuarios finales donde se presentaba el estado actual de la aplicación y ellos retroalimentan el ciclo con sus aportes.

* HCI conjunto de técnicas y principios relacionados con el diseño desarrollo y estudio de los fenómenos que rodean los sistemas informáticos para uso humano. referencia: albertolacalle.com/hci.htm

* UML Es un lenguaje gráfico para visualizar y documentar cada una de las partes que comprende el desarrollo de software. Referencia: users.doc.uchile.cl/~psalinas/uml/introducción.html

* GUI Es la interfaz gráfica que un software, aplicativo o programa le enseña al usuario para interactuar con el. Referencia: archwindevelopment.techtarget.com/definition/GUI

1. JUSTIFICACIÓN

La tablatura es una forma de representar un conjunto de notas para su posterior interpretación. Los elementos que usa para su representación son los identificadores de cuerdas y de trastes. Se usa en la práctica para diferentes instrumentos musicales, incluyendo la guitarra¹.

Con la evolución del uso de la información en internet, la tablatura ha sido aceptada por guitarristas profesionales y empíricos de todo el mundo, llegando a convertirse en un mapa de convenciones universal. La tablatura gráficamente se ve como una matriz de posiciones donde las filas representan las cuerdas y los números sobre las filas el traste a tocar.²

Para los lectores de tablaturas interpretarlas de esta manera conlleva un esfuerzo mental y visual. Gracias a la tecnología es posible desarrollar visualizadores que transformen la información estática de una tablatura en información dinámica³.

El presente proyecto tiene como objetivo desarrollar un visualizador de tablatura dinámico para transformar la representación estática en una representación audiovisual dinámica usando un diapasón digital.

¹ OSORIO Frank Octavio. Curso Básico de Guitarra Popular Tópicos Elementales para su Aprendizaje. [en línea] [citado 10 de julio de 2016] Disponible en: <http://www.mundomanuales.com/manuales/3037.pdf>

² NAVARRO, José Luis. Método Postmoderno de la Guitarra en: Curso de guitarra Clásica 2007 [en línea] [citado 10 de julio de 2016] Disponible en: <http://cursodeguitarraclasicaenlinea.blogspot.com>

³ FERREYRO, J. Blogs, una Reflexión Compartida sobre Buenas Experiencias de Enseñanza. Revista Electrónica de Tecnología Educativa, 24.2007. [en línea] [citado 10 de julio de 2016] Disponible en: <http://edutec.rediris.es/Revelec2/revelec24/pdf/Edutec24-JFerreyro-Blogs.pdf>

2. OBJETIVOS

2.1. OBJETIVO GENERAL

Desarrollar una representación dinámica de una tablatura para guitarra en un diapasón digital, aplicando un algoritmo de transformación, como medio para la práctica de la guitarra.

2.2. OBJETIVOS ESPECÍFICOS

- Analizar la información contenida en los archivos de tablatura mediante el desarrollo de un algoritmo de interpretación que permita leerlos correctamente.
- Transformar adecuadamente la información contenida en ellos implementando un algoritmo con el fin de representar dinámicamente las tablatras en un diapasón digital.
- Diseñar una interfaz de usuario implementando mecanismos de control para emular el comportamiento de un diapasón digital adaptable a las necesidades de los usuarios y sus líneas de aprendizaje.
- Plantear y desarrollar un conjunto de pruebas de utilidad y usabilidad para evaluar el impacto de la herramienta desarrollada en la comunidad de practicantes de la guitarra.

3. FUNDAMENTACIÓN TEÓRICA

3.1 SOPORTE TEÓRICO

3.1.1 Tablatura. La tablatura es un documento que define una forma de escritura especial para ciertos instrumentos musicales. A diferencia de la notación musical corriente o tradicional, en ella se presentan únicamente las posiciones y colocaciones en el instrumento para la interpretación de una pieza, y no las alturas ni las duraciones de los tonos.

La tablatura en cambio enfatiza sólo en el trabajo operacional de la guitarra, por lo que sólo indica dónde y cuándo un dedo se debe colocar para reproducir una nota. Para el caso de las tablaturas en instrumentos de cuerdas, se representan las notas por medio de un sistema de coordenadas, relacionando las cuerdas y los trastes del instrumento.

Las tablaturas para instrumentos de cuerda hacen uso de letras o números sobre líneas o letras colocadas libremente. En el caso de las tablaturas de laúd o de guitarra, normalmente sólo se indica el momento de inicio de un sonido, pero no su duración ni la conducción de las diferentes voces en piezas polifónicas. En las tablaturas para guitarra actuales (muchas de ellas encontradas en Internet), muchas veces no se especifica el ritmo, de forma que el intérprete debe conocer la pieza para poder tocarla.

Cuando la tablatura comenzó a utilizarse en el siglo XVII se presentaron diferentes incidentes para interpretarla. Sin embargo gracias a la motivación de personas

contemporáneas de ese entonces, el cambio a usar la notación de tablatura pudo comenzar.⁴

Le sieur Perrine fue uno de los promotores de este cambio, y lo hizo cuando en 1679 escribió un libro de teoría llamado: *laúd Livre de Musique pour le Lut*⁵. En el texto resalta la importancia que tendrá la tablatura en el aprendizaje del Laúd que para aquel entonces era el instrumento que la mayoría de músicos quería aprender tocar.

El noble propósito de Perrine para facilitar la integración del laúd y su música en la sociedad general de los músicos, hizo que la tablatura se convirtiera en el modo preferido de la transcripción de los transcriptores académicos en el siglo XX⁶.

Desde entonces la acogida que ha tenido la tablatura ha venido evolucionando convirtiéndose en una estandarización tal y como Perrine lo pronosticó siglos antes, en la actualidad esta notación es común para instrumentos de cuerda con trastes tales como el laúd, vihuela o guitarra⁷. Otros instrumentos con los que se puede utilizar la tablatura son la armónica, órgano y la ocarina⁸.

Una diferencia notable entre la notación estándar y la notación de tablatura, es el cuerpo de conocimiento que expresan. La notación clásica representa el ritmo y la duración de cada una de las notas, así como su altura con respecto a la escala, la cual está basada en una división de doce tonos⁹.

⁴ MATANYA Ophée. A History of Lute TABlature Transcriptions.2000 [en línea] [citado 12 de julio de 2016]
Disponible en: <http://www.guitarandluteissues.com/trans/trans.html>

⁵ Ibid

⁶ Ibid

⁷ Ibid

⁸ MAESTROS OF THE GUITAR. The history of Tablature Notation. 2005 [en línea] [citado 13 de julio de 2016]
Disponible en: <http://www.maestros-of-the-guitar.com/tablature.html>

⁹ Ibid

3.1.2 Interpretación de tablaturas. La tablatura, que abreviado se denomina TAB, es un método para escribir música para ser interpretada por instrumentos de cuerdas como la guitarra y el bajo.

Se utilizan letras (ASCII) y números , haciéndolo ideal para lugares como internet en donde cualquier persona puede tener acceso a un archivo de texto copiarlo y leerlo.

La tablatura es una representación del brazo de la guitarra por medio de seis líneas dispuestas de manera horizontal representando las cuerdas de la guitarra (si se trata de un bajo, serán cuatro líneas y así con cualquier instrumento). A la izquierda, al principio de cada línea horizontal se colocan letras en cifrado americano que dicen las notas de cada cuerda y se deduce en que afinación debemos tocar la tablatura, si no hay ninguna indicación se toma la afinación estándar. La cuerda más aguda estará representada arriba de todo, es decir es la primera línea que se dibuja, en ocasiones la primera cuerda se dibuja con una "e" minúscula al lado.

```
e |-----|  
B |-----|  
G |-----|  
D |-----|  
A |-----|  
E |-----|
```

Básicamente, la tablatura indica que nota se debe tocar en el lugar exacto (esta es la gran diferencia que existe con respecto a la partitura en donde uno debe deducir que digitación usar). Los números representaran el traste en el que está ubicada la nota, en el caso de que diga 0 se tomará como una cuerda al aire.

Es importante saber que en un TAB no se indican la duración de las notas, es mejor conocer la canción para darte una idea de cómo tocarla. En ocasiones la duración puede deducirse según la distancia que hay entre cada número, por ejemplo si están muy juntos el tema será rápido. Tampoco te indicará que dedos debes utilizar para presionar las cuerdas¹⁰. [9]

Un ejemplo de tablatura puede verse más o menos así:

```

E---5---7-----3---7-----
B---5---8-----0-1-----1-1---5-----8--7-5-7-----2-3
G---5---7-----2---2-----5-----4---
D---7---9---7-9---2-----4-----
A---7---10-----0-----0-----2-----
E---5---7-----8-7-----

```

Al principio puede resultar bastante complicado pero la tablatura requiere mucha menos dedicación que la partitura. Observa que cuando hay varios números uno encima de otro (como en el ejemplo) indicara que deben tocarse al mismo tiempo, representando un acorde. El TAB debe leerse al igual que cualquier texto, de izquierda a derecha.

3.2 SOPORTE TECNOLÓGICO

3.2.1 Netbeans. Es un entorno de desarrollo integrado libre (IDE)[10]. Creado principalmente para el lenguaje de programación Java. Existe además un gran

¹⁰ BÁEZ Manuel. Cómo afinar una guitarra: todo sobre la afinación en línea] [citado 13 de julio de 2016] Disponible en: www.guitarristas.info/tutoriales/como-afinar-guitarra-todo-sobre-afinacion/2787#

número de módulos para extender sus funcionalidades y alcances. 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 evolución y de búsqueda constante de nuevos retos.

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 API's de NetBeans y un archivo especial (Manifest File) que lo identifica como módulo¹¹

Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en esta plataforma pueden ser extendidas fácilmente por otros desarrolladores de software. Con el paso del tiempo la comunidad de desarrolladores decidió comenzar a construir sus propios plug-ins[12], transformando para siempre la manera de ver un entorno de desarrollo.

Este auge de plug-ins generó el nacimiento de un mercado de desarrollos pequeños pero muy útiles y documentados por cada uno de sus creadores. Esto le permitió a Java hacerse con un fama llamativa impulsando el interés de más y más desarrolladores para estudiar el lenguaje y hacer uso de estos plug-ins de alta documentación.

Los Servicios más comunes que ofrece NetBeans están orientados hacia la plataforma de escritorio y con esto sus principales características y ventajas:

- Administración de las interfaces de usuario (ej. Menús y barras de herramientas).

¹¹ Manifest File es un archivo que especifica contenido en un archivo jar. En este archivo se define cual es la clase principal de la aplicación. referencia: es.wikipedia.org/wiki/Manifest

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

3.2.2 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.

Debido a la existencia de distintas arquitecturas de ordenadores y con la evolución constante de las tecnologías era necesaria la invención de una nueva herramienta que funcione de manera suficiente independientemente de la naturaleza de la plataforma (electrodoméstico).

Con esta causa se llevó al surgimiento de Java. No fue sino hasta finales de 1995 que Java se introdujo como lenguaje de programación para computadores. El éxito del proceso se debió a la incorporación de un intérprete Java en la versión 2.0 del programa Netscape Navigator, produciendo una verdadera revolución en internet.

Dos años más tarde en 1997 se lanza la versión 1.1 mejorando sustancialmente a primera versión del lenguaje. y su sucesor el famoso Java 1.2 o Java 2 como también se le conoce saldría hasta finales de 1998.

Una característica fundamental cuando se trabaja con Java es la cantidad de clases preexistentes y su fácil adaptación en fases iniciales de desarrollo. La gran parte de clases preexistentes se encuentran en la AP* oficial de Java.

Dentro de su propia estructura este lenguaje reúne muchos aspectos que en cualquier otro lenguaje son extensiones propiedad de empresas de software o fabricantes de ordenadores (ej: ejecución remota, Componentes, Hilos de ejecución, Bases de datos y Seguridad).

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 estado en constante evolución y con esto su comunidad de programadores (6,5 millones aproximadamente) siendo seguramente la más grande y activa del mundo.

Después de tantas versiones y haber cautivo a una comunidad tan numerosa de desarrolladores en la actualidad el reto principal de Java es el de convertirse en el “Nexo universal” que permita conectar a todos los usuarios con la información, accediendo a ella de manera local o remota en dispositivos de escritorio o móviles.

*

4. METODOLOGÍA DE DESARROLLO DEL PROYECTO.

En toda tarea que se realiza en la Vida Cotidiana. Se Debe tener un orden y establecer distintas prioridades para que la actividad que se planifique se desarrolle y se alcance su respectivo éxito y alcanzando el objetivo que se ha planteado desde un principio, evitando que este resultado esté condicionado por factores aleatorios y que la cuota del azar o la suerte no sea la más importante a la hora de efectuarlo.

Para poder realizar cada una de estas tareas es necesaria la aplicación de una metodología de Trabajo, teniendo para ello que contar con un conocimiento previo que permita establecer una forma de llevarse a cabo, por lo que se considera siempre que el primer paso metodológico consiste en la observación del campo de aplicación.

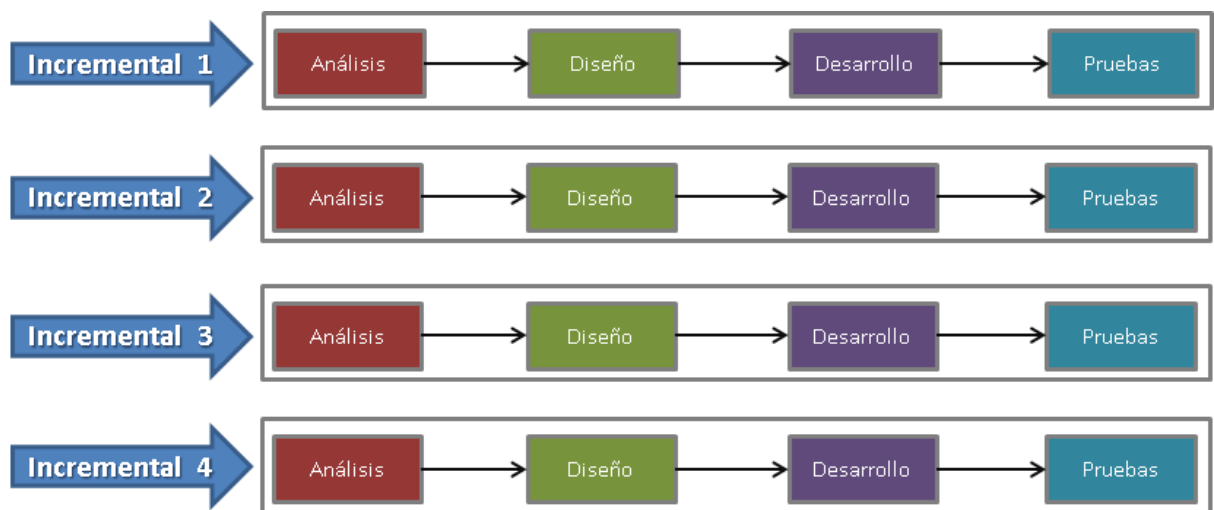
Luego de esta tarea es cuando se plantea cómo llevar a cabo una tarea, para obtener una conclusión o resultado que sea el esperado, y es allí cuando surgen los conocidos pasos metodológicos, que consisten en la forma en la que se debe realizar la ejecución del método, siguiendo una serie de instrucciones que se ejecutarán en forma Empírica.

Estos pasos no pueden ser desordenados, sino que tienen que tener un orden de ejecución determinado, que muchas veces es referido como un Sistema metodológico, garantizando que cada una de las acciones que se irán llevando a cabo estén ordenadas y lo más importante, que el objetivo no sea alcanzado en forma aleatoria, sino que la ejecución del método pueda ser aplicado para un caso particular, o bien estableciendo una metodología General.

La Metodología es muy importante en el mundo de la ciencia y los conocimientos, refiriéndonos en este caso bajo el concepto de método científico, aunque también es aplicable por ejemplo al ámbito laboral, donde se tiene una metodología de trabajo que lleva a lograr un mayor rendimiento y productividad, como también una metodología de estudio que permite alcanzar una mayor eficiencia a la hora de estudiar y realizar alguna labor educativa o didáctica.¹²

La metodología que se implementó en el proyecto es iterativa e incremental. La misma esta segmentada en 4 fases incrementales denominadas: incrementos. Las cuales se subdividen en etapas de Análisis, Diseño, Desarrollo y Pruebas.

Figura 1. Metodología de trabajo



Las etapas de Análisis consisten en la identificación de requerimientos demandados por los usuarios y su posterior incorporación al conjunto de tareas por desarrollar.

¹² IMPORTANCIA.ORG. Importancia de la Metodología.[en línea] [citado 15 de julio de 2016] Disponible en: <http://www.importancia.org/metodologia.php>

En cada uno de los incrementos se replanteó el conjunto de requerimientos a desarrollar y se determinó la viabilidad y la prioridad asignada a cada uno de los requisitos. Después de finalizar las etapas de pruebas en cada incremental se tomaron en consideración nuevos requerimientos debido a los resultados de las pruebas y con esto se adicionaron los nuevos requerimientos para el desarrollo.

Las etapas de diseño se diseñó la estructura lógica del software basándose en una distribución de clases que modelan el comportamiento de las tablaturas. Fue necesario refinar y reestructurar la lógica y estructura de los algoritmos de la herramienta debido a que por el sistema de pruebas con los usuarios en las partes finales de las etapas incrementales.

El diseño de la interfaz fue evolutivamente cambiando sin mayores retrasos pues los elementos más importantes y especiales del software fueron identificados arduamente en el desarrollo del análisis de requerimientos. De igual manera fue importante el conocimiento musical y de usabilidad transferido del director de este proyecto.

Las etapas de desarrollo o de implementación. Abarca todo el desarrollo de algoritmos y la lógica del software, tomando como base inicial los requerimientos iniciales tomados de la comunidad de practicantes y de la necesidad de diseñar una herramienta que haga más sencilla la práctica de tablaturas en la guitarra.

Las etapas de pruebas El sistema de pruebas es la etapa que generó mayor impacto en el proyecto porque con los resultados de las pruebas se termina una iteración y se da comienzo a la siguiente con una mayor profundidad.

Las pruebas que se utilizaron están divididas en pruebas de usabilidad y pruebas de utilidad, las cuales tienen como finalidad medir el impacto que la herramienta entrega a los usuarios y propiciar la realimentación necesaria para que la aplicación tenga sus ciclos incrementales.

4.1 INCREMENTO 1

Etapas de Análisis:

Los requerimientos principales para el Desarrollo de TabDig fueron los siguientes:

- Leer archivos de texto.
- Analizar la información contenida en ellos.
- Identificar el traste y la cuerda de la nota.
- Representar visualmente las notas (por medio de un carácter ASCII)
- Crear una representación visual de la guitarra y una grilla de elementos donde visualizar la representación visual de las notas.

Etapas de Diseño:

- Se diseñó un algoritmo que permitiera leer los archivos de texto de manera predefinida sin que el usuario pudiera escogerlo.
- Una vez recibido el archivo de texto se procedió a analizar su contenido para validar si correspondía realmente a un archivo de tablatura o no.

- Para la identificación del traste y cuerda de la nota correspondiente se tuvo en cuenta para trastes de un dígito.
- Una vez identificadas las notas correspondientes se diseñó la primera versión de la GUI en donde se representó el diapasón digital a través de espacios en blanco los cuales se reemplazarían por el carácter “*” (código ASCII 42)¹³.

Etapa de Desarrollo:

En la etapa de desarrollo se implementaron los algoritmos de lectura de archivo único, validación de archivo y simulación. El tiempo de respuesta de los algoritmos fue aceptable por este motivo se conservaron.

Etapa de Prueba:

Durante la ejecución de la primera prueba de la aplicación se identificó que la GUI era demasiado pobre, no tenía un diseño estético claro y la representación de los trastes por medio de un carácter generaba una mala visualización.

Además quedó en evidencia que las tablaturas debían ser estandarizadas algorítmicamente debido a que el algoritmo de simulación tomaba solo notas de un dígito, por lo que cuando la nota estaba asociada a un traste de dos dígitos este dividía la nota en dos uno para cada dígito, fue por este motivo que posteriormente se agregó el requerimiento de un algoritmo de estandarización.

¹³ El Código ASCII, es un código que fue propuesto por Robert W. Bemer como una solución para unificar la representación de caracteres alfanuméricos en las computadoras. Tomado de <http://lacomputadora.org/que-es-el-codigo-ascii-y-para-que-sirve/166>

4.2 INCREMENTO 2

Etapa de Análisis:

- Leer cualquier archivo de texto suministrado por la interfaz de explorador de archivos.^{14]}
- Analizar el contenido de los archivos, separando las notas de cada cuerda por aparte.
- Tener en cuenta los trastes de dos dígitos.
- Representar el indicador de pulsación de una forma más amigable
- Indicar la afinación estándar de las 6 cuerdas de la guitarra.
- Crear dos botones en la GUI uno para pausar/reanudar la reproducción y otro para control el volumen de las notas.

Etapa de Diseño:

- Se diseñó un sistema de botones de menú para permitirle al usuario seleccionar las tablaturas desde el explorador de archivos.
- Para arreglar los problemas presentados en la realimentación del incremental 1 se reestructuro los algoritmos que permiten analizar el contenido de los archivos y así mismo distribuir correctamente los trastes cuando se presentan dos dígitos.

¹⁴ Explorador de archivos se le dice a la ventana de archivos en Windows

- Para mejorar el indicador de pulsación, se aumentó de tamaño, se coloreó con un degradado de azul y blanco y posterior a eso se editó para que no se visualizara tan pixelado^{15]}
- Se dispone del espacio apropiado para representar gráficamente la nota asociada a cada cuerda cuando se tocan al aire¹⁶.
- El diseño de los iconos para los botones de la interfaz fueron creación propia del autor siguiendo los parámetros de diseño de muchas interfaces modernas.

Etapas de Desarrollo:

Para implementar la interfaz de explorador de archivos, fue necesaria la implementación de la clase JFileChooser. El reto importante en este incremental fue tener en consideración los caracteres dobles cuando se leía un traste de dos dígitos, para esto fue necesario guardar temporalmente las cifras de los trastes y validar en el siguiente ciclo si continuaba la sucesión numérica de los trastes.

Etapas de Pruebas:

La corrección oportuna del algoritmo de análisis permitiendo la validación de los trastes de dos dígitos, permitió que las tablaturas pudieran ser leídas correctamente. Sin embargo se detectó que el formato de tablatura no está estandarizado y por eso era necesario contemplar más de un posible formato, esto llevo a analizar los archivos por medio de Expresiones Regulares¹⁷.

¹⁵ Pixelado es un efecto causado por visualizar una imagen o una sección de una imagen a un tamaño en el que los pixels individuales son visibles al ojo.

¹⁶ Tocar al Aire es una expresión utilizada por músicos para indicar que las cuerdas se tocan sin marcar ninguna nota en el diapasón. Es decir la nota producida es la de la cuerda misma.

¹⁷ Expresiones Regulares es una secuencia de caracteres que forma un patrón de búsqueda, principalmente utilizada para la búsqueda de patrones de cadenas.

La representación del indicador de pulsación causo mejoras en la sensación de los usuarios sin embargo no es del todo la versión final del indicador, pues es necesario añadirle algún efecto visual que permita identificar con mayor claridad el traste que se está tocando.

4.3 INCREMENTO 3

Etapa de Análisis:

- Analizar los Archivos de texto a través de las Expresiones Regulares para garantizar la fidelidad de la información leída.
- Mejorar el indicador de pulsación para permitirle una mayor interpretación a los usuarios.
- Representar con una imagen de alta calidad la interfaz del diapasón de la guitarra.
- Adicionar más controles de simulación a la interfaz (control de velocidad, reinicio y bucles de reproducción).
- Relacionar la velocidad por tiempo de simulación y la métrica de música bpm (beats per minute)¹⁸.

¹⁸ bpm proviene de las siglas "Beat por Minute", que en castellano significa "golpes por minuto". sirve para establecer la duración y/o velocidad de las figuras musicales con exactitud.

Etapa de Diseño:

La utilización de Expresiones Regulares ameritaba elegir una que permitiera el rápido entendimiento de la misma y funcionara perfectamente con caracteres especiales que es la base para identificar cuando un archivo de texto es una tablatura.

Para la interfaz de alta calidad se optó por ilustrar el diapasón de una guitarra desde cero permitiendo adaptar la imagen a una representación realista y clara para los usuarios. Igualmente los controles de la simulación debían ser intuitivos y guiados por la línea de diseño de los anteriores.

Etapa de Desarrollo:

Se implementaron la utilización de expresiones regulares para analizar correctamente los archivos ingresados. Con esto se valida muchas variantes de las tablaturas permitiéndole al usuario usar múltiples archivos sin ninguna complicación.

Se implementó un diseño para el indicador de pulsación diferente, y se sometió a un cambio de color y al final se decidió mantener el color azul. Se implementaron los controles correspondientes para tener mayor control en la simulación los iconos se diseñaron desde ceros manteniendo la línea grafica de los demás.

También se implementó un método que permitiera transformar las unidades de la velocidad, pasar del tiempo de vuelta del timer y pasarlo a unidades de bpm. Con esto se implementó también los textos visualizadores de cada una de estas métricas.

Etapa de Pruebas:

Durante las pruebas se evidenciaron grandes cambios para la funcionalidad principal del software, las personas consultadas manifestaron que para las personas menos experimentadas sería bueno guiarlos con un directorio predefinido de tablaturas de manera que se pudieran encontrar más fácilmente.

También se evidenciaron fallos en los ciclos de reproducción principalmente al momento de seleccionar el fin de los mismos. Con esto se plantea implementar unos elementos visuales que permitan la modificación del ciclo sin ningún problema. Estos elementos también tienen que ser visibles en los controles que lanzan los bucles.

4.4 INCREMENTO 4

Etapa de Análisis:

- Los controles de bucles de reproducción presentaban fallos al activarse y no es posible modificar los ciclos cerrados.
- Se deben implementar nuevos algoritmos para permitir la selección de tablaturas por medio de un directorio predefinido y unas tablaturas llamadas demos previamente descargadas.

Etapa de Diseño:

- Para modificar el problema de la definición de los tramos para inicio y fin de bucle fue necesario establecer una métrica que permitiera predefinir estos

valores teniendo en cuenta la relación entre pixeles y desplazamiento por tramo de la barra de progreso.

Etapas de Desarrollo:

Lo primero que se tuvo en consideración para esta última etapa de desarrollo fue establecer una métrica que permitiera relacionar los pixeles de desplazamiento con la diferencia entre los tramos que se ejecutan en la barra de progreso.

Una vez establecida esta métrica se procedió a calcular las posibles posiciones donde las marcas podían marcar el inicio y el fin de los bucles. Con esto se corrigió el problema del descuadre de tramos que se estaba presentando.

Se creó un directorio por defecto que logrará serles claro a los usuarios para que de esta manera pudieran guardar sus tablaturas de manera ordenada y de fácil identificación y acceso.

Etapas de Pruebas:

El resultado final de las pruebas determinó que TabDig en su primera versión cumple plenamente con sus objetivos y que ambiciona con crecer y llegar a convertirse en una aplicación muy versátil para las personas que constantemente están estudiando las tablaturas.

El listado de las recomendaciones que están por fuera del alcance de este proyecto pero que permitirán que TabDig evolucione y se convierta en una herramienta de más prestaciones, se encuentran en la sección de recomendaciones.

4.5 TABLA DE REQUERIMIENTOS

El siguiente es el consolidado de requerimientos utilizado para el análisis de cada uno de los incrementos de la metodología de desarrollo.

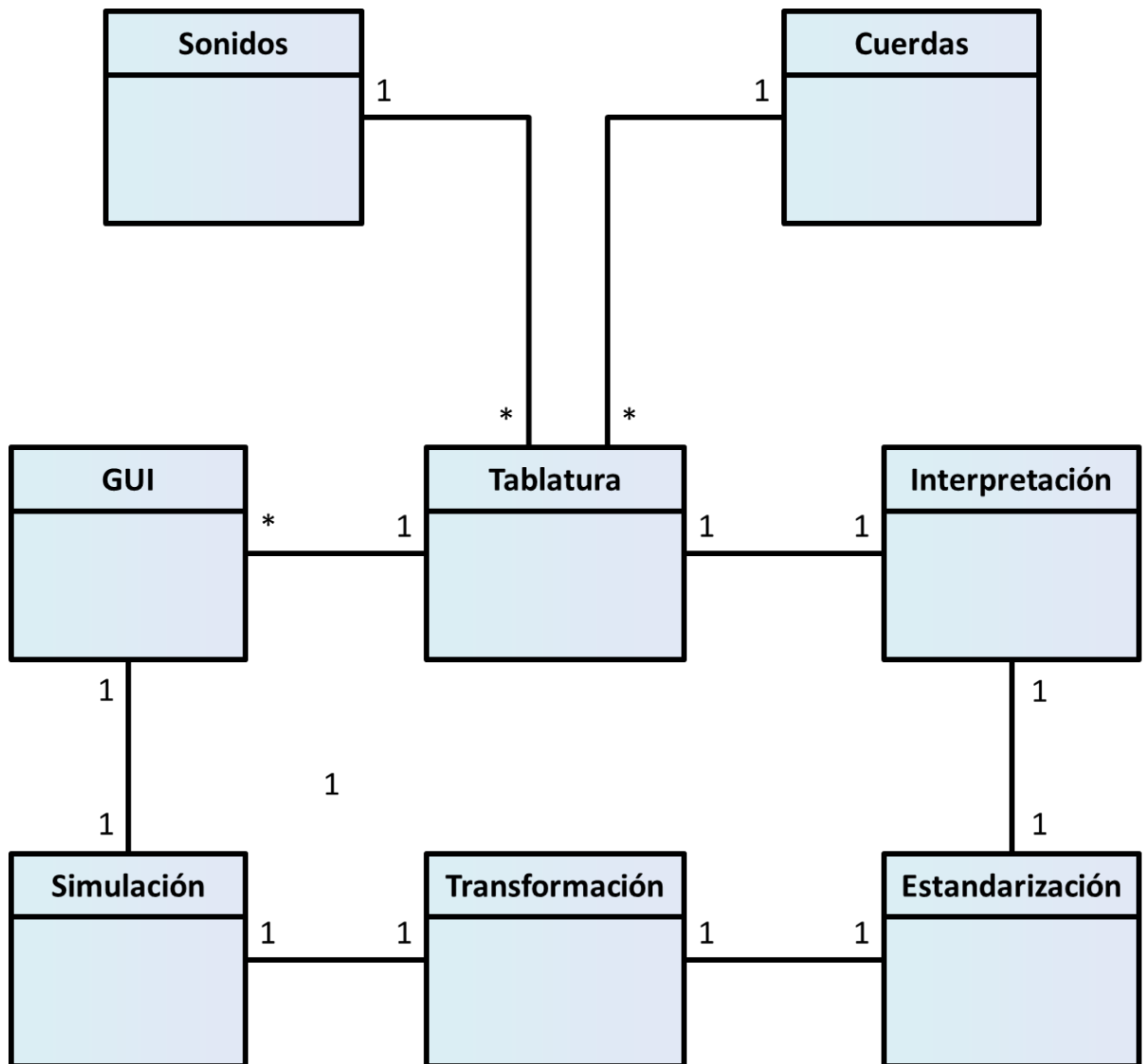
Tabla 1. Requerimientos

#	Requerimiento	Descripción	Peso
R1	Leer cualquier archivo de texto.	Leer archivos de texto para extraer la información encontrada en ellos.	5
R2	Validar el archivo de texto.	Validar si el archivo de texto suministrado es o no una tablatura.	5
R3	Analizar la información de los archivos de texto.	Analizar la información de los archivos para guardar lo relevante.	5
R4	Identificar el traste y cuerda de la nota.	Con el fin de identificar las notas para reproducirlas correctamente en la simulación.	5
R5	Representar visualmente las notas a tocar.	Le permitirá a los usuarios a posicionar correctamente las Notas en el diapasón.	5
R6	Crear una interfaz amigable y clara para los usuarios	Se tiene pensado que los usuarios serán personas de pocos conceptos de informática. Por eso la presentación visual de la interfaz es sumamente importante.	5
R7	Tener en cuenta la lectura de notas con traste de dos dígitos.	Inicialmente no se tenía contemplado como un requerimiento ya que fue necesario hacer las pruebas del primer incremental para notar la deficiencia que presentaban los algoritmos iniciales.	5

#	Requerimiento	Descripción	Peso
R8	Crear botón de control para la simulación.	Era sumamente importante implementar el botón que controla la reproducción de la simulación.	5
R9	Implementar el control de velocidad para la simulación.	Esto permitirá a los usuarios el tener un control de la velocidad de la simulación, ayudándole a su aprendizaje.	5
R10	Representación de la guitarra de alta calidad.	Es un aspecto meramente estético para que cautive la atención de los usuarios.	3
R11	Crear botón de control para el volumen de la simulación.	Permitiendo modificar el valor de los decibeles de los sonidos durante la simulación	3
R12	Indicar la afinación estándar de las cuerdas de la guitarra.	La configuración de notas para las cuerdas al aire de la guitarra, está pre definido en las tablaturas debido a que es un estándar oficial de las mismas.	3
R13	Control de Reinicio.	Permite a los usuarios volver al inicio de la tablatura.	3
R14	Control de Bucles de reproducción.	Es la funcionalidad estrella de la aplicación, permitiendo el aprendizaje de las tablatura a los usuarios de manera de una manera más apropiada e interactiva.	☆

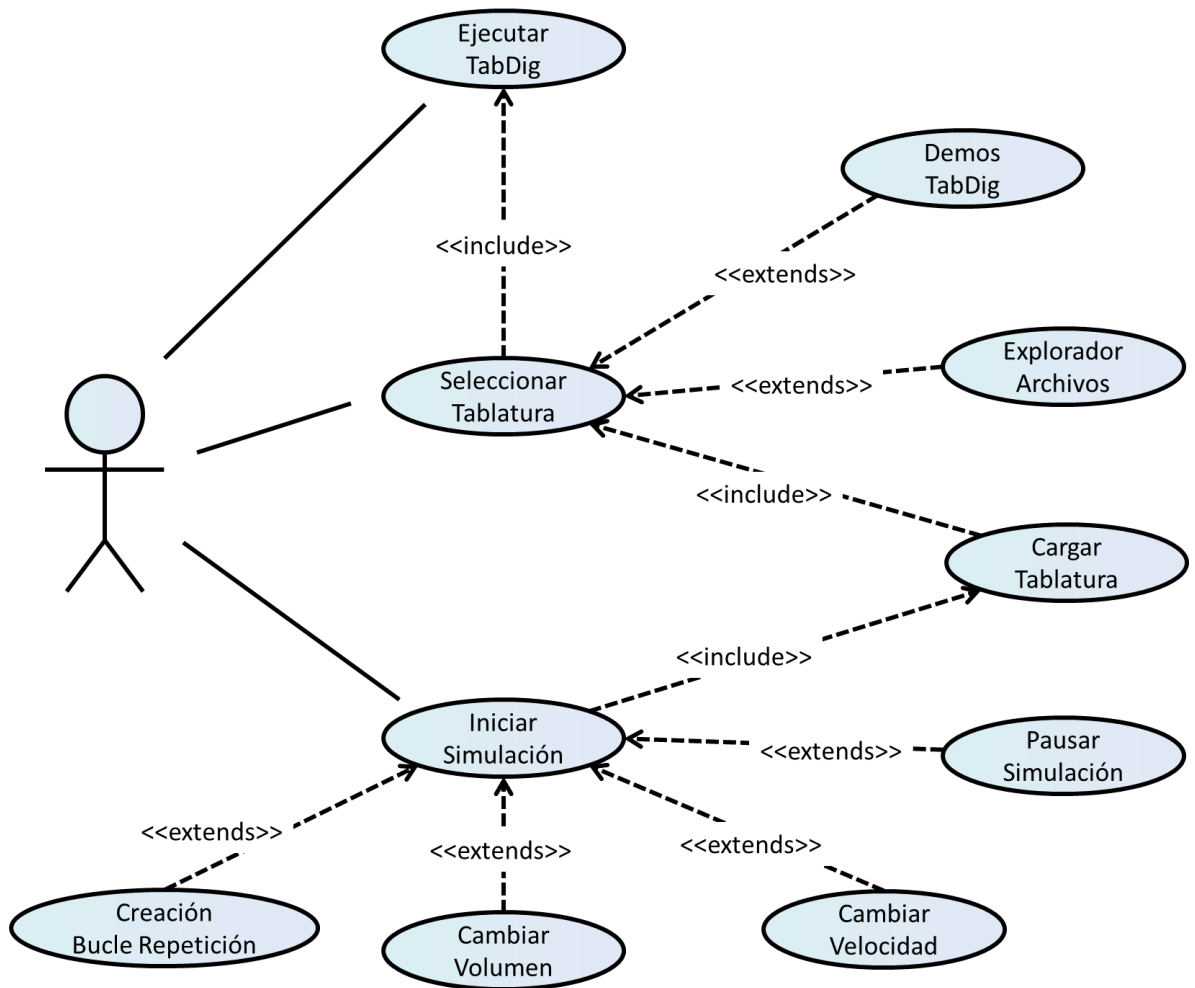
4.6 DIAGRAMA DE CLASES

Figura 2. Diagrama de clases



4.7 DIAGRAMA DE CASO DE USO GENERAL

Figura 3. Diagrama caso de uso general



4.8 ESPECIFICACIÓN DE CASOS DE USO

Tabla 2. CASO DE USO 1: Cargar tablatura

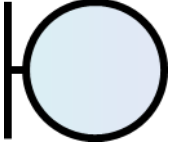


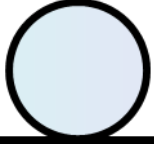


CASO DE USO 1: Cargar tablatura					
Actor(es)	Usuario				
Propósito	Cambiar la Tablatura cargada.				
Precondición(es)	Que TabDig esté abierta.				
Flujo Principal	Acciones Actor(es)	Respuestas del Sistema			
	1. El usuario ejecuta el Software.	2. La aplicación le informa que se ha cargado la tablatura por defecto.			
	3. El usuario clickea en la lista de demos o en el botón Mis tablaturas.	4. La aplicación despliega una lista con los posibles demos / abre el explorador de archivos.			
	5. El usuario selecciona la tablatura que desea.	6. La aplicación analiza el archivo y por medio de un mensaje le muestra al usuario que ha sido cargada la tablatura.			
Flujo Alternativo	5. El usuario selecciona la tablatura que desea	6. La aplicación analiza el archivo y detecta que no corresponde a una tablatura y lanza nuevamente el explorador de archivos.			
Poscondición(es)	La aplicación establece y actualiza el nombre de la tablatura y el número de tramos que esta tendrá.				
OBJETOS QUE INTERVIENEN (Borde - Control - Entidad)					
					
Mensaje de Información	Listar Tablaturas	Consultar Tablaturas	Tablaturas	Validar Tablaturas	Mensaje Respuesta

Tabla 3. CASO DE USO 2: Reproducción de Tablatura



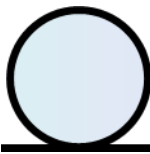
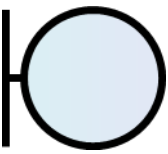
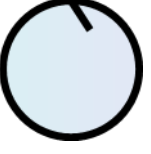
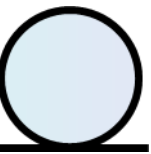
CASO DE USO 2: Reproducción de Tablatura					
Actor(es)	Usuario				
Propósito	Controlar la reproducción de la simulación				
Precondición(es)	La tablatura debe de estar cargada y lista para reproducirse.				
Flujo Principal	1. El usuario inicia la reproducción clickeando en el botón de reproducción.		2. El sistema inicia la simulación teniendo en cuenta los parámetros que se han estipulado por defecto.		
	3. El usuario pausa la reproducción clickeando nuevamente en el botón de reproducción.		4. El sistema detiene la simulación en el último tramo que ha sido reproducido		
Flujo Alternativo	3. El usuario reinicia la simulación clickeando en el botón de reinicio.		4. El sistema cambia el valor del tramo actual por el tramo inicial de la tablatura.		
Poscondición(es)	La aplicación reproduce las animaciones y sonidos asociados a las notas encontradas en la tablatura elegida.				
OBJETOS QUE INTERVIENEN (Borde – Control – Entidad)					
					
Botón Reproducción	Control Simulación	Parámetros Simulación	Botón Reinicio	Control Simulación	Tramo Simulación

Tabla 4. CASO DE USO 3: Desplazamiento a través de los tramos de reproducción

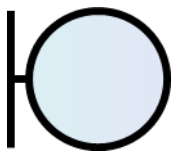

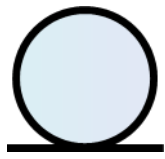
CASO DE USO 3: Desplazamiento a través de los tramos de reproducción		
Actor(es)	Usuario	
Propósito	Desplazarse a través de la simulación utilizando la barra de simulación utilizando el indicador de tramos.	
Precondición(es)	La tablatura debe de estar cargada y lista para reproducirse.	
Flujo Principal	1. El usuario mueve el cabezal de la barra de progreso para posicionarse en un tramo diferente.	2. La aplicación actualiza el tramo actual de la simulación conforme el cabezal se desplaza.
Poscondición(es)	La aplicación actualiza el tramo actual durante la simulación e inmediatamente la simulación se actualiza.	
OBJETOS QUE INTERVIENEN (Borde – Control – Entidad)		
 <p>Barra de progreso</p>	 <p>Control Simulación</p>	 <p>Tramo Simulación</p>

Tabla 5. CASO DE USO 4: Cambiar la velocidad de la Simulación

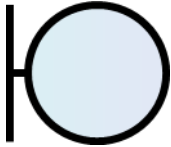

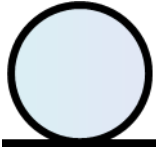
CASO DE USO 4: Cambiar la velocidad de la Simulación		
Actor(es)	Usuario	
Propósito	Transformar la velocidad de la simulación por medio de la unidad de medida bpm.	
Precondición(es)	La tablatura debe de estar cargada y lista para reproducirse.	
Flujo Principal	1. El usuario mueve el cabezal de la barra de velocidad para cambiar los bpm de la simulación.	2. La aplicación actualiza la velocidad de la simulación conforme el cabezal se desplaza.
Poscondición	La aplicación actualiza la velocidad de reproducción de la tablatura mientras esta sigue ejecutándose.	
OBJETOS QUE INTERVIENEN (Borde – Control – Entidad)		
		
Barra de Velocidad	Control de Simulación	Velocidad Simulación

Tabla 6. CASO DE USO 5: Cambiar el Volumen de los sonidos

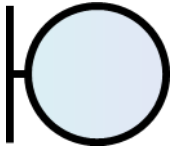

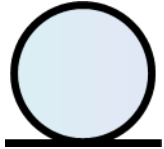
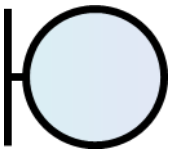

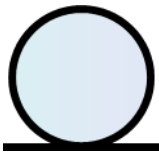
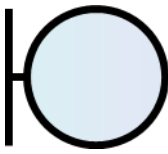

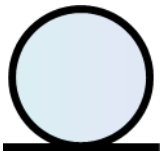
CASO DE USO 5: Cambiar el Volumen de los sonidos		
Actor(es)	Usuario	
Propósito	Cambiar el valor de los decibeles del sonido para los sonidos de reproducción dentro de la simulación.	
Precondición(es)	La tablatura debe de estar cargada y lista para reproducirse.	
Flujo Principal	1. El usuario mueve el cabezal de la barra de volumen para cambiar el valor de los decibeles de la simulación.	2. La aplicación actualiza el valor de los decibeles de la simulación mientras se sigue reproduciendo.
Poscondición	La aplicación actualiza el volumen de la reproducción de los sonidos mientras se está reproduciendo la simulación.	
OBJETOS QUE INTERVIENEN (Borde – Control – Entidad)		
 <p>Barra Volumen</p>	 <p>Control Simulación</p>	 <p>Volumen Simulación</p>

Tabla 7. CASO DE USO 6: Creación de bucles de repetición

CASO DE USO 6: Creación de bucles de repetición					
Actor(es)	Usuario				
Propósito	Crear bucle de repetición definido por el usuario.				
Precondición(es)	La tablatura debe de estar cargada y lista para reproducirse.				
Flujo Principal	<p>1. El usuario Clickea en el botón de control de inicio para marcar el inicio del bucle.</p>	<p>2. El sistema Dibuja la marca de inicio de bucle y guarda la posición del tramo donde está el cabezal de simulación en ese momento.</p>			
	<p>3. El usuario Clickea en el botón de control de fin para marcar el inicio del bucle.</p>	<p>4. El sistema dibuja la marca de fin de bucle y guarda la posición del tramo donde está el cabezal de simulación en ese momento.</p> <p>5. El sistema inmediatamente cambia el valor del tramo actual por el del ciclo de inicio y el ciclo comienza a ejecutarse.</p>			
OBJETOS QUE INTERVIENEN (Borde – Control – Entidad)					
					
Marca Inicio	Control Marca	Inicio Bucle	Marca Fin	Control Marca	Fin Bucle

4.9 FORMATO DE PRUEBA DE USABILIDAD Y UTILIDAD

Tabla 8. Formato de prueba de usabilidad y utilidad

PRUEBA USABILIDAD Y UTILIDAD TABDIG	
Datos personales	<p>Nombre: _____</p> <p>Edad: _____</p> <p>Ocupación: _____</p> <p>Estudios académicos: _____</p> <p>Tiempo (meses) de practica con la guitarra: _____</p> <p>Tiempo (meses) de conocer las tablaturas: _____</p>
Exactitud	<p>¿Cuáles fueron los errores que el usuario cometió?</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>¿Cuántos errores cometió el usuario durante su experiencia con el software? _____</p> <p>¿Cuántos de estos errores fueron reiterativos? _____</p> <p>¿En alguno de estos errores el usuario manifestó confusión? ¿Cuál? _____</p> <p>Después de dada la explicación de los errores anteriores. ¿Hubo alguno que para el usuario no haya sido aclarado? ¿Cuál? _____</p>
Tiempo	<p>¿Cuánto tiempo demoro la prueba del usuario?</p> <p>_____</p>
Recuerdo	<p>Pasado 10 minutos ¿Que explicación y recorrido mental hizo el usuario de la aplicación? (escala de 1 a 5) _____</p>
Utilidad	<p>¿Qué tan útil representa para el usuario el uso de la aplicación en su estudio de las tablaturas? (escala de 1 a 5) _____</p>
Recomendaciones	<p>¿Qué recomendaciones le haría a la aplicación para que mejore su desempeño?</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p>
Sensación final	<p>Cuál es la sensación final despertada por el usuario? _____</p>

4.10 RESULTADO DE PRUEBA DE USABILIDAD Y UTILIDAD

Figura 4. Gráfico de los errores más frecuentes de los usuarios.

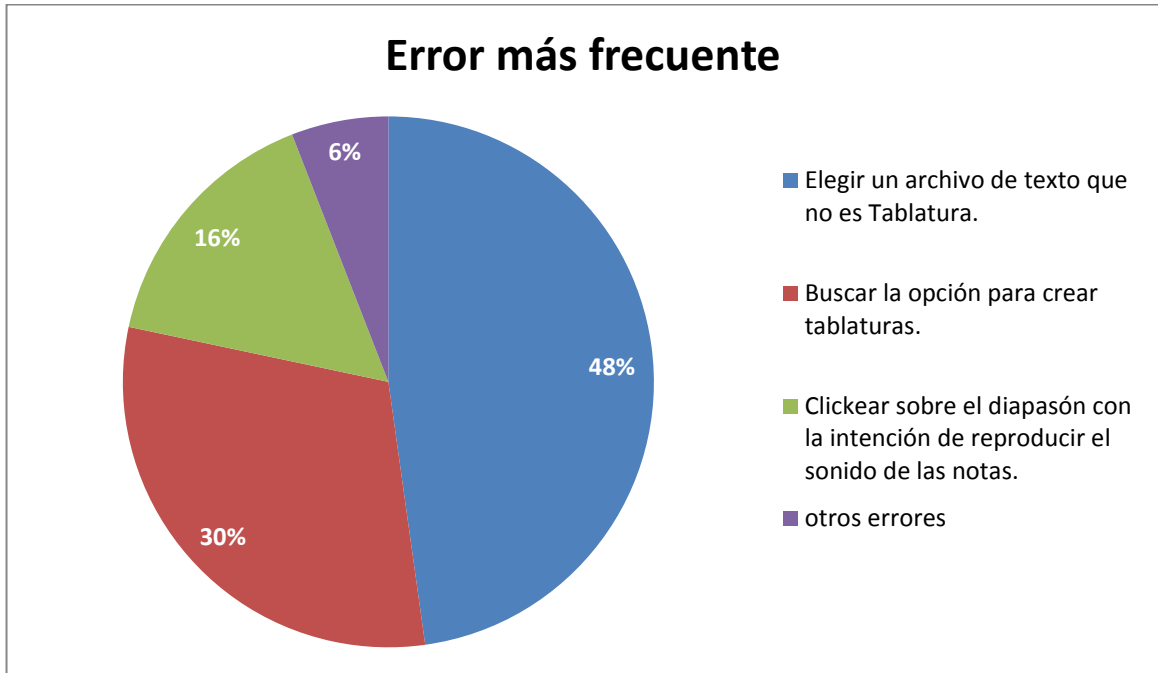


Figura 5. Gráfico con los errores de mayor confusión para los usuarios.

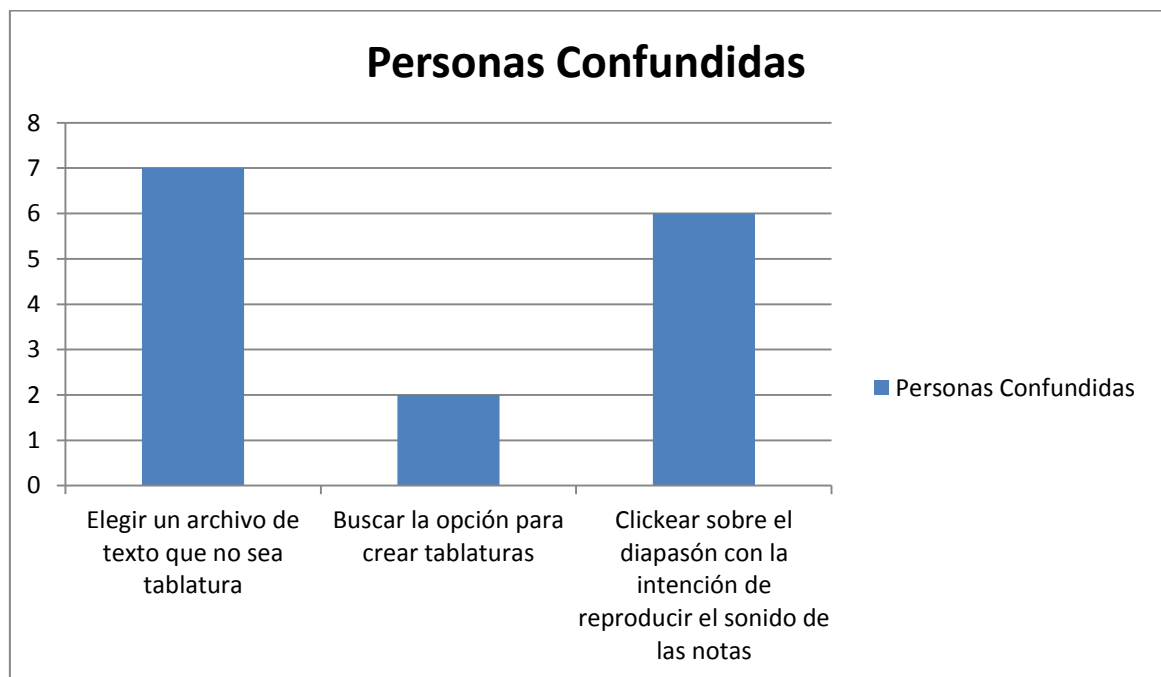


Figura 6. Grafica con los tiempos de prueba de cada uno de los usuarios

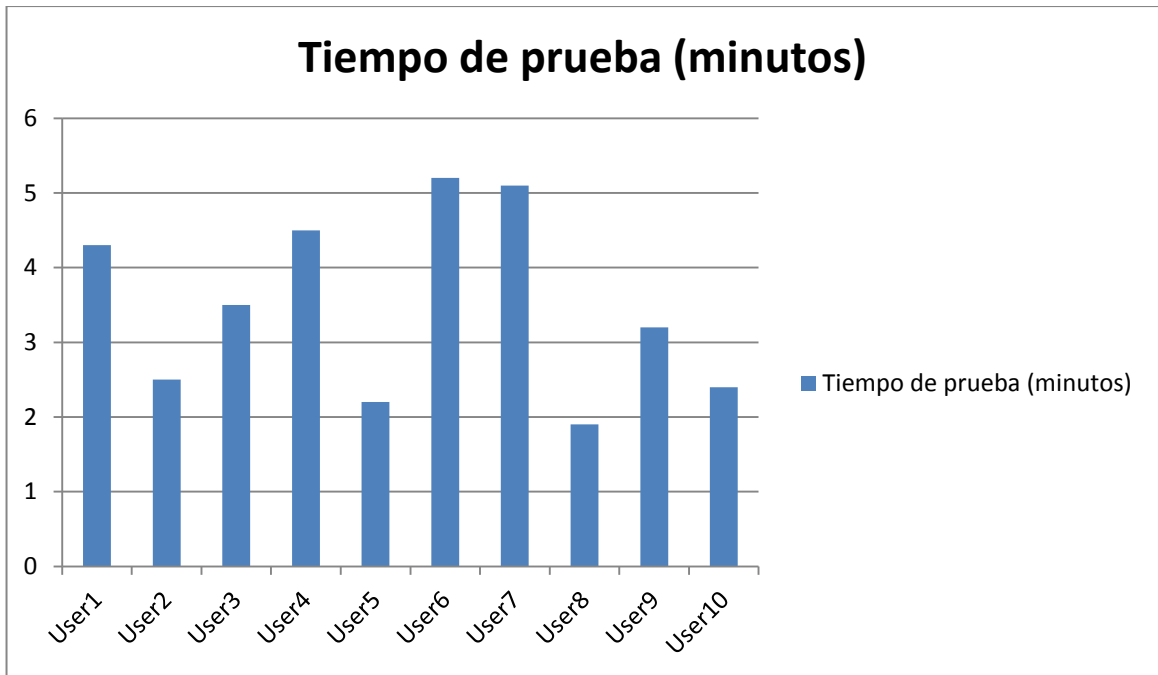


Figura 7. Grafico del impacto de memoria rápida de los usuarios.

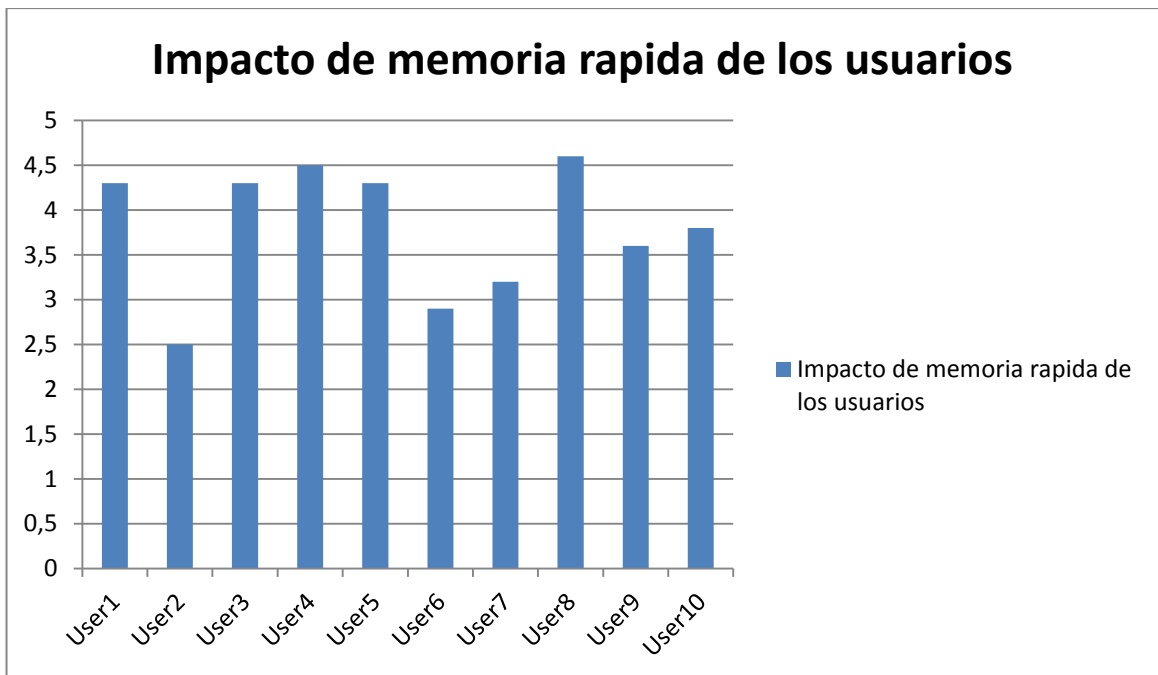
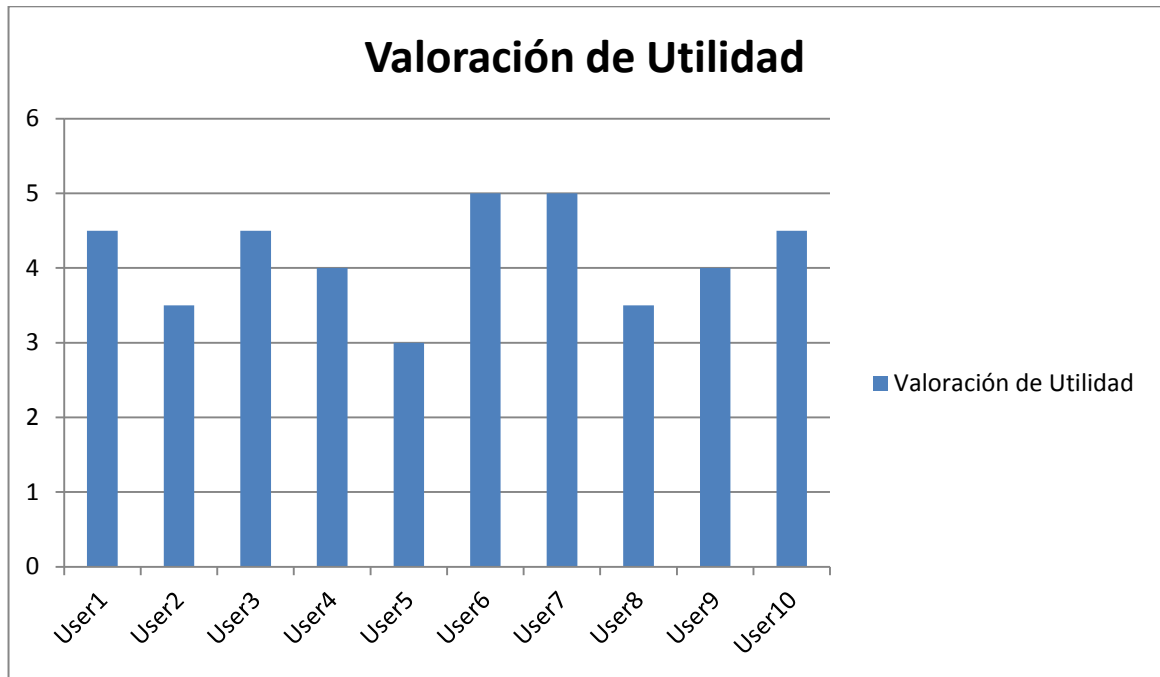


Figura 8. Gráfico con la valoración de utilidad por parte de los usuarios.



5. CONCLUSIONES

- La metodología de desarrollo por incrementos es sumamente eficiente para identificar los requerimientos del software directamente de las necesidades de los usuarios a través de su sistema de pruebas y realimentación de requerimientos.
- Las pruebas finales demostraron que las sensaciones de los usuarios al utilizar la aplicación fueron altamente positivas. Se cumplieron los objetivos principales y se entregó una aplicación escalable que permitirá perfeccionarse a medida que se desarrollen nuevas versiones.
- El desarrollo de aplicaciones altamente intuitivas y de gráficos de alta calidad, permite ampliar la población de usuarios incluyendo aquellos que sus conocimientos de informática y tecnología sean muy limitados.
- El planteamientos de proyectos que involucre la investigación de otras áreas del conocimiento como la música y el diseño intuitivo favorece la interacción de los estudiantes con comunidades ajenas a su campo de acción permitiendo elaborar proyectos interdisciplinarios y con múltiples áreas de interés.

REFERENCIAS:

BÁEZ Manuel. Cómo afinar una guitarra: todo sobre la afinación [en línea] [citado 13 de julio de 2016] Disponible en: www.guitarristas.info/tutoriales/como-afinar-guitarra-todo-sobre-afinacion/2787#

ESCRIBIR CANCIONES. bpm proviene de las siglas "Beat por Minute", que en castellano significa "golpes por minuto". Sirve para establecer la duración y/o velocidad de las figuras musicales con exactitud. [en línea] [citado 15 de julio de 2016] Disponible en: www.escribircanciones.com.ar/icomocomponer-musica/217-ique-es-el-tempo-bpm-y-como-afecta-la-musica.html

FERREYRO, J. Blogs, una Reflexión Compartida sobre Buenas Experiencias de Enseñanza. Revista Electrónica de Tecnología Educativa, 24.2007. [en línea] [citado 10 de julio de 2016] Disponible en: <http://edutec.rediris.es/Revelec2/revelec24/pdf/Edutec24-JFerreyro-Blogs.pdf>

GUI Es la interfaz gráfica que un software, aplicativo o programa le enseña al usuario para interactuar con el. Referencia: [en línea] [citado 24 de julio de 2016] Disponible en: archwindevelopment.techtarget.com/definition/GUI

HCI conjunto de técnicas y principios relacionados con el diseño desarrollo y estudio de los fenómenos que rodean los sistemas informáticos para uso humano. referencia: albertolacalle.com/hci.htm

IMPORTANCIA.ORG. Importancia de la Metodología.[en línea] [citado 15 de julio de 2016] Disponible en: <http://www.importancia.org/metodologia.php>

MAESTROS OF THE GUITAR (2005). The history of Tablature Notation. 2000 [en línea] [citado 13 de julio de 2016] Disponible en: <http://www.maestros-of-the-guitar.com/tablature.html>

MANIFEST File es un archivo que especifica contenido en un archivo jar. En este archivo se define cual es la clase principal de la aplicación. referencia: es.wikipedia.org/wiki/Manifest

MATANYA Ophee. A History of Lute TABlature Transcriptions.2000 [en línea] [citado 12 de julio de 2016] Disponible en: <http://www.guitarandluteissues.com/trans/trans.html>

NAVARRO, José Luis. Método Postmoderno de la Guitarra en: Curso de guitarra Clásica 2007 [en línea] [citado 10 de julio de 2016] Disponible en: <http://cursodeguitarraclasicaenlinea.blogspot.com>

OSORIO Frank Octavio. Curso Básico de Guitarra Popular Tópicos Elementales para su Aprendizaje. [en línea] [citado 10 de julio de 2016] Disponible en: <http://www.mundomanuales.com/manuales/3037.pdf>

UML Es un lenguaje grafico para visualizar y documentar cada una de las partes que comprende el desarrollo de software. [en línea] [citado 23 de julio de 2016] Disponible en: Referencia:users.doc.uchile.cl/~psalinas/uml/introducción.html

WIKIPEDIA. Expresiones Regulares es una secuencia de caracteres que forma un patrón de búsqueda, principalmente utilizada para la búsqueda de patrones de cadenas. [en línea] [citado 23 de julio de 2016] Disponible en: https://es.wikipedia.org/wiki/Expresi%C3%B3n_regular