

Instituto Tecnológico de Costa Rica

Administración de Tecnologías de Información

TI-2402 Algoritmos y Estructuras de Datos

Prof: Andréi Fuentes Leiva.

Primera Tarea Programada en Java

Daniel Navarro Romero
carné: 2014044199

Ernesto Brenes Calderón
carné: 2014044135

Steven Rojas Cortes
carné: 2014071312

Tabla de Contenidos:**Índice de contenido**

Descripción del problema.....	3
Diseño del programa.....	4
Preparando el entorno:.....	4
Diseño de interfaz:.....	4
Librerías utilizadas.....	6
Librería de interfaz gráfica java swing y awt:.....	6
Librería para la extracción de información mp3agic:.....	6
Librería java io:.....	7
Librería de reproducción de audio BasicPlayer3:.....	7
Análisis de Resultados.....	7
Objetivos Alcanzados:.....	7
Objetivos no alcanzados:.....	9
Manual de Usuario.....	9
Conclusión personal.....	9

Descripción del problema

Esta tarea programada presenta varios retos que superar, pero mas que retos son oportunidades para fortalecer el conocimiento mediante la práctica. Uno de los principales problemas es entender completamente la lógica de programación orientada a objetos y aplicarla a las estructuras de datos, las cuales ya por si solas representan una dificultad en aspectos como orden, organización, control, seguimiento de datos, entre otras.

El uso de nuevas herramientas como el administrador de control de versiones Git y ANT para crear los ejecutables e incluso el mismo lenguaje java representan retos en esta tarea programada. Esto debido a que ninguno de los integrantes del grupo tiene experiencia en el uso de estas herramientas. Además como la tarea programada es crear un reproductor de mp3 implica el manejo de archivos multimedia y la creación de una interfaz gráfica, es casi inevitable el uso de librerías, tanto las propias de java como las librerías externas.

La labor de aprender a utilizar una librería agrega más dificultad a la tarea programada, pero una vez aprendido como se usa, simplifica en gran medida la programación. Una de las librería más esencial es la librería para reproducir canciones, la cual es la de mayor complejidad.

Aunque la interfaz no presenta mayor problema por la existencia de IDEs que permiten crearla, como Netbeans, ya se había experimentado problemas de incompatibilidad entre la versión de Netbeans, la versión del JDK y ubuntu.

Otra dificultad es el uso de un nuevo sistema operativo, ubuntu. Antes de empezar la tarea programada ya existían problemas al usar este nuevo sistema. Una de las máquinas no lo ejecutaba del todo y otra no reproducía audio, lo cual es importante, tratándose de un programa que reproduce de música.

Diseño del programa

Preparando el entorno:

A pesar que cada integrante ya había intentado preparar el entorno con anterioridad, aun existían varios problemas y aspectos a tomar en cuenta. Como primer aspecto está el funcionamiento óptimo de Ubuntu, el cual era posible virtualizar en las computadoras usadas, pero no había fluidez a la hora de usarlo y esto retardaba mucho la programación. Por esta razón se decidió hacer una instalación limpia del sistema operativo, pero surgieron los siguientes problemas en dos de las cuatro computadoras usadas:

Problema 1: La computadora de escritorio ejecutaba Ubuntu hasta el inicio de sesión y se pegaba.

Solución 1: Se modificó el arranque de Ubuntu lo cual permitió iniciar sesión y cambiar el controlador de vídeo por uno privativo.

Problema 2: En una de las portátiles en ocasiones no se reconocía la tarjeta de audio.

Solución 2: No se pudo solucionar el problema al no existir controladores privativos o uno que se adaptará óptimamente.

El segundo aspecto a considerar fue el uso de IDEs porque Netbeans no funciona fluidamente en ninguna de las máquinas. La incompatibilidad se solucionó instalando la última versión de Netbeans y el JDK, pero mejor se decidió usar el editor de texto Sublime Text 2.

Diseño de interfaz:

Toda la interfaz se diseñó en un solo archivo .java y se utilizó una sola ventana. La interfaz hereda de la clase JFrame que permite crear ventanas. Esto se hizo con el fin de obtener acceso directo a los métodos y atributos de la clase JFrame para reducir la cantidad de código usado. Por ejemplo:

```
En vez de usar: JFrame ventana = new JFrame("titulo de ventana") ;  
                ventana.setSize(ancho, largo);
```

```
Simplemente se usa: setSize(ancho, alto);
```

Librerías utilizadas

Librería de interfaz gráfica java swing y awt:

Son librerías que permiten crear objetos gráficos, agregarles o capturar eventos, ingresar, obtener y mostrar información. Estas librerías son propias de java, es decir, no se tiene que descargar por separado. Entre los objetos usado de estas librerías están:

1. JFrame: permite crear ventanas. Se uso heredando la clase interfaz de la clase JFrame.
2. JButton: permite crear botones. Se usaron métodos para especificar el tamaño de los botones.
3. JLabel: sirve para crear etiquetas, ya sea con texto o imágenes. Se utilizaron métodos para cambiar el tamaño, color y tipo de letra.
4. JTextField: funcionan para crear campos de texto. Se usaron para que el usuario digite información. Mediante métodos se extrajo información y también se mostró información.
5. JComboBox: sirve para crear campos de textos con opciones desplegables. Solo se usó un JComboBox para mostrar las opciones de búsqueda (Nombre, Artista, Álbum, Género). Funciona mediante arreglos.
6. JList: sirve para crear campos de lista. Se utilizaron métodos para extraer el dato seleccionado, el índice del dato seleccionado, agregar datos al final y eliminar datos. En la interfaz se implementaron dos campos de lista, uno para mostrar la lista de canciones agregadas y otro para mostrar los resultados de la búsqueda.
7. Container: es el contenedor que permite agregar los elementos a la ventana con un alto grado de precisión, colocándolos mediante ejes x y ejes y.
8. Mensajes de dialogo: se usaron para mostrar errores o información.

Librería para la extracción de información mp3agic:

Esta librería permite leer y manipular los datos de las canciones o la etiquetas ID3, también llamados metadatos. Entre los datos están el titulo de la canción, artista, álbum, género, año, duración, entre otras. Se pueden obtener más datos de las etiquetas ID3, pero en lo que respecta a este proyecto, solo se usó utilizaron las antes mencionadas.

En esta tarea programada se importó mp3agic en el archivo Interfaz.java. El código para ejecutar esta librería responde después de presionar el botón agregar y haber seleccionado la canción. Los datos extraídos son enviados como parámetros mediante el método agregarCancion() de la clase listaEnlazada. Luego este método envía los datos al constructor del nodo y se añaden como atributos del nodo creado. Esta librería solo se utiliza cuando se agregan canciones.

Librería java io:

Esta es la librería estándar de entrada y salida de java. Más que todo se utilizó en la clase ReproducirMP3 a lo interno de la librería BasicPlayer3, que se explica a continuación.

Librería de reproducción de audio BasicPlayer3:

BasicPlayer3 es la encargada de realizar la tarea de reproducir, pausar, continuar y detener las canciones. Cuando se selecciona una canción y se oprime el botón de play, se ejecuta el método buscarNodo con el nombre de la canción seleccionada como parámetro. Este método busca el nodo que contiene el nombre de canción que recibió, cuando lo encuentra lo establece como nodo actual y retorna true para que el código del botón play siga trabajando. El resto del código del botón play envía la ruta del nodo actual (mediante el método abrirFichero de un objeto declarado e inicializado con anterioridad) a la clase ReproducirMP3 para ser reproducido.

Análisis de Resultados**Objetivos Alcanzados:**

1. Cumplimiento de todas las funcionalidades del reproductor:
 - Reproducción de música: el programa final puede reproducir canciones en formato MP3. Y el usuario puede reproducir cualquier canción de las que haya agregado e incluso puede ir agregando más canciones mientras reproduce.
 - Manejo de información de canciones: el programa es capaz de mostrar los datos de cada canción según se vayan seleccionando. Además aunque existan dos o más canciones con todos los datos iguales, son consideradas diferentes gracias a la programación orientada a objetos. Por esta razón lo que se haga con una canción no afecta a la otra.
 - Gestión de la biblioteca de música: la gestión de la biblioteca permite agregar canciones una a la vez, modificar los datos nombre de canción, artista, álbum, género, año. También es posible eliminar canciones una por una y la consulta de los datos es automática al seleccionar una canción.

- Búsqueda avanzada de canciones: se logró obtener una búsqueda avanzada de canciones muy intuitiva y fácil de usar. Además permite cuatro tipos de búsqueda, por nombre de canción, por artista, álbum y género. En los resultados de la búsqueda se incluye todas las canciones que coinciden con el dato ingresado. Esta búsqueda es capaz de reconocer mayúsculas y minúsculas.

2. Utilización de lo aprendido en clase sobre estructuras de datos:

La implementación de la lista doblemente enlazada por nodos fue un gran logro, ya que se presentaron muchas dificultades en el proceso. Esto se debió a que la mayoría de las funcionalidades dependían fuertemente de la buena implementación de la lista enlazada. Aunque esta fue la única estructura de datos elaborada manualmente requirió de un conocimiento absoluto de los conceptos enlazada simple, doblemente enlazada, circular, los conceptos siguiente, anterior, actual, primero, ultimo, sobre todo a la hora de crear un método.

3. Mayor conocimiento en la programación orientada a objetos y java:

Un objetivo que se alcanza con cada tarea realizada es la adquisición de nuevo conocimiento o fortalecimiento del que ya se tiene. En esta tarea este objetivo se cumplió a grandes rasgos porque se obtuvo conocimiento en el uso de nuevas herramientas, fortalecimiento sobre el conocimiento del lenguaje de programación java, estructuras de datos y la programación orientada a objetos.

Con respecto a la programación orientada a objetos se obtuvo un concepto más claro de lo que son datos de referencia y datos primitivos o nativos, debido a la cantidad de comparaciones y asignaciones que se realizaron.

4. Conocimiento de nuevas herramientas: esta tarea programada permitió aprendizaje sobre el uso de nuevas herramientas como el sistema operativo Ubuntu. No solo se obtuvo conocimiento sobre el uso de estas herramientas, sino que también se pudo percibir aspectos generales de estas, como la desatención de los fabricantes de hardware respecto al sistema operativo Ubuntu o incluso las otras distribuciones de Linux

Objetivos no alcanzados:

1. No implementar el JList mediante una lista manual:

Se usaron JList para mostrar las listas de canciones y se comprendió que en su estructura interna funcionaba por medio de lista. Por esta razón se intentó hacerlo funcionar por medio de una lista por nodos o por arreglos, pero no se pudo desarrollar un método para ingresar la lista al JList. Además un punto importante que se discutió fue que se podía comprometer el funcionamiento óptimo del JList al tratar de realizar cambios en su estructura.

Otro tema respecto a los elementos de la interfaz es el JComboBox, el cual funciona por medio de arreglos. Solo se implemento un JComboBox en la tarea programada y se pudo haber desarrollado por medio de arreglos creados manualmente, pero al ser un solo arreglo el que se necesito y este no requiere de métodos, ya que siempre mantiene su estructura, es decir, nunca se agregan, se modifican o se eliminan elementos del JComboBox.

Por lo tanto ambos elementos se implementaron de la forma estándar o predeterminada.

2. No modificar las etiquetas ID3 propias de cada canción:

Aunque los datos de las etiquetas ID3 de cada canción se añaden a los nodos de la lista como sus atributos y cuando se modifica los datos de una canción se hace a dichos atributos y no a las etiquetas ID3 directamente, se consideró importante poder modificar las etiquetas ID3. Por una cuestión de prioridad de funcionalidades y tiempo no se logró realizar la modificación de las etiquetas ID3.

3. No mostrar barra de progreso

Una característica propia de todo reproductor de música o vídeo es la barra de progreso, por eso se tomó en cuenta para el desarrolló del programa, pero está no se le dio una prioridad alta debido a que el funcionamiento del programa no depende directamente de este elemento. Por esta razón al final no se logró implementar la barra de progreso.

4. Realizar un algoritmo más eficiente para guardar las modificaciones de los datos de las canciones:

No se pudo desarrollar un algoritmo que permitiera guardar los cambios realizados a una canción sin tener que volver a insertar dicha canción al JList. En otras palabras se buscaba un método que guardara los cambios realizados a una canción y mantuviera la posición original de la canción. Una alternativa era guardar todos los datos del JList y volverlos a ingresar en el orden que se deseaba e insertar la canción modificada en su posición original. Pero esta alternativa no se implemento por considerarse ineficiente.

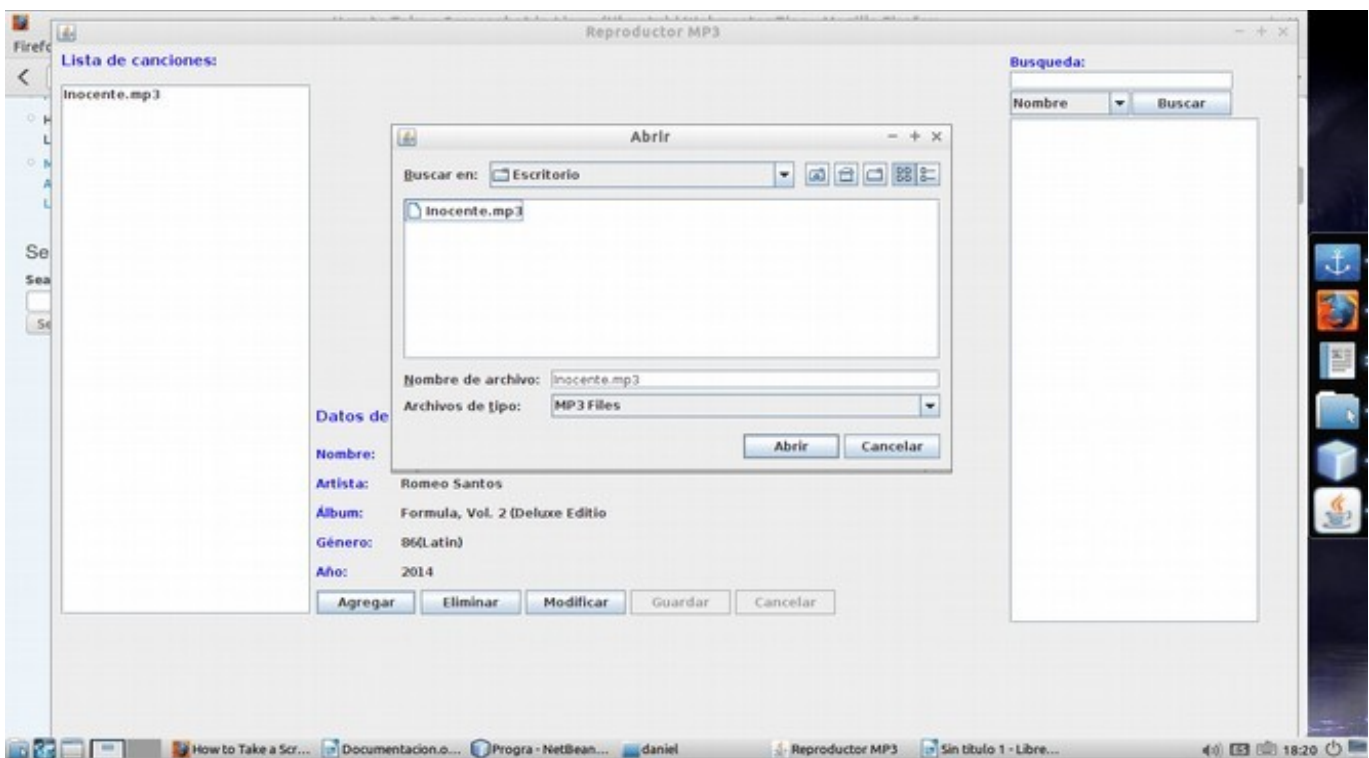
El problema más que todo se baso en el atributo nombre, ya que si este es modificado debe de actualizarse en el JList automáticamente.

Manual de Usuario

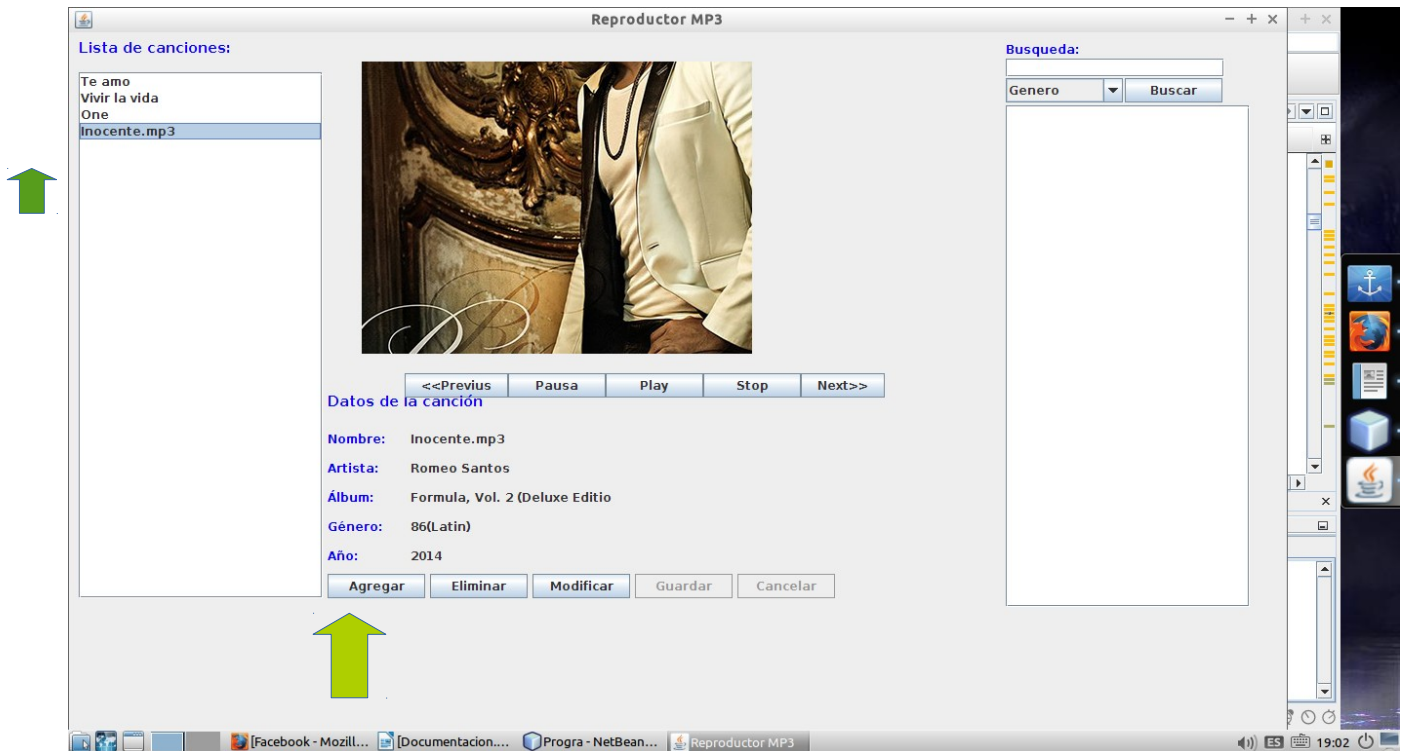
El programa realizado el cual consiste en un reproductor, consta de diversas formas de presentar las canciones agregadas al reproductor, así como la manera de manipular su información accediendo por medio de un botón si desea modificar los datos correspondientes de la canción seleccionada.

El programa presenta la opción de eliminar una canción que se encuentre registrada; además si el usuario desea buscar una canción en su lista de canciones, se le facilita la opción de búsqueda avanzada, donde se deberá elegir por el medio en que se desea buscar(nombre,artista,genero,álbum,) e ingresar el nombre.

Para agregar una canción a nuestra lista, deberás seleccionar agregar:

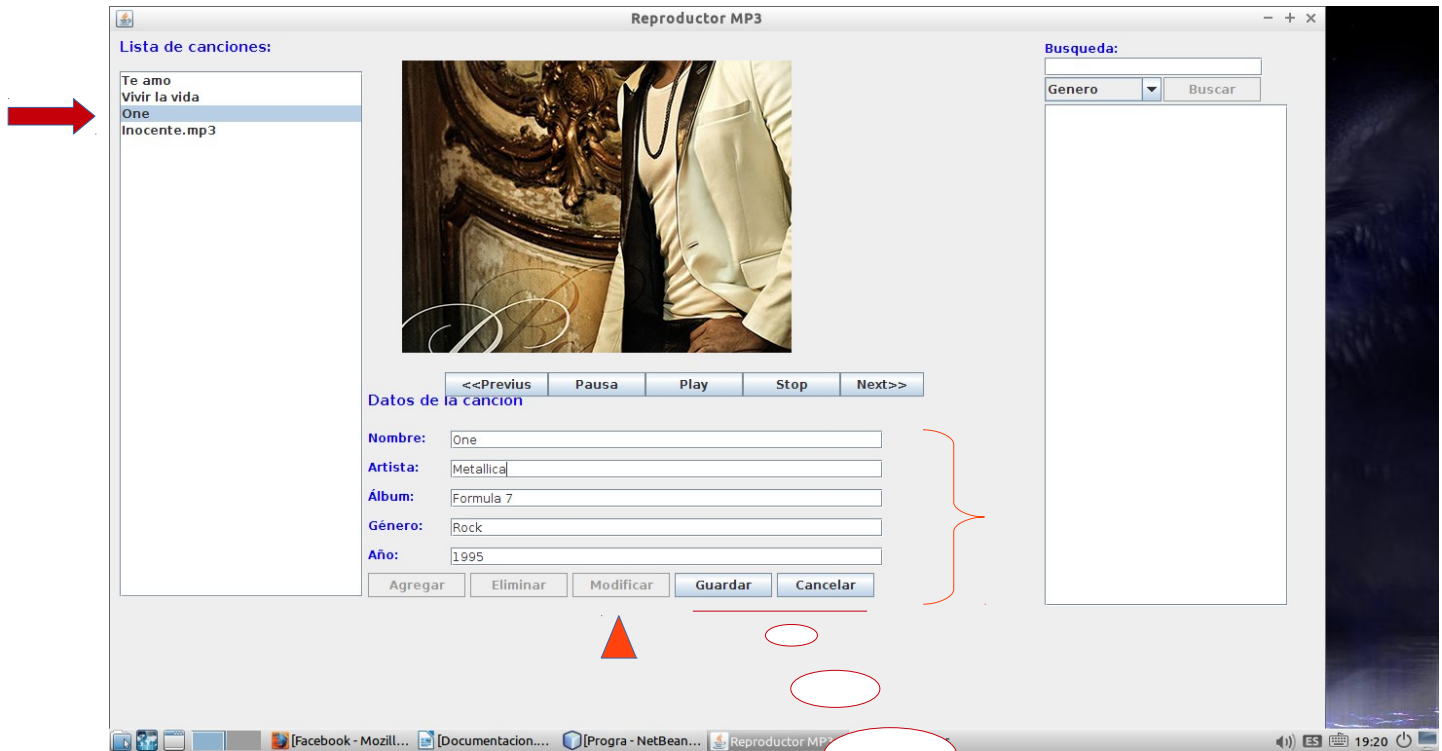


Como podemos observar en este caso nuestra canción “Inocente” seleccionado de el ordenador se agrega automáticamente a la lista de canciones.



En la imagen anterior observamos como eliminar una canción en tu lista de canciones, seleccionamos el tema a eliminar y oprimos la opción señalada.

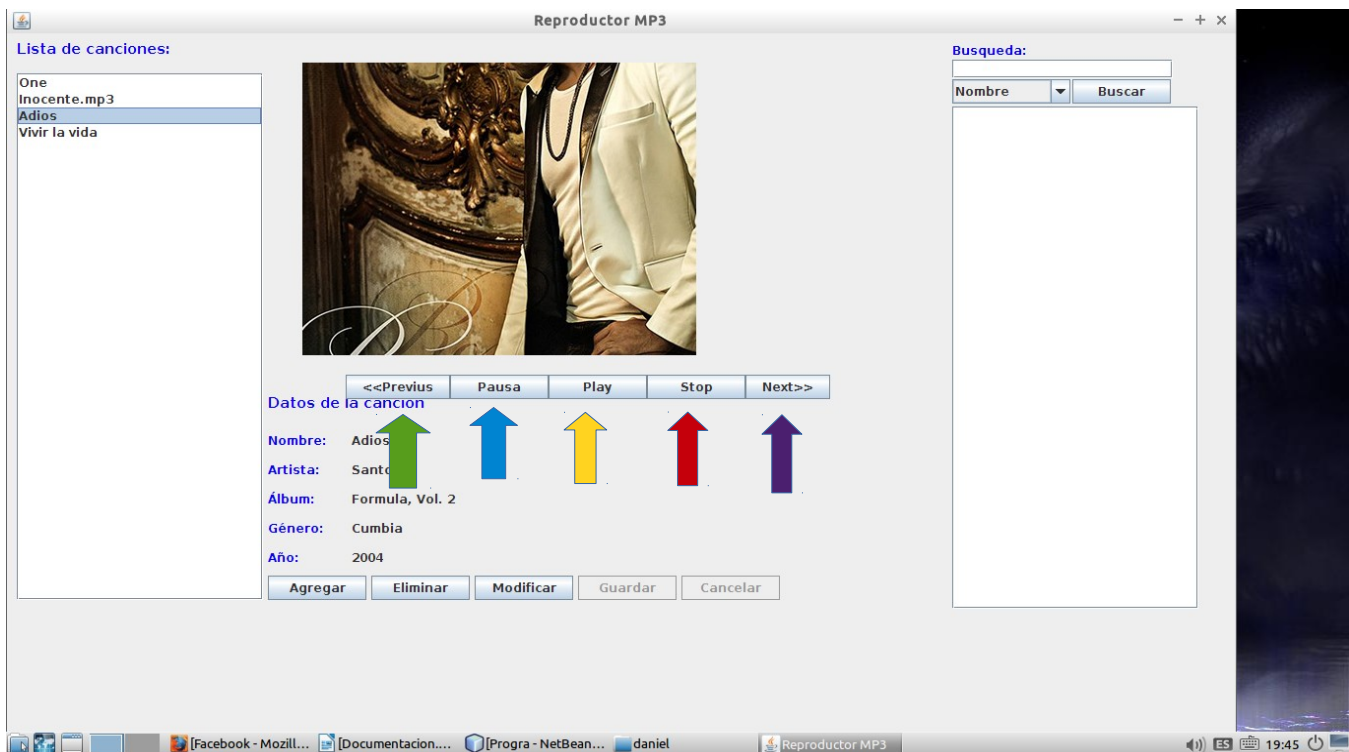
Si el usuario desea manipular cada uno de los componentes de la información de una canción, se le presenta la siguiente opción, en la cual se le ofrece modificar: el nombre, artista, álbum, género y año.



Guardar : Si desea aplicar los cambios a la información.

Cancelar: Si desea salir de la opción modificar sin aplicar cambios.

Seguidamente se presenta las opciones que permiten acceder a la herramienta de reproducir en la canción seleccionada :



La primera opción permite al usuario reproducir la canción anterior, siempre y cuando hayan mas de un tema.

La siguiente opción facilita la posibilidad de pausar nuestra reproducción, y al ser seleccionada de forma automática permite al usuario continuar con la Reproducción.

Posteriormente play, hace que el tema escogido en nuestra lista sea reproducido.

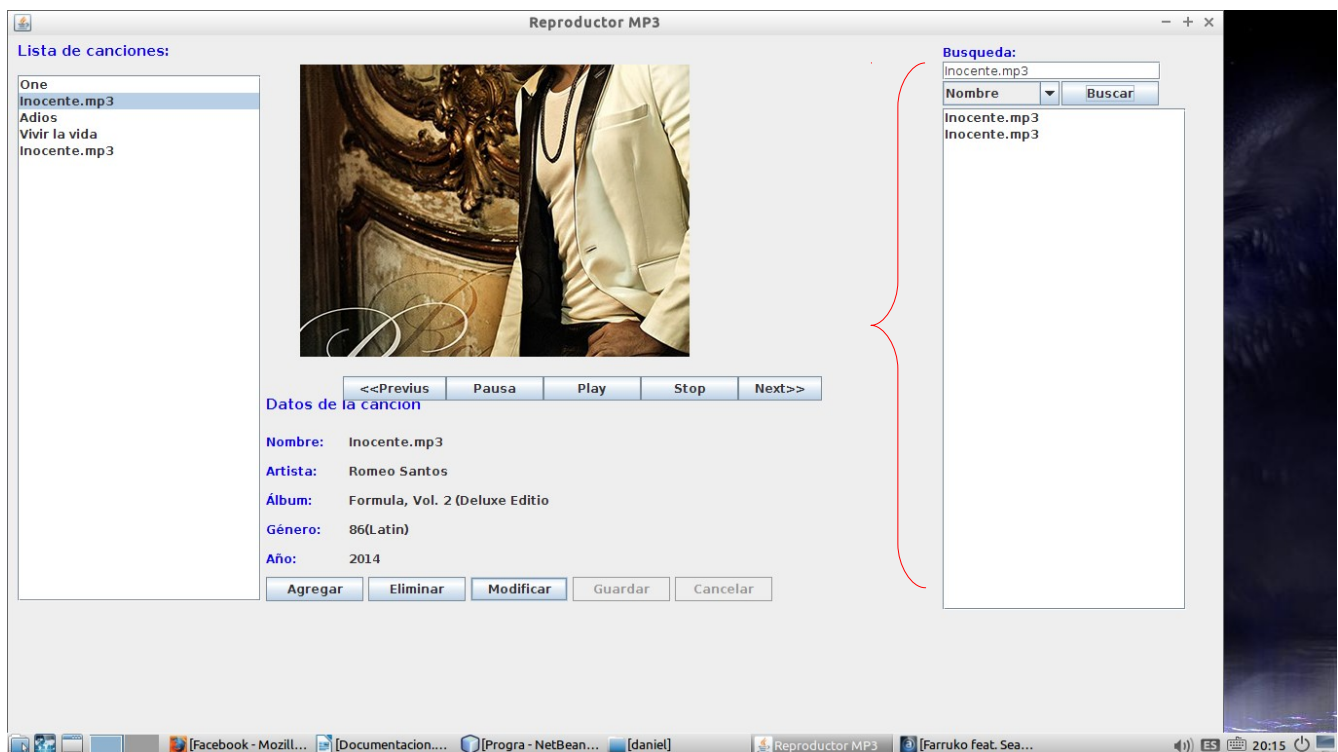
La opción stop, detiene de manera total la reproducción en curso.

Por ultimo, al presionar esta alternativa, estamos indicando que se ejecute la canción siguiente en caso de que hayan mas de un tema.

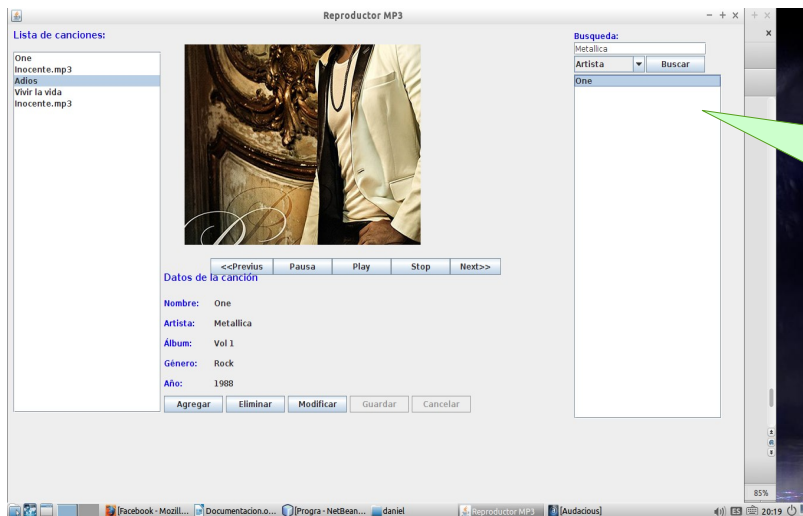
Búsqueda avanzada: Al usuario se le facilita buscar un tema en su lista de canciones, por medio de una búsqueda avanzada.

Esta herramienta consiste en buscar una canción eligiendo el tipo de dato con el que desea explorar e ingresando el dato correspondiente:

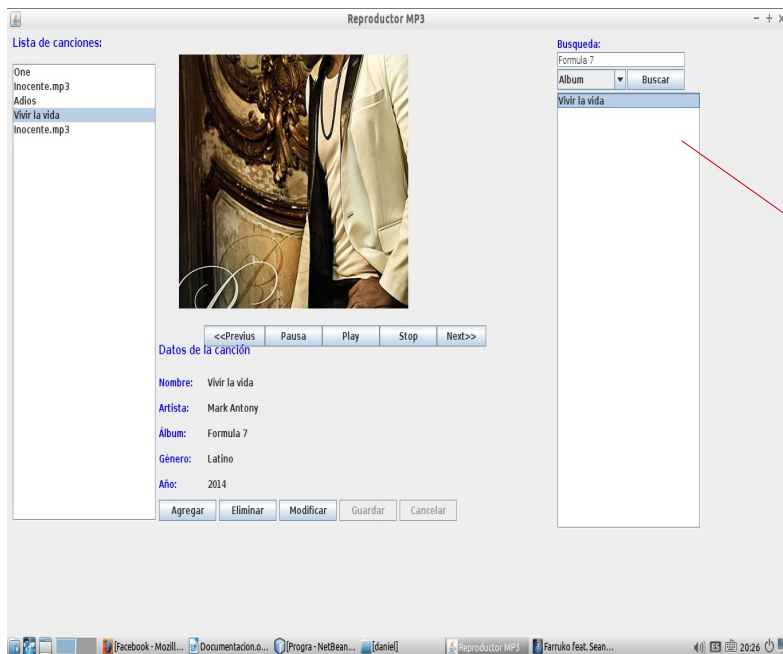
Nombre:



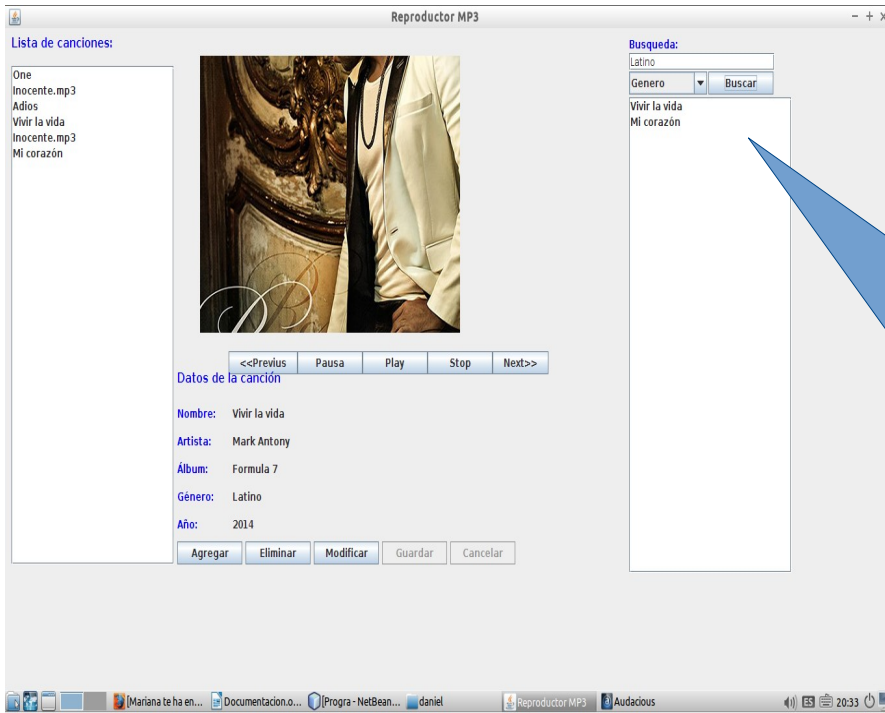
Como pueden observar en este caso tenemos dos canciones con el mismo nombre, así que de forma automática aparecen los dos temas.

Artista:

Aquí identificamos como en el tipo de dato por medio del cual queremos buscar es por artista, entonces ingresamos el artista y nos retornara el nombre de la canción

Álbum:

Al igual que el ejemplo anterior, con la diferencia de que aquí estamos implementando la búsqueda por medio del álbum

Género:

Igualmente la búsqueda por género es efectuada, sin embargo aquí tenemos presente que el tipo de género ingresado corresponde a mas de un tema en la lista de canciones.

Conclusión personal

El trabajo efectuado en esta tarea programada, permitió de forma individual y conjunta realizar distintas tareas y obtener diversidad de conocimientos, con las cuales se ha llegado al fin propuesto.

El razonamiento adquirido por parte de los miembros del grupo ha sido favorecido gracias a la utilización de distintos mecanismos que facilitaron nuestro trabajo, tales como, estructuras de datos(principalmente listas doblemente enlazadas compuestas por nodos), el lenguaje java y su diversidad de herramientas, la implementación de un entorno de desarrollo libre llamado **Netbeans**, el cual fue mayormente utilizado para facilitar el uso en las distintas librerías necesarias de importar y la verificación de errores.

Otros instrumentos que por primera vez experimentamos los cuales nos obligó a investigar y aprender variedad de operaciones, por ejemplo el sistema operativo ubuntu, el cual nos trajo dificultades a la hora de instalar programas y ejecutar diversos comandos ya que era nuevo en nuestro entorno.

Un importante argumento en este trabajo corresponde al uso de un sistema de control de versiones,en este caso (Github), el cual nos enseñó y aportó de gran ayuda a la hora de administrar nuestros avances con el manejo de repositorios, ya que estos son archivos en un sitio centralizado donde se almacena y mantiene información digital.