



Tecnológico de Costa Rica

# Instituto Tecnológico de Costa Rica

## DOCUMENTACIÓN

Escuela de Computación

Administración De Tecnologías De Información

TI-2402 Algoritmos y Estructuras de Datos

Profesor: Andrei Fuentes Leiva

Tarea Programada #1

Título: "i-MP3"

Lenguaje de Programación: Java

Integrantes:

Guillermo Ávila

Marvin Fernández Coto

Carlos Ramírez Cerdas

## Tabla de Contenido

• Descripción del problema.....	3
• Solución al problema:.....	3
• Diseño del programa:.....	3
• Decisiones de diseño: .....	3
• Diagrama lógico. ....	4
• Objetivos alcanzados: .....	5
• Objetivos no alcanzados: .....	5
• Librerías usadas .....	5
• Manual de usuario .....	6
• Conclusion personal. ....	7

### Descripcion del Problema:

Se creará un programa con el fin de reproducir musica mp3. Este contará con diferentes estructuras de datos las cuales tendrán varias funcionalidades como agregar, modificar y eliminar una canción por lo cual será necesario dominar ampliamente las estructuras de datos estudiadas en clase con sus respectivos algoritmos . El lenguaje de programación empleado será Java y se ejecutara en el sistema operativo Linux.

### Diseño del Programa

#### Decisiones de Diseño.

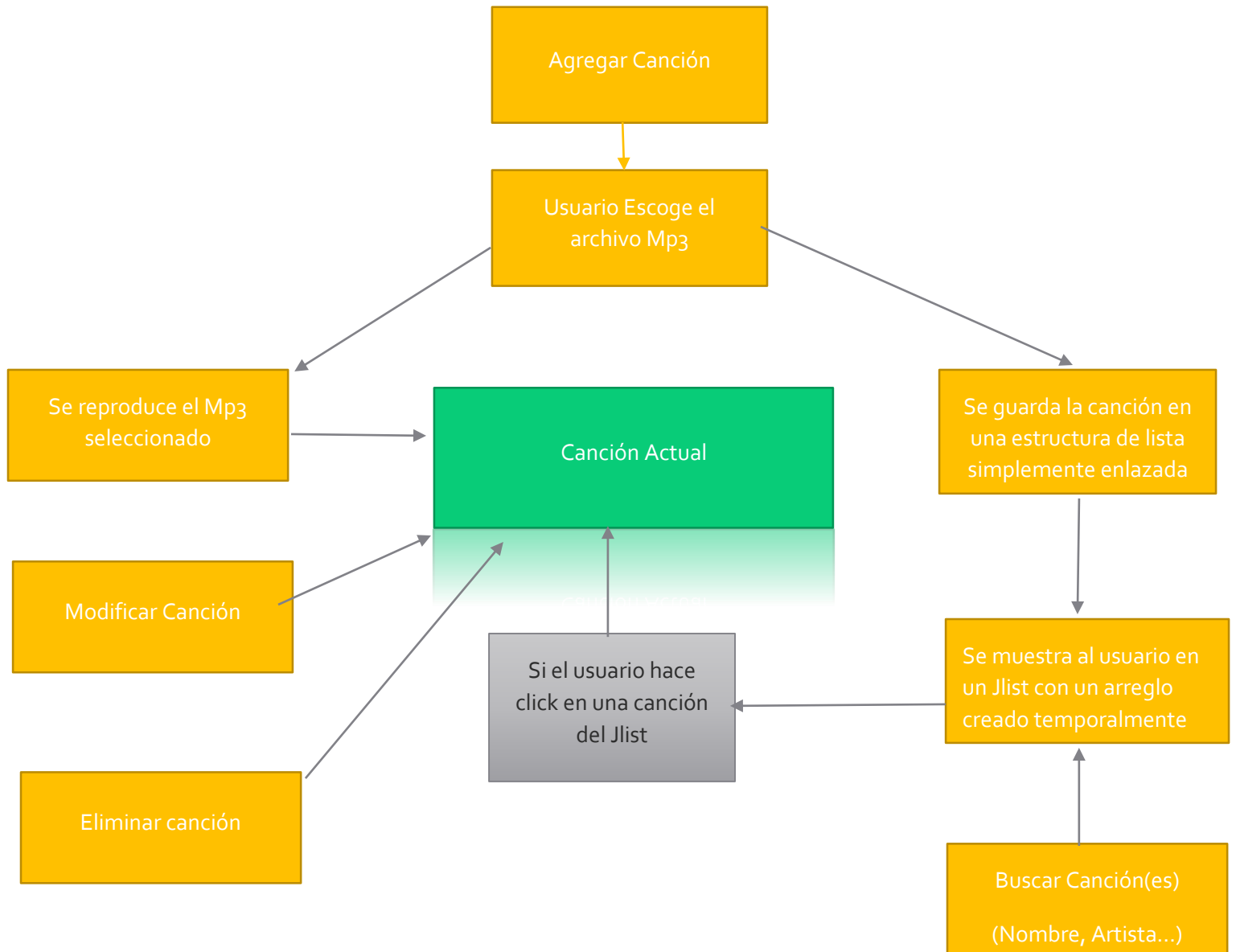
El programa contará solamente con una ventana interactiva la cual permitirá al usuario agregar, modificar, eliminar, reproducir y pausar una cancion, además contiene búsqueda avanzada por nombre(predeterminada), artista, álbum y género. Con motivo de que el programa sea amigable con el usuario y no utilizar mas ventanas ya que con lo descrito en el problema estas no serán necesarias.

#### Algoritmos Utilizados.

Todos los algoritmos están basados en la canción reproducida/consultada en tiempo real, esto se implementa con un puntero al objeto canción.

Los algoritmos de búsqueda avanzada utilizados son secuenciales, con la mejora continua se buscarán alternativas mas eficientes, en cuanto a la funcionalidad agregar nueva canción solamente se agrega al final de una estructura de datos (lista enlazada), para la función de eliminar se itera toda la lista al encontrar la canción consultada solamente la ignora por lo cual el **garbage collector** de Java se encarga de eliminarla, en cuanto a modificar canción simplemente se aplica el conocimiento adquirido en clase sobre Java el cual es **shallow copy**, es decir para la canción consultada en tiempo real solamente se utilizan los métodos de la clase canción **setArtista(nuevo artista)**, **setGenero(nuevo genero)** entre otros y para la barra de reproducción se utiliza un thread el cual "duerme" el progreso de la barra de reproducción un segundo mientras la canción avanza un segundo para que así el progreso sea mostrado en tiempo real hasta que termine la canción.

## Diagrama Lógico



Librerías Usadas:

java.io.File; (i:input(entrada),o:out(salida) de datos)

static java.lang.Thread.sleep ( Uso de para “dormir” el Thread de la barra)

java.util.logging.Level (Librería de interfaz gráfica)

java.util.logging.Logger (Librería de interfaz gráfica)

javax.swing.DefaultListModel (Librería de lista default para el Jlist(presentar nombres de canciones))

javax.swing.ImageIcon (Librería para mostrar imagen en la ventana)

javax.swing.JFileChooser (Librería para escoger archivos)

javax.swing.JOptionPane (Librería de mostrar ventanas de:  
mensaje,dialogo,pregunta.)

javax.swing.filechooser.FileNameExtensionFilter (Librería para filtrar archivos mp3)

Jaudiotagger: esta librería nos permitió extraer los meta datos de las canciones y también permite introducir los meta datos a dichas canciones, el uso que se le dio fue el de extraer los meta datos y la fotografía de la canción.

JSOUP: Permite hacer un “parser” del código html y lo extrae para poder manejarlo a placer, el uso que se le dio fue simplemente extraer la letra de algunas canciones y funciona con el titulo de la canción y el artista de dicha canción.

Jl: Permite la reproducción de MP3 de una forma sencilla y eficiente, el uso que se le dio fue reproducir las canciones de forma fácil y sencilla sin mucha dificultad de manejarla.

Graphics2D: Esta librería permite hacer “Resize” a las fotografías y crear fotografías de diferentes formatos, el uso fue extraer los datos binarios de la foto y reconstruirla para poder mostrarla en pantalla.

### Análisis de Resultados:

#### Objetivos Alcanzados:

Se logró realizar todos los objetivos descritos en las instrucciones de la tarea programada #1 entre los principales están:

- Reproducir Mp3
- Agregar Canción
- Modificar Canción
- Eliminar Canción
- Búsquedas Avanzadas
- Diseñar Interfaz Grafica

#### Objetivos No Alcanzados:

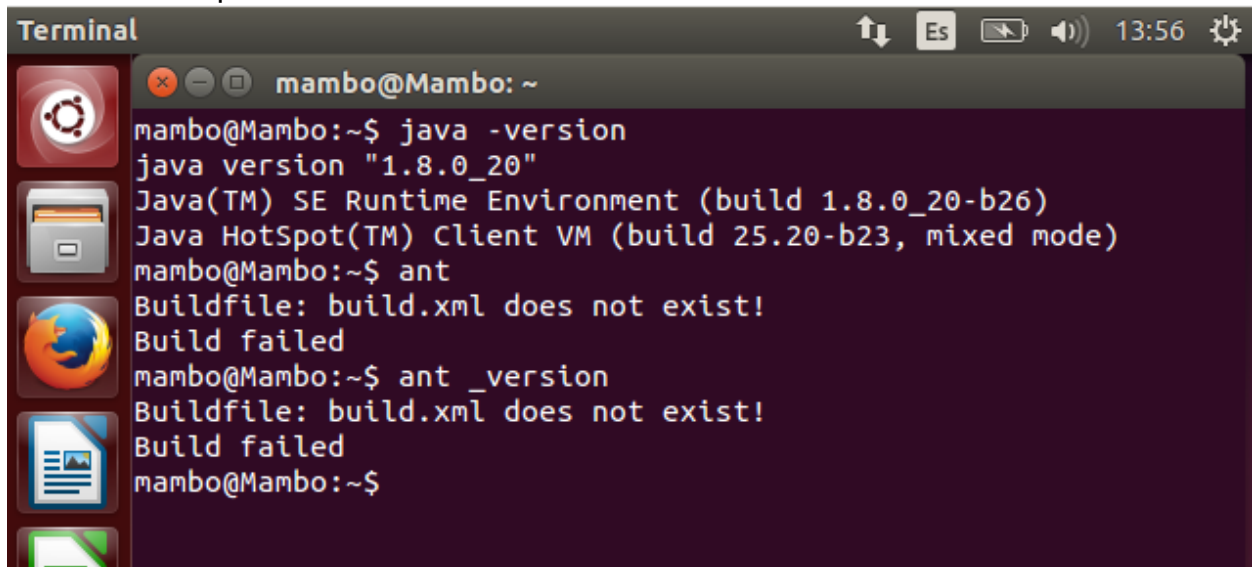
Se cumplieron todos los objetivos establecidos previamente sin embargo por cuestiones de tiempo no se pudieron lograr las siguientes extras: siguiente canción, almacenamiento en memoria secundaria, conexión a internet, las cuales no se descarta que a futuro se implementen.

## Manual de Usuario

### Requisitos del Programa

- JDK 1.8
- JAVA
- ANT 1.9.3
- Linux

Para verificar su versión de Java y Ant puede ejecutar los siguientes comandos desde su computadora.

A screenshot of a Linux terminal window titled "Terminal". The window shows a user named "mambo" at a host named "Mambo". The user enters the command "java -version", and the output shows "java version '1.8.0\_20'", "Java(TM) SE Runtime Environment (build 1.8.0\_20-b26)", and "Java HotSpot(TM) Client VM (build 25.20-b23, mixed mode)". Then, the user enters "ant", and the output shows "Buildfile: build.xml does not exist!" and "Build failed". Finally, the user enters "ant \_version", and the output shows "Buildfile: build.xml does not exist!" and "Build failed". The terminal window has a sidebar with icons for various applications like a file manager, web browser, and text editor. The top bar shows system icons like network, battery, and time (13:56).

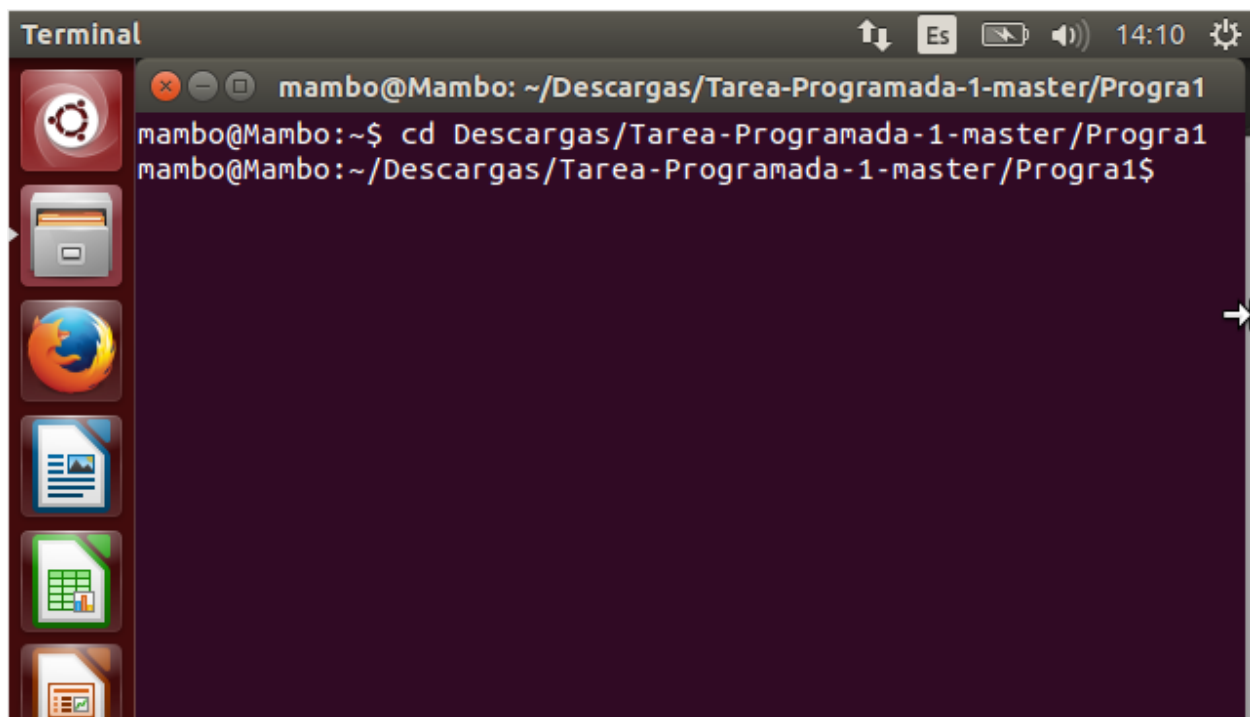
```
Terminal
mambo@Mambo: ~
mambo@Mambo:~$ java -version
java version "1.8.0_20"
Java(TM) SE Runtime Environment (build 1.8.0_20-b26)
Java HotSpot(TM) Client VM (build 25.20-b23, mixed mode)
mambo@Mambo:~$ ant
Buildfile: build.xml does not exist!
Build failed
mambo@Mambo:~$ ant _version
Buildfile: build.xml does not exist!
Build failed
mambo@Mambo:~$
```

En este caso vemos que la versión mostrada es 1.8.0\_20

Link de descarga del zip: <https://github.com/xXMAMBOXx/Tarea-Programada-1.git>

**NOTA: Para efectos de compilación con ANT se debe de extraer el .zip y luego se debe de entrar en la carpeta del proyecto llamada "Progra1" para así llamar al programa ant desde la terminal y que compile y automáticamente cree la carpeta dist donde se encontrará el archivo Progra1.jar el cual puede ser ejecutado desde terminal o bien con doble click.**

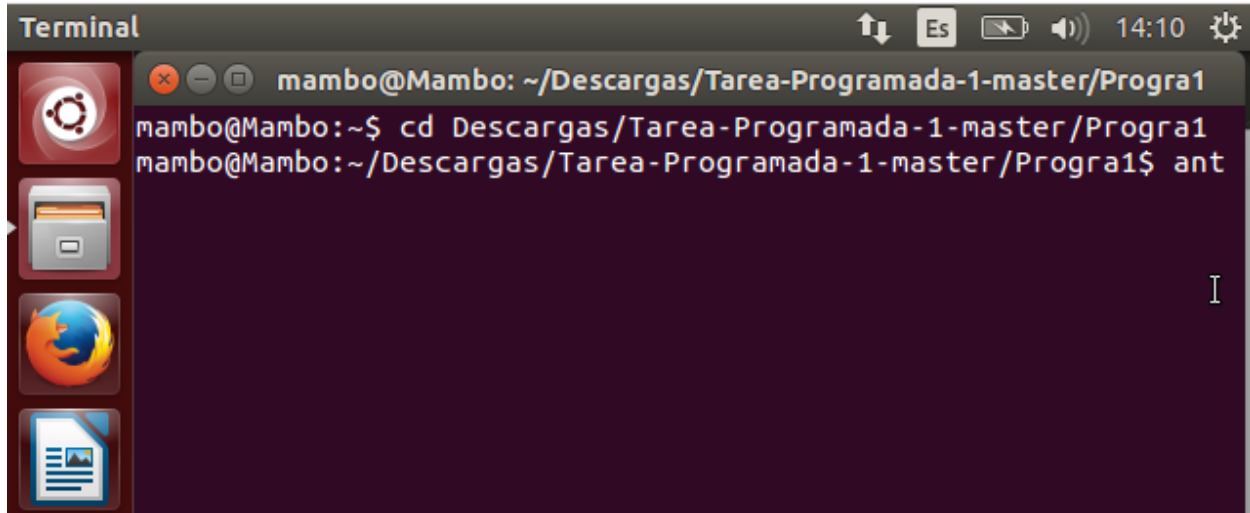
Ahora se debe de buscar la carpeta en la terminal de ubuntu o Linux (Progra1)

A screenshot of a Linux terminal window titled "Terminal". The window has a dark purple background. On the left side, there is a vertical dock with several application icons: a red circle with a white gear, a folder icon, the Firefox logo, a document icon, a spreadsheet icon, and a presentation icon. The terminal's title bar shows standard window controls (minimize, maximize, close) and system status icons (network, battery, volume, and a clock showing 14:10). The terminal text shows the user "mambo" at host "Mambo" in the directory "~/Descargas/Tarea-Programada-1-master/Progra1". The prompt "mambo@Mambo:~\$" is followed by the command "cd Descargas/Tarea-Programada-1-master/Progra1", and the next line shows the prompt "mambo@Mambo:~/Descargas/Tarea-Programada-1-master/Progra1\$".

```
Terminal
mambo@Mambo: ~/Descargas/Tarea-Programada-1-master/Progra1
mambo@Mambo:~$ cd Descargas/Tarea-Programada-1-master/Progra1
mambo@Mambo:~/Descargas/Tarea-Programada-1-master/Progra1$
```

Ya por estar dentro de la carpeta ejecutamos la acción de ANT

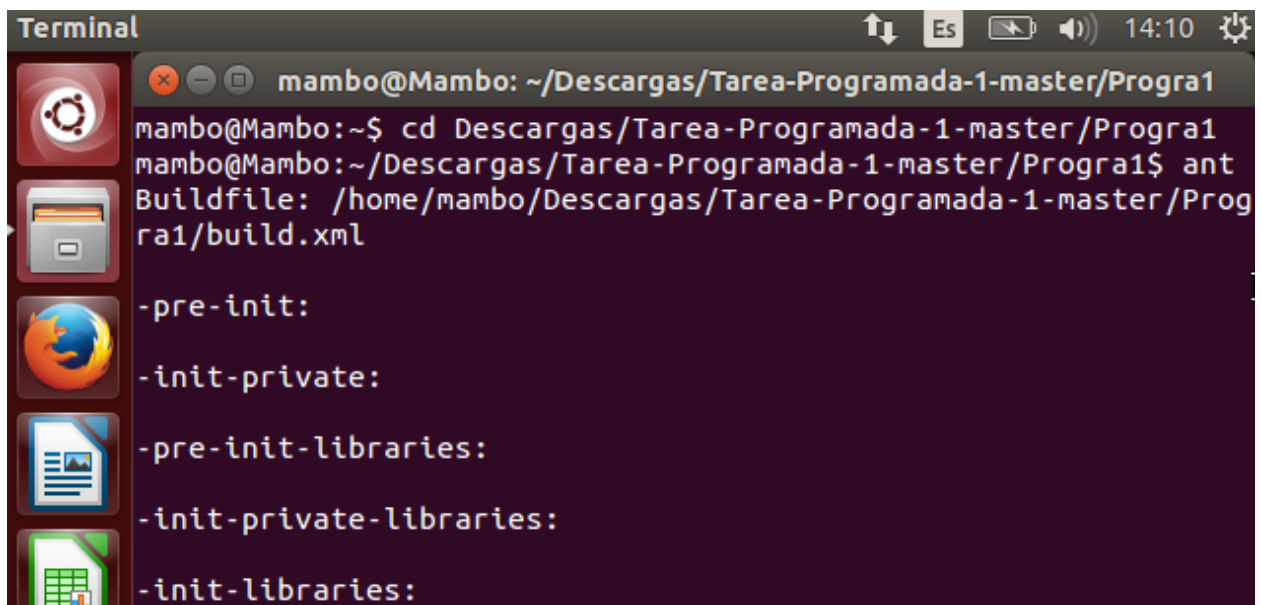




A terminal window titled "Terminal" with a dark background. The window shows the user "mambo" at host "Mambo" in the directory "~/Descargas/Tarea-Programada-1-master/Progra1". The user enters the command "cd Descargas/Tarea-Programada-1-master/Progra1" and then "ant". The terminal output shows the current directory path. On the left side of the terminal, there is a vertical dock with icons for a file manager, a web browser (Firefox), and a document editor.

```
Terminal
mambo@Mambo: ~/Descargas/Tarea-Programada-1-master/Progra1
mambo@Mambo:~$ cd Descargas/Tarea-Programada-1-master/Progra1
mambo@Mambo:~/Descargas/Tarea-Programada-1-master/Progra1$ ant
```

Esto encontrara automáticamente la dirección del build.xml la cual será la encargada de guiar todo el proceso de compilación

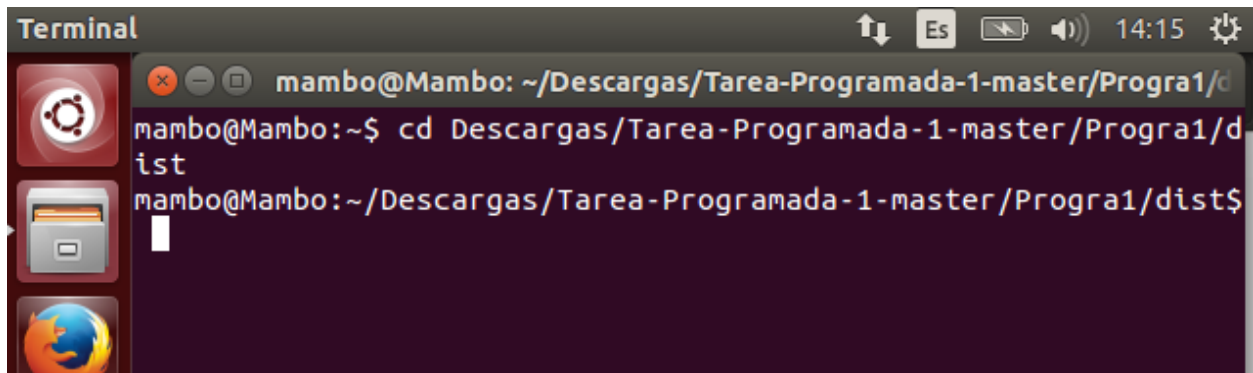


A terminal window titled "Terminal" with a dark background. The window shows the user "mambo" at host "Mambo" in the directory "~/Descargas/Tarea-Programada-1-master/Progra1". The user enters the command "ant". The terminal output shows the build file path and the start of the Ant build process, including the "-pre-init:" and "-init-private:" steps. On the left side of the terminal, there is a vertical dock with icons for a file manager, a web browser (Firefox), and a document editor.

```
Terminal
mambo@Mambo: ~/Descargas/Tarea-Programada-1-master/Progra1
mambo@Mambo:~$ cd Descargas/Tarea-Programada-1-master/Progra1
mambo@Mambo:~/Descargas/Tarea-Programada-1-master/Progra1$ ant
Buildfile: /home/mambo/Descargas/Tarea-Programada-1-master/Progra1/build.xml

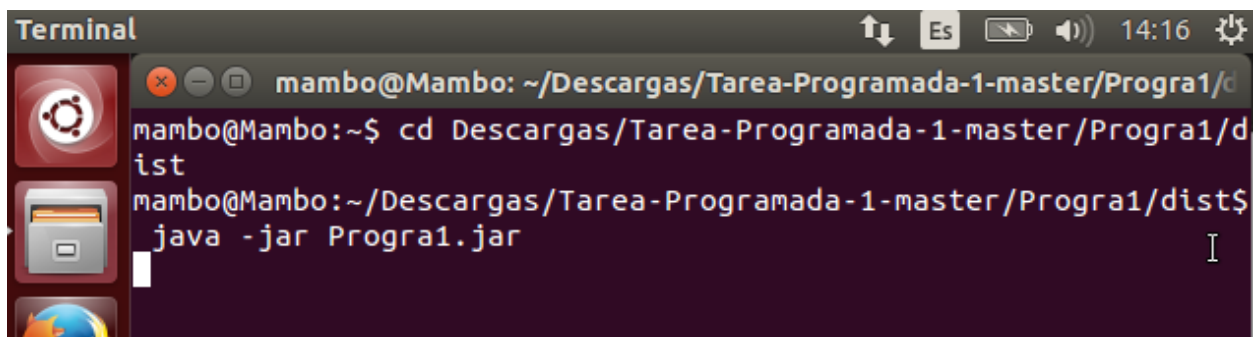
-pre-init:
-init-private:
-pre-init-libraries:
-init-private-libraries:
-init-libraries:
```

Aquí se ve cómo se va dando el proceso de compilación a través de la terminal de ant.

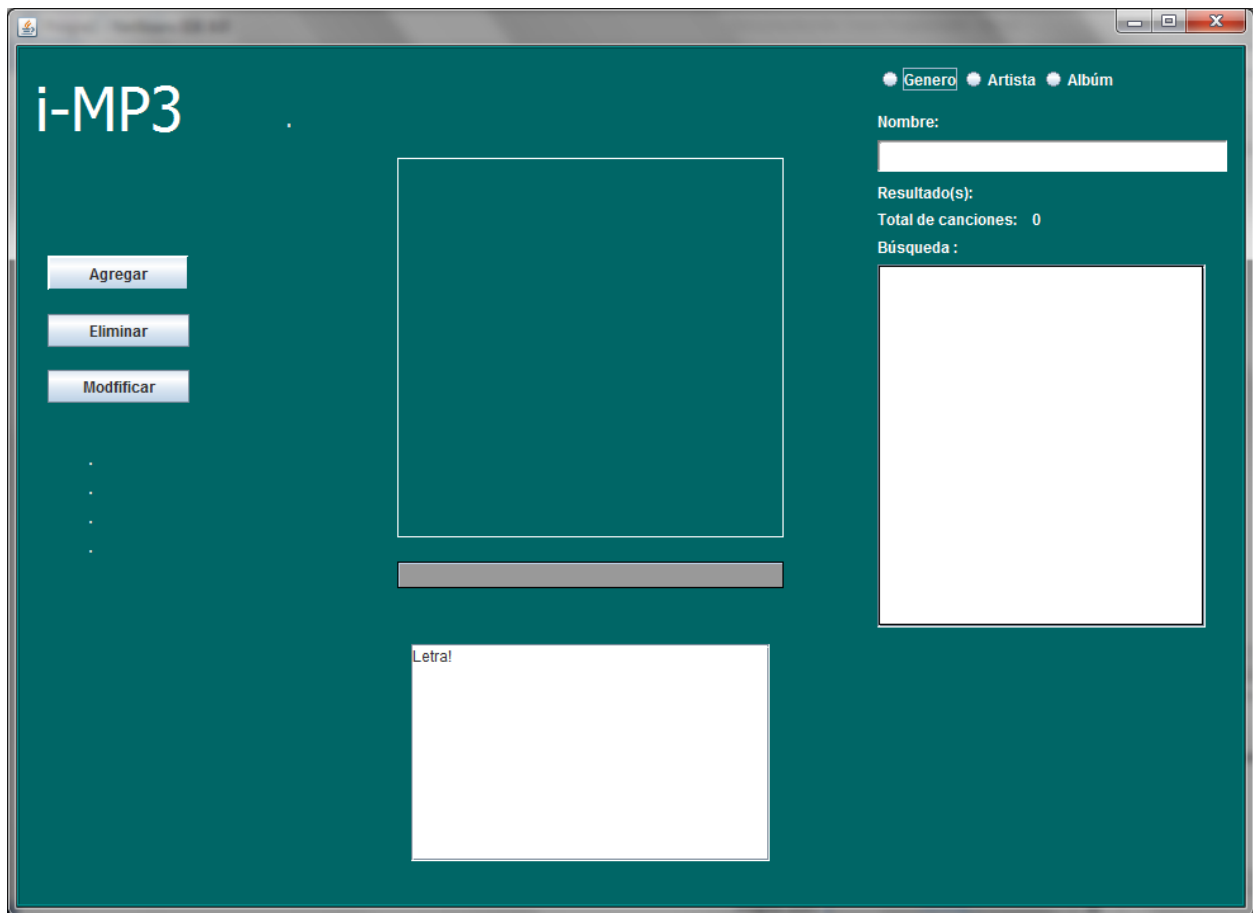
A terminal window titled "Terminal" with a dark background. The window shows a user named "mambo" at a machine named "Mambo". The current directory is "~/Descargas/Tarea-Programada-1-master/Progra1/d". The user enters the command "cd Descargas/Tarea-Programada-1-master/Progra1/dist" and presses enter. The prompt changes to "mambo@Mambo:~/Descargas/Tarea-Programada-1-master/Progra1/dist\$".

```
Terminal
mambo@Mambo: ~/Descargas/Tarea-Programada-1-master/Progra1/d
mambo@Mambo:~$ cd Descargas/Tarea-Programada-1-master/Progra1/dist
mambo@Mambo:~/Descargas/Tarea-Programada-1-master/Progra1/dist$
```

Y al finalizar este el siguiente comando adentro de la carpeta generada automáticamente por ant "dist" ,se mostrara la pantalla principal del programa l-mp3.

A terminal window titled "Terminal" with a dark background. The window shows a user named "mambo" at a machine named "Mambo". The current directory is "~/Descargas/Tarea-Programada-1-master/Progra1/d". The user enters the command "cd Descargas/Tarea-Programada-1-master/Progra1/dist" and presses enter. The prompt changes to "mambo@Mambo:~/Descargas/Tarea-Programada-1-master/Progra1/dist\$". The user then enters the command "java -jar Progra1.jar" and presses enter. The prompt changes to "mambo@Mambo:~/Descargas/Tarea-Programada-1-master/Progra1/dist\$".

```
Terminal
mambo@Mambo: ~/Descargas/Tarea-Programada-1-master/Progra1/d
mambo@Mambo:~$ cd Descargas/Tarea-Programada-1-master/Progra1/dist
mambo@Mambo:~/Descargas/Tarea-Programada-1-master/Progra1/dist$ java -jar Progra1.jar
mambo@Mambo:~/Descargas/Tarea-Programada-1-master/Progra1/dist$
```



Esta es la pantalla principal y desde aquí se hará todas las funciones del programa.

Componentes:

- Nombre del Programa

- Titulo. Esta ira cambiando de acuerdo a la canción que se esté reproduciendo.

Botones principales

- Agregar: Permite agregar canciones, se deben de agregar una por una.

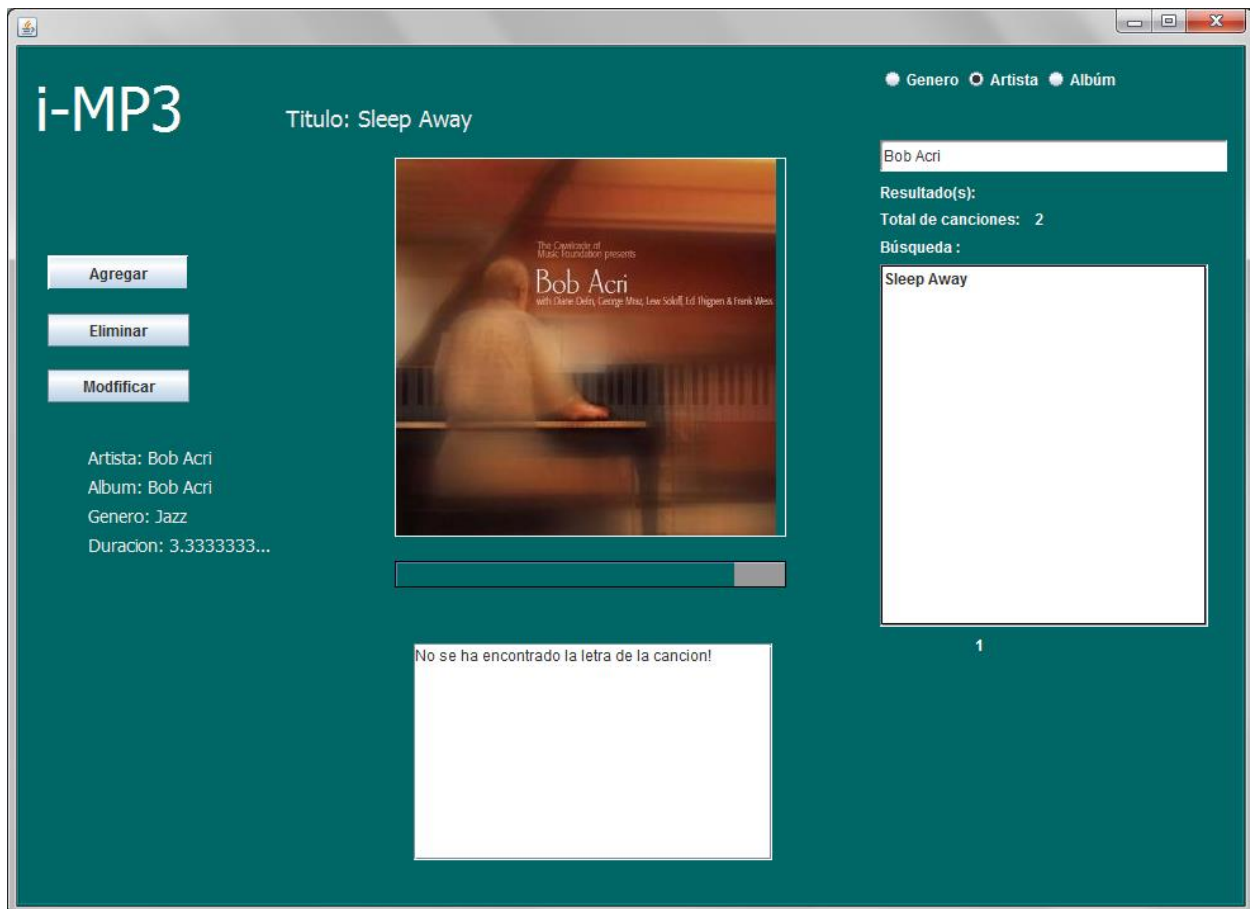
- Modificar: Abre la opción de modificar los datos de la canción como por ejemplo; título, género, álbum y Artista.

- Eliminar: Permite eliminar alguna canción del programa.

El programa cuenta con un recuadro en el centro de la pantalla aquí se mostrara la imagen que trae la canción por defecto. En tal caso que la canción no venga con su imagen de portada simplemente quedara en blanco.

El programa también incluye búsquedas avanzadas de canciones, el que viene por defecto es de búsqueda por nombre de canción pero usted puede seleccionar por género, álbum, o artista.

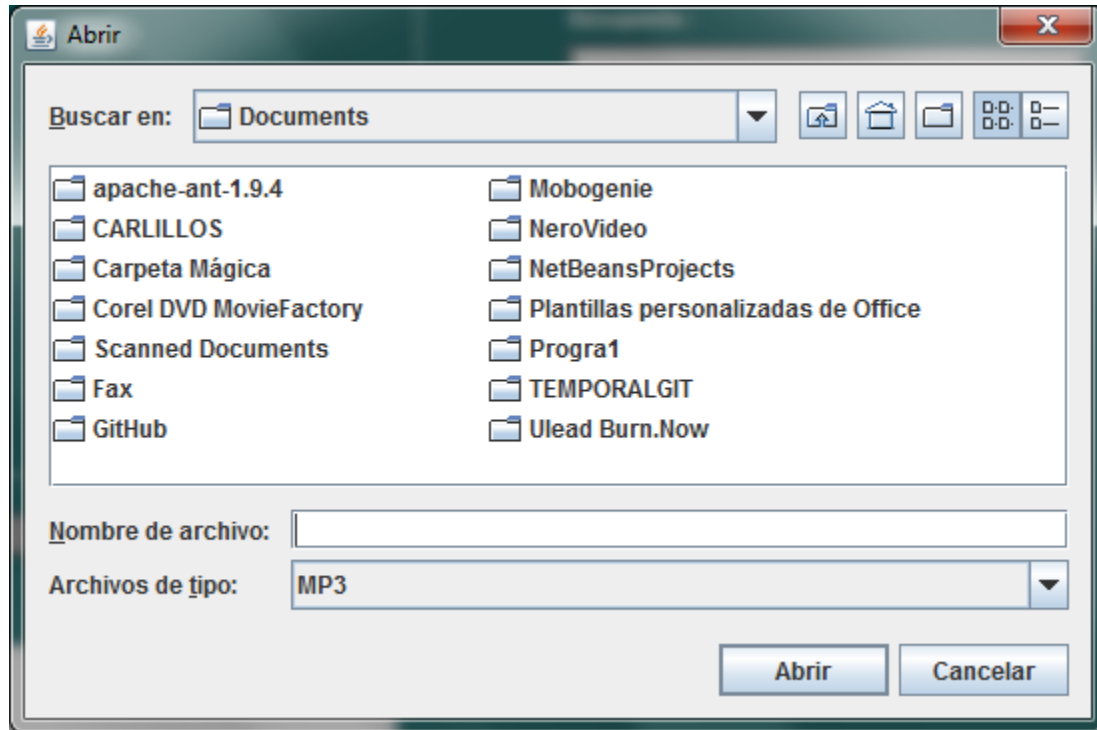
Los resultados de las búsquedas se mostraran en una lista de reproducción donde usted como usuario puede observar todas las canciones, y así también seleccionar la que guste.



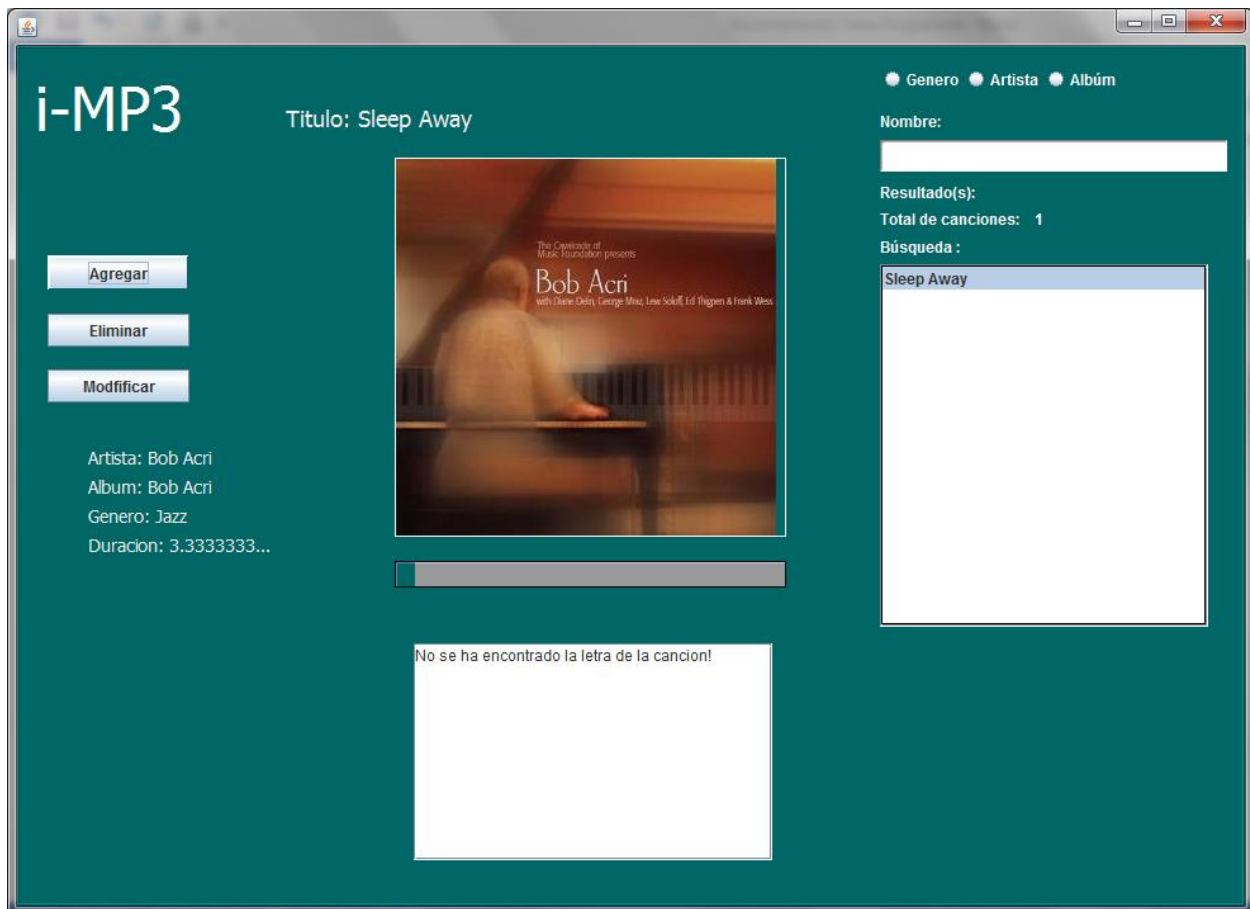
En este ejemplo se hizo una búsqueda por artista la cual genero una acción. Es necesario primero agregar el nombre en la pantallita y después seleccionar la búsqueda que se desea para su correcto funcionamiento.

Como agregar nuestra primera canción.

Le damos clic al botón de agregar



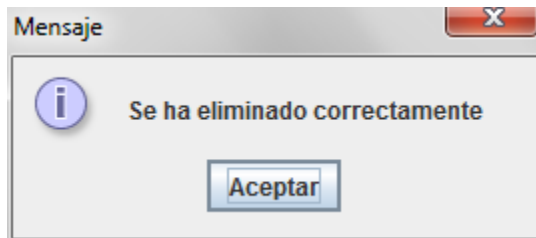
Se abrirá esta pantalla donde vamos a buscar la canción, seleccionarla y al final se le da clic en abrir.



Ya aquí estaría agregada la primera canción, y se observan cómo se llenaron los datos correspondientes con la información de la canción, y así podrá subir todas las canciones que guste.

En la parte de abajo se observa dos extras del programa, el primero es la barra que nos indica el avance de la canción en reproducción y la segunda es un recuadro de abajo el cual busca la letra en internet, y de ser encontrada nos mostrara la información.

Para la opción de eliminar seleccionar la canción que se desea eliminar, le damos clic al botón de eliminar y nos saldrá un mensaje como el de la siguiente imagen, esto nos indicara que la eliminación se realizó correctamente.



Para la opción de modificar, se le da al botón de modificar esto permite cambiar tres atributos.

-Titulo

-Álbum

-Artista

Se debe de tomar en cuenta que cada vez que se inicie el programa empezará desde cero, eso quiere decir que se eliminaran las canciones agregadas. En las futuras actualizaciones esto se mejorara

Para dudas o consultas no dude en comunicarse con nosotros por medio del correo electrónico

[l-mp3@gmail.com](mailto:l-mp3@gmail.com)

## **Conclusiones Personales:**

Marvin Fernández Coto:

Realizar la primera tarea programada manejando estructuras de datos simples fue un enriquecedor reto ya que difiere mucho del lenguaje acostumbrado (Python), sin embargo no fue tan complejo como esperaba, también se incrementó el conocimiento del lenguaje de programación (Java), el cual me gustó por la razón principal de que hay más fuentes de información que en python, es decir no había tanto problema en cuanto a encontrar información de librerías y documentación como si hay problemas en algunos casos con python.

Carlos Ramírez Cerdas

Como todo proyecto tiene sus aspectos positivos y negativos, aunque en este caso son mas los positivos. Gracias a experiencias anteriores para este proyecto nos organizamos bien y desde que se dieron las especificaciones, empezamos con las investigaciones respectivas. Lo primero fue una reunión entre los 3 integrantes para ver las opciones principales de formato, y llegar a un acuerdo entre todos. En cuanto a conocimientos se mejoro mucho el manejo de datos por medio de nodos, listas y arreglos y manejar mejor el lenguaje Java. Además se aprendió nuevas herramientas útiles para proyectos en grupos como Github. Ahora a corregir los pequeños errores para la segunda tarea programada y seguir aprendiendo algo nuevo todos los días.

Guillermo Ávila Chaves

En esta tarea programada aprendí mucho sobre estructuras de datos, manejo de archivos y la complejidad de algunas librerías a su vez, aprendí lo simple que puede llegar a ser java, ya que no existe la molestia de la indentación que existe en otros lenguajes (Python). Un buen punto del lenguaje fue la facilidad de encontrar ejemplos de código con los cuales se puede guiar y ayudarse para encontrar algoritmos los cuales puede ser más simples de los pensados, también aprendí el uso de herramientas de control de versiones como GitHub.

Ahora solo faltaría aprender a optimizar los algoritmos e investigar más sobre java y su gran potencial actual.



