

Instituto Tecnológico de Costa Rica

Tarea Programada # 1

Algoritmos y Estructuras de Datos– TI-2402



Segundo Semestre 2014

Fecha de entrega: 04/09/2014

Kenneth Martínez Avendaño (2014027338)

Silvia Elena Alpízar Rivera (2014128772)

Andrey Sánchez Aguilar (2014010852)

Tabla Contenido

Instituto Tecnológico de Costa Rica	1
Descripción del Problema	4
Diseño del Programa.....	5
.....	9
.....	10
Librerías Utilizadas	11
Basic Player	11
Mp3agic	15
Análisis de Resultados	18
Objetivos alcanzados	18
Objetivos no alcanzados	19
Manual de Usuario	21
Vista General	21
Menú	22
Listas de Reproducción	22
Lista de Canciones.....	24
Seleccionar Canción	24
Eliminar canción	25
Mostrar Datos	25
Reproducción de Música.....	26
Botón Play	26
Botón de Pausa	27

Regulador de Volumen	27
Botón Stop	28
Botón Siguiente	28
Botón Anterior	28
Búsqueda Avanzada	29
Ejecución de reproductor en el terminal de Linux	30
Conclusiones.....	31

Descripción del Problema

Este programa sirve como herramienta para facilitar el manejo y reproducción de la música contenida en el ordenador.

En él es posible encontrar las funcionalidades básicas de un reproductor de música: reproducir, pausar, reanudar, detener, pasar a la siguiente canción, pasar a la canción anterior, regular el volumen y observar el progreso del tiempo de la canción en tiempo real. Además de poder crear listas de reproducción donde se colocarán las canciones preferidas por el usuario.

El reproductor contiene una búsqueda avanzada en la que se podrán obtener las canciones desde diferentes puntos de acceso, la búsqueda puede realizarse por medio del álbum, el nombre de la canción, el título de la canción y el género

Al iniciarse el programa se podrán encontrar canciones que ya el usuario posee en su ordenador, estas canciones son extraídas de las carpetas Música, esta fue elegida puesto que es en el lugar donde la mayoría de los usuarios almacenan la música.

Diseño del Programa

Este programa fue creado en el lenguaje de programación Java utilizando como herramientas SublimeText2 y la plataforma de Netbeans IDE 8.0.

Se utilizó NetBeans IDE 8.0 debido a que facilita el desarrollo de la interface de usuario y la utilización de herramientas como ANT y Github.

Algunos de los archivos que incluían la parte lógica del proyecto fueron realizados en SublimeText2, tales como: los algoritmos de búsqueda, obtener los metadatos, persistencia de datos y estructuras de datos.

Esta decisión fue tomada ya que en un principio se había acordado realizar todo el proyecto en SublimeText2, pero a lo largo del desarrollo se fueron presentando situaciones que retrasaban el avance del proyecto, y que existían herramientas en las que esas situaciones carecían de problema alguno, la herramienta que se tomó en cuenta fue NetBeans.

En un principio se realizó la búsqueda de librerías para la reproducción de música, se decidió utilizar BasicPlayer, puesto que ofrecía todas las funcionalidades que se requerían tales como reproducir, pausar, reanudar y detener. Con esta librería se logró la reproducción de música.

Luego de lograr la reproducción de música mediante la dirección del archivo se procedió a elegir la estructura de datos para guardar esas direcciones, la estructura elegida fue la de listas doblemente enlazadas por medio de nodos, ya que estas facilitan el acceso a todas las canciones, pues contiene punteros hacia el siguiente nodo y el nodo anterior.

Luego de conocer la estructura a utilizar se ideó la forma de hacer que los datos fueran persistentes en la computadora, esto significa que cuando el programa se ejecuta crea una serie de archivos y carpetas en donde se van a guardar todos los

datos que el mismo genere. Como las direcciones de las canciones, estas direcciones son leídas por el reproductor para finalmente reproducirlas.

Ya teniendo un respaldo con la información se procedió a extraer los metadatos de las canciones los cuales también se guardarían en los registros generados por el reproductor por medio de la estructura de datos previamente implementada.

Al obtener lo metadatos se pudo crear un algoritmo de búsqueda el cual permite encontrar las canciones desde diferentes puntos de acceso como lo son el nombre del álbum, le nombre del artista, el género de la canción o le nombre de la canción.

El algoritmo de búsqueda es el siguiente:

```
public class AlgoritmosBusqueda{

    Datos cajaDatos;

    public AlgoritmosBusqueda(Datos cajaDatos){

        this.cajaDatos = cajaDatos;

    }

    public String buscarCancion(String nombre, int tipo){

        for(int contador = 0; contador < this.cajaDatos.getTamaño(); contador++){

            String variable = this.cajaDatos.obtenerDatos(contador)[tipo];

            if(variable == nombre){

                return this.cajaDatos.obtenerDatos(contador)[0];

            }

        }

        return null;

    }

}
```

Mediante SublimeText2 se creó la funcionalidad de leer los archivos de música de la carpeta Música, esta fue elegida por ser el lugar más común en la que los usuarios almacenan su música, la idea es básicamente que estas canciones leídas de esta carpeta aparezcan en una lista de canciones para que el usuario pueda disfrutar y disponer de ellas desde el inicio.

Algoritmo para acceder a la carpeta de Música y obtener las canciones:

- Método cargar música:

```
public void cargarmusica(){
    try{
        Archivos s = new Archivos();
        String p=s.obtenerDirectorioMusica();
        String[] canciones = s.leerArchivosCarpeta(p, null)[1];
        System.out.println("Prueba de cargar musica"+canciones.length);

        Archivos h = new Archivos();
        //System.out.println(d+canc);
        for(int i =0;i<canciones.length;i++){
            if(canciones[i]!=null)
                h.redireccionarFichero(canciones[i],null );}
        //System.out.println("Se ha agregado: "+canciones[i]);
        System.out.println("Se han agregado las canciones");
    }catch(Exception a){
        System.out.println("No se ha podido cargar la musica");
    }
}
```

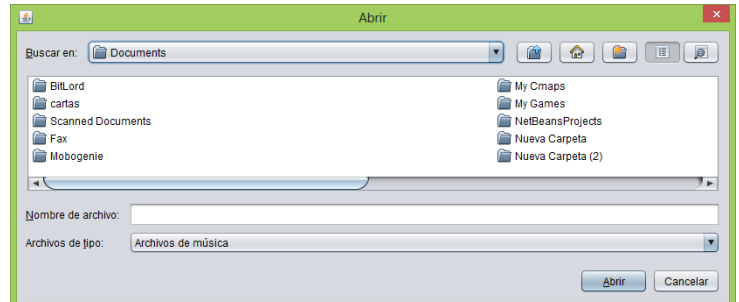
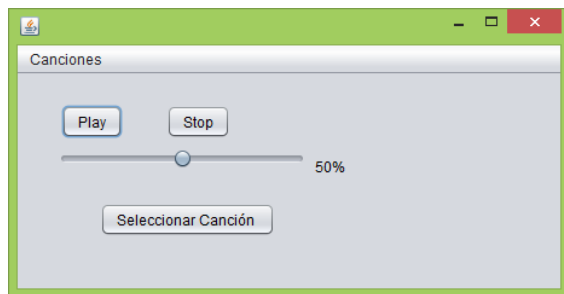
- Método leerArchivosCarpeta (necesario para cargar la música):

```
public String[] leerArchivosCarpeta(String directorio, String carpeta){
    File fichero = new File(directorio);
    File[] arregloFichero = fichero.listFiles(); // Lista de ficheros
    int cantidad = arregloFichero.length;
    String[] listaCanciones = new String[cantidad];

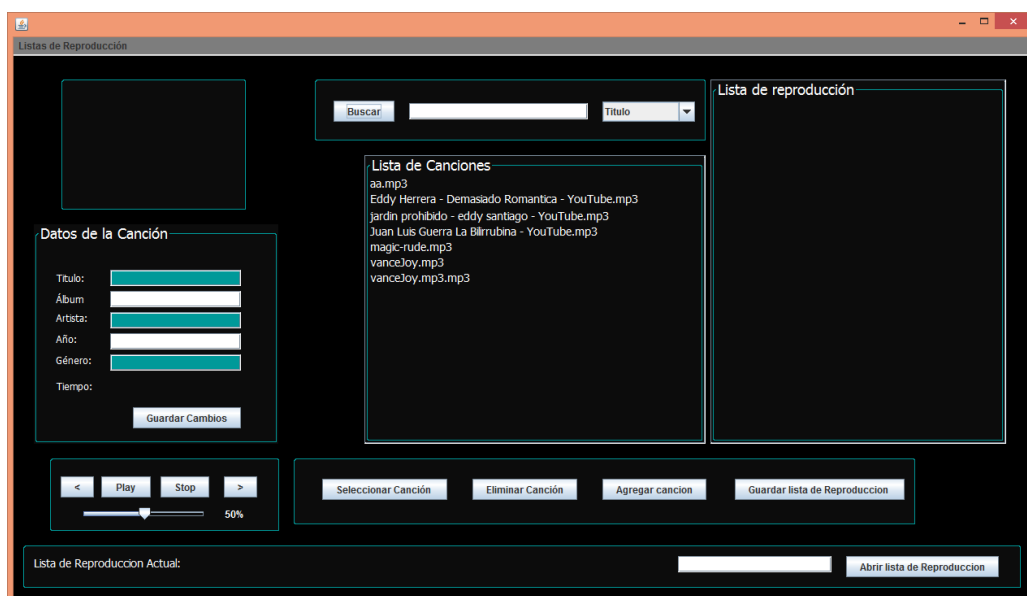
    for(int contador = 0; contador < arregloFichero.length;contador++){
        String cancion = arregloFichero[contador].getName(); // Guarda las direcciones

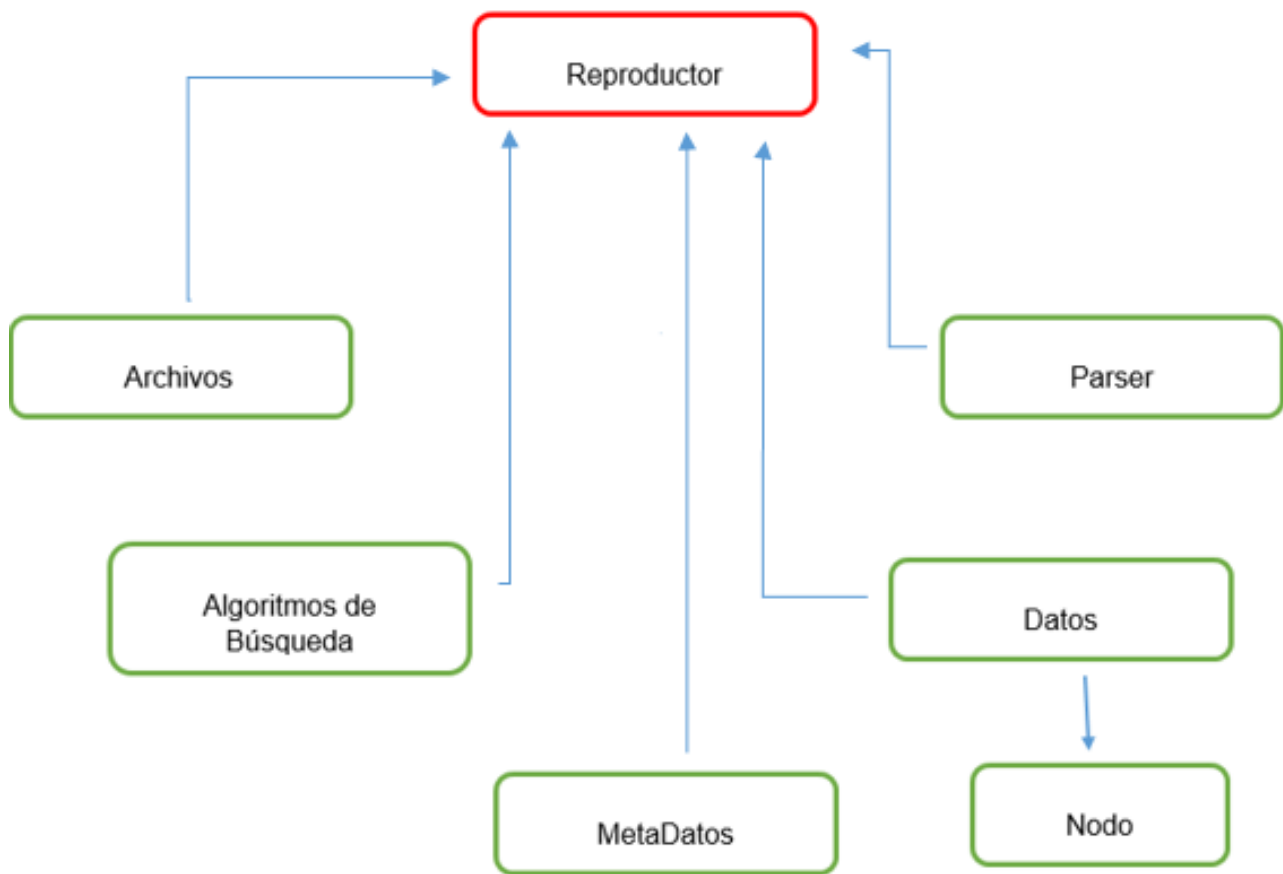
        if (cancion.endsWith(".mp3")||cancion.endsWith(".wav")||cancion.endsWith(".ogg")){
            // Busca solo archivos de musica
            listaCanciones[contador]=cancion;
            Archivos archivoTemp = new Archivos();
            archivoTemp.redireccionarFichero(cancion,carpeta);
        }
    }
    return listaCanciones;
}
```

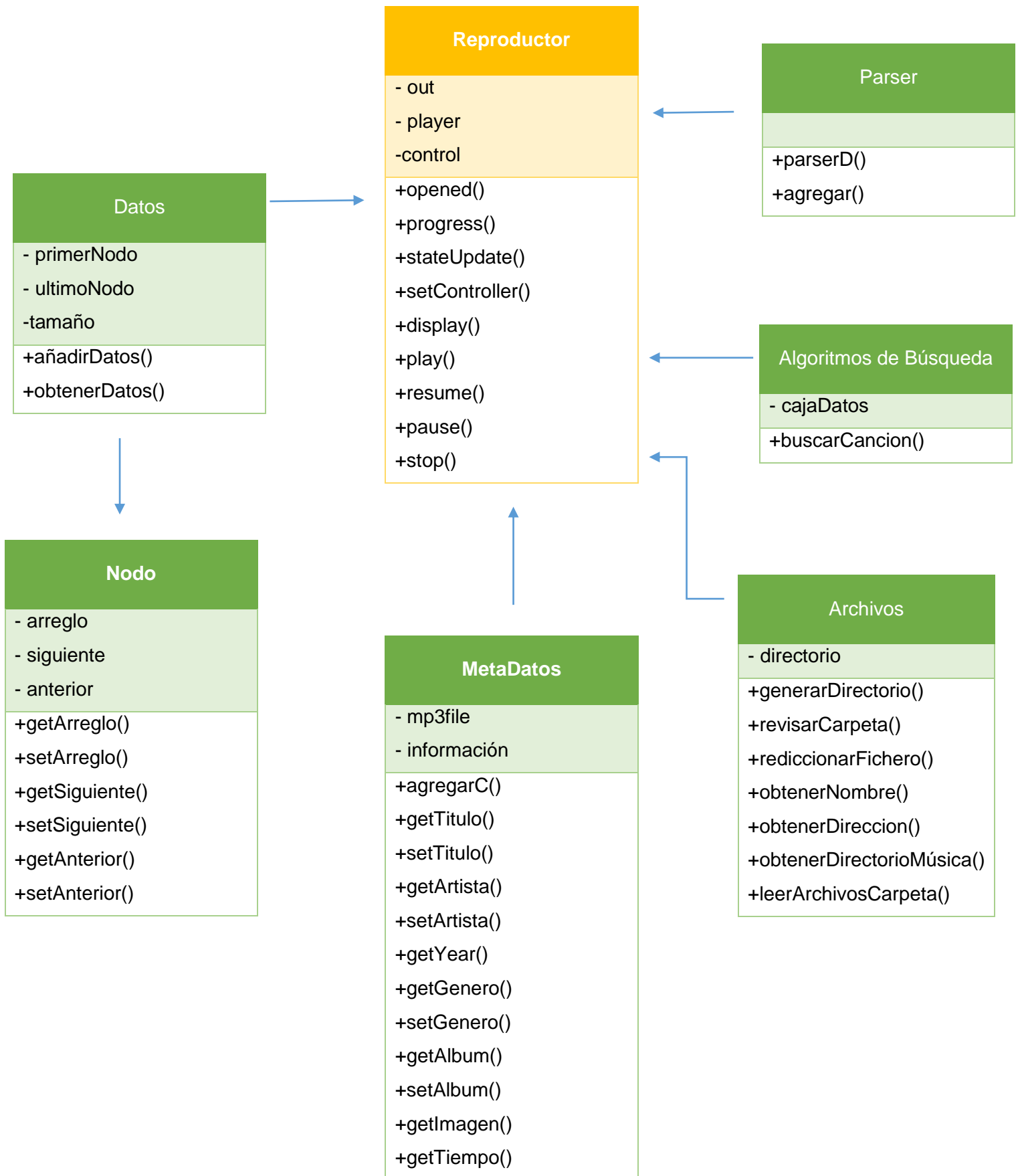
Se creó una pequeña interface que solo mostraba el administrador de archivos por medio de un botón y la dirección que se abría en el administrador era la que estaba lista para reproducir.



A partir de esa interface se fueron incorporando las nuevas funciones, mostrando en una lista las canciones extraídas de las carpetas Música y Descargas, se implementó la modificación de los metadatos, y la creación de las listas de reproducción. Las listas de reproducción son archivos que también serán guardados por el reproductor en estos están contenidos los nombres de las canciones, esos nombres serán cargados en una lista para que estén listos para la reproducción. La función de los botones anterior y siguiente y la búsqueda avanzada también fueron implementadas, mostrando una vista general de la siguiente manera:







Librerías Utilizadas

Basic Player

Para la reproducción de la música se decidió utilizar la librería de javazoom “BasicPlayer 3.0.jar”, esta librería proporciona un API de alto nivel en el que se contempla la música de diferentes formatos, (mp3, ogg y wav). Este archivo contiene dentro 8 archivos “.jar” que son estrictamente necesarios para que esta librería funcione de forma correcta. Estos son archivos son:

- Commons-logging-api.jar
- JI1.0.jar
- Jogg-0.0.7.jar
- Jorbis-0.0.15.jar
- Jspeex 0.9.7.jar
- Mp3spi1.9.4.jar
- Tritonus-share.jar
- Vorbisspi1.0.2.jar

Los métodos que esta posee son simples y sencillos de utilizar, gracias a que posee una clase controladora, podemos observar el proceso de ejecución de una canción, cuando abre el archivo, cuando se reproduce, si se pausó, si se detiene la canción, regular el volumen, etc.

Forma de uso:

La clase ReproductorM es la que contiene todo el control de ejecución de las canciones a través de basicplayer, es importante recalcar que durante la ejecución del programa toda la información estará impresa en consola y actualizándose hasta el momento en que la canción acaba o se selecciona una nueva canción. Esta clase se implementa de la siguiente manera:

1. Importar las librerías.

```
/*
 *Importar librerías necesarias para el control y la reproduccion de archivos
 */
import java.io.PrintStream;
import java.util.Map;
import javazoom.jlgui.basicplayer.*;
```

- java.io.PrintStream: implementa un flujo de salida, tiene la capacidad de imprimir representaciones de varios valores de datos convenientemente, en este caso mostrará los datos de la canción que se está reproduciendo.
- Java.util.Map: implementa un mapeo de claves a valores, es una clase define el orden en que las claves son traducidas dentro del mapa y devuelvan sus elementos, en este caso las propiedades de la canción.
- Javazoom.jlgui.basicplayer.*: accede a la clase basicplayer para la reproducción de la música y el control de esta.

2. Creación de la clase ReproductorM la cual implementará de la clase BasicPlayer, por lo que se deben sobrescribir los métodos.

```
public class ReproductorM implements BasicPlayerListener {

    private PrintStream out = null; //Stream que mostrará en consola toda la accion desarrollada por la reproducción del archivo
    BasicPlayer player = new BasicPlayer(); //Instancia de BasicPlayer
    BasicController control = (BasicController) player; //Controlador del reproductor

    /*
    *Constructor de la clase ReproductorM
    */
    public ReproductorM () {
        //Indica si en el proceso de reproducción ocurrieron acciones como pular, reproducir, reanudar, etc.
        player.addBasicPlayerListener(this);
        out = System.out;
    }
}
```

3. Crear los métodos sobrescritos de la clase BasicPlayer.

4. Método que proporciona información del archivo abierto, recibiendo de parámetros un objeto stream y propiedades tipo Map que accede a las propiedades de la canción convirtiéndola en información que se mostrará en la consola

```
public void opened(Object stream, Map properties) {  
    display("opened : " + properties.toString());  
}
```

5. Crear el controlador de la ejecución de la reproducción. Recibe como parámetros una variable int (bytesread) en la que se mostrará la cantidad de bytes leídos, el tiempo por medio de la variable tipo long (microseconds), un arreglo de bytes (pcmdata) que muestra la posición de los bytes,

```
public void progress(int bytesread, long microseconds, byte[] pcmdata, Map properties) {  
    display("progress : " + properties.toString());  
}
```

6. Método que imprime toda la información generada por los métodos anteriores y los imprime en la consola.

```
public void display(String msg) {  
    if (out != null) {  
        out.println(msg);  
    }  
}
```

7. Luego de crear la clase ReproductorM que se encarga del control de la ejecución se procede a hacer una nueva clase Música con los métodos play, stop, pausar, reanudar; los procedimientos básicos de un reproductor.

8. Crear la clase música e instanciar una variable tipo ReproductorM para controlar la reproducción de la música que se desea reproducir.

```
public class Musica{
    public ReproductorM miReproductor;// Instancia del ReproductorM para controlar la ejecución.

    public Musica() {
        miReproductor = new ReproductorM();
    }
}
```

9. Método que abre el fichero, este método recibe la dirección de la canción a reproducir.

```
public void AbrirFichero(String ruta) throws Exception {
    miReproductor.open(new File(ruta));
}
```

10. Métodos play, reanudar, pausa y stop.

```
public void Play() throws Exception{
    miReproductor.control.play(); //Reproduce el archivo
}

public void Reanudar() throws Exception{
    miReproductor.control.resume(); // Reanuda la canción
}

public void Pausa() throws Exception{
    miReproductor.control.pause(); // Pone pausa a la canción
}

public void Stop() throws Exception{
    miReproductor.control.stop(); //Detiene el proceso de reproducción
}
```

11. Crear un main para probar el reproductor en donde creamos un objeto de tipo Música al que se le agrega una dirección por medio del método “AbrirFichero” y finalmente reproduce la canción.

```
public static void main(String args[]) throws Exception{
    Musica cancion = new Musica();
    cancion.AbrirFichero("C:/Users/Public/Music/Sample Music/Kalimba.mp3");
    cancion.Play();
}
```

Mp3agic

Esta librería se utilizó para obtener los metadatos de la canción accedida, la información obtenida de cada canción es la siguiente:

- Artista
- Título
- Álbum
- Año
- Genero
- Fotografía del álbum
- Duración de la canción

Forma de uso:

1. Importar las librerías

```
import com.mpatric.mp3agic.ID3v1;
import com.mpatric.mp3agic.ID3v1Tag;
import com.mpatric.mp3agic.ID3v2;
import com.mpatric.mp3agic.ID3v24Tag;
import com.mpatric.mp3agic.InvalidDataException;
import com.mpatric.mp3agic.Mp3File;
import com.mpatric.mp3agic.NotSupportedException;
import com.mpatric.mp3agic.UnsupportedTagException;
```

2. Crear el constructor y declarar las variables tipo Mp3File (mp3file) y ID3vi (información), estas variables son necesarias para acceder a los datos de la canción.

```
public class MetaDatos1{
    static Mp3File mp3file;
    ID3v1 informacion;

    public MetaDatos1(){
    }
}
```

3. Crear método para obtener el nombre del artista de la canción accedida:

```
public String getArtista() {  
    System.out.println(informacion.getArtist());  
    return informacion.getArtist();  
}
```

4. Crear método para obtener el nombre del álbum:

```
public String getalbum() {  
    return informacion.getAlbum();  
}
```

5. Crear método para obtener año:

```
public String getYear() {  
    return informacion.getYear();  
}
```

6. Crear método para obtener el género:

```
public String getGenero() {  
    return informacion.getGenre();  
}
```

7. Crear método para obtener el título de la canción:

```
public String getTitulo() {  
    return informacion.getTitle();  
}
```


8. Crear método para obtener la duración de la canción:

```
public String getTime() {
    long num = mp3file.getLengthInSeconds();
    long hor,min,seg;
    String tiempo;
    hor = num/3600;
    min = (num-(3600*hor))/60;
    seg=num-((hor*3600)+(min*60));
    tiempo = hor+"h:"+min+"m:"+seg+"s";
    return tiempo;
}
```

9. Crear método para obtener la imagen del álbum:

```
public void getImage() throws UnsupportedTagException, InvalidDataException, IOException, NotSupportedException{
    if (mp3file.hasId3v2Tag()) {
        ID3v2 imagen = mp3file.getId3v2Tag();
        byte[] imageData = imagen.getAlbumImage();
        if (imageData != null) {
            String mimeType = imagen.getAlbumImageMimeType();
            System.out.println("Mime type: " + mimeType);
            try ( // Write image to file - can determine appropriate file extension from the mime type
                RandomAccessFile file = new RandomAccessFile("cover.jpg", "rw")) {
                file.write(imageData); //
            } //
            mp3file.save("MyMp3File.mp3");
        }
    }
    else
        System.out.println("No tiene imagen");
}
```

Análisis de Resultados

Objetivos alcanzados

- **Reproducción de la música:**

Este objetivo se cumplió gracias a la implementación de la librería “BasicPlayer”, esta contiene funcionalidades que permitieron la reproducción de la música en diferentes formatos: mp3, wav, ogg. Gracias a esta fue posible también regular el volumen de la canción.

- **Metadatos:**

Se colocaron los metadatos en espacios de texto donde se puede modificar al presionar el botón guardar cambios.

- **Persistencia de datos:**

Para la persistencia de datos se utilizó la librería de java llamada: “java.io.*”, el objetivo de esto era lograr que el reproductor creara una carpeta propia en la que pudiera administrar los datos generados por el mismo; las canciones agregadas, listas de reproducción, direcciones, entre otros.

- **Obtener las canciones contenidas en las carpetas Música del computador:**

La implementación de obtener las canciones que el usuario posee en su carpeta de Música surgió por la necesidad de darle una apariencia más real de un reproductor al proyecto, ya que abrir el programa y encontrar canciones directamente desde las carpetas del usuario resta tiempo y es más sencillo para este comenzar a disfrutar de su música, además se tomó en cuenta que

puede ser tedioso agregar canción por canción al inicio del programa. Si bien existe la funcionalidad de agregar canciones al reproductor estas serían canciones que se encuentran en otras carpetas diferentes distribuidas dentro del computador del usuario.

- **Crear listas de reproducción:**

La creación de las listas de reproducción se implementó para manejar y controlar las canciones elegidas por el usuario en una lista específica. El usuario puede crear las listas de reproducción que desee.

- **Búsqueda avanzada:**

Esta funcionalidad facilita el acceso a las canciones a las que el usuario desea acceder por medio del título, el álbum, el género y el artista. Al usuario se le presentan las canciones relacionadas con la descripción de la búsqueda.

Objetivos no alcanzados

- **Leer archivos de música de todas las carpetas de la computadora.**

Después de analizar varios puntos como el tiempo que iba a requerir esta funcionalidad en revisar todas las carpetas buscando archivos de música, se tomó la decisión de aplicar esta funcionalidad a la carpeta en la que regularmente los usuarios almacenan su música: Música

Aunque el objetivo de leer todas las canciones contenidas en el computador no fue posible, lo satisface la lectura de la carpeta Música, ya que el objetivo principal consistía en mostrar canciones al iniciar el programa del reproductor, canciones que pertenecieran al usuario.

- **Progress Bar**

Aunque se implementaron las funciones básicas de un reproductor de música, por falta de tiempo no se logró implementar la barra de progreso de la canción.

- **SublimeText**

Desde un principio se llegó a una convención de realizar el proyecto solo por medio de esta herramienta, pero conforme fue pasando el tiempo se fueron presentando situaciones que causaban retrasos. La interface era uno de esos aspectos, más aun en lo que es el uso de las librerías externas, por lo que se dividió en dos secciones, la parte lógica se creó en Sublime y la interface se creó en NetBeans.

Manual de Usuario

Este manual servirá como herramienta para aprender a utilizar todas las funcionalidades del Reproductor de Música. En su contenido podremos observar los aspectos más esenciales para poder utilizar el reproductor de una manera más simple.

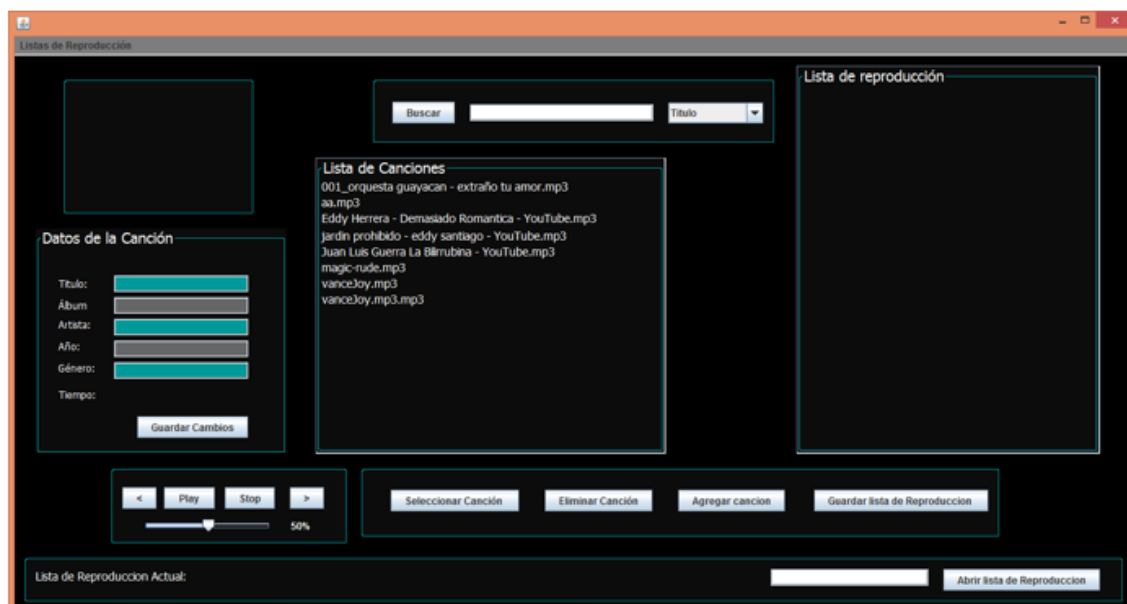
A continuación se presentan las instrucciones de uso:

Vista General

En la pantalla se mostrará la visión general del reproductor con los siguientes elementos:

- Un menú con la opción Lista de Reproducción.
- Una lista de canciones.
- Una lista de reproducción.
- La información de la canción.
- Botón “Seleccionar Canción”.
- Botón “Eliminar Canción”.
- Botón “Guardar cambios”.
- Botón “Guardar lista de reproducción ”
- Botón “Play”
- Botón “Stop”.
- Botón siguiente “>”
- Botón anterior “<”.
- Botón “Abrir lista de reproducción ”
- Botón “Buscar”
- Combo box para elegir el tipo de búsqueda.
- Una barra que regula el volumen de la canción.

Vista general

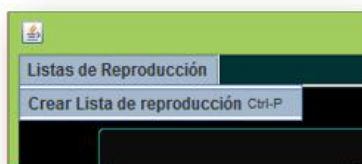


Menú

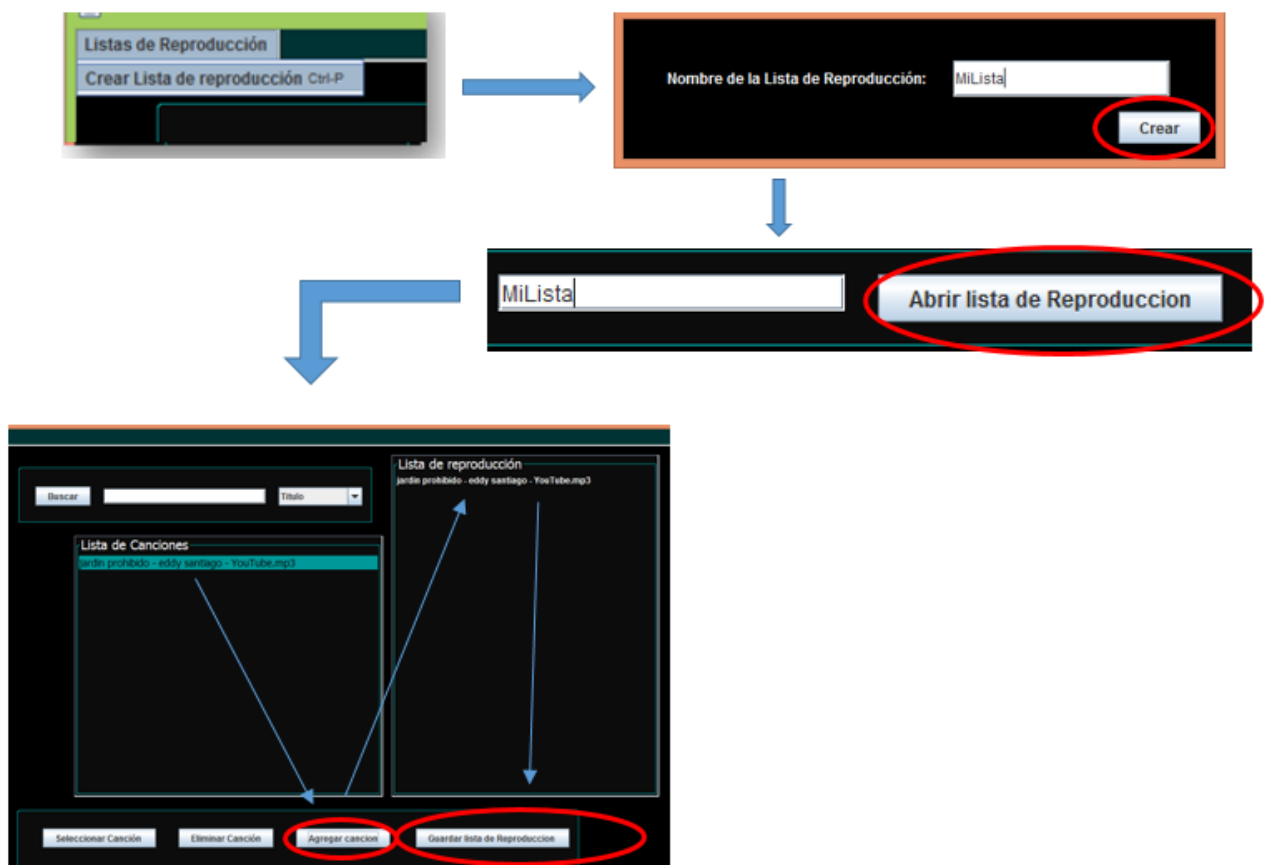
Listas de Reproducción

Al presionar la opción “Lista de Reproducción” se desplegará una nueva opción que indica “Crear lista de reproducción”.

La expresión “Ctrl+C” se refiere a una combinación de botones que usted presiona en su teclado para realizar esta acción más rápidamente sin necesidad de dirigirse al menú y presionar la opción “Crear Lista de Reproducción”. Presione la tecla “Control” y la tecla “C” simultáneamente.



Al presionarlo se abrirá una pequeña pantalla en la que se le solicitará el nombre de la lista de reproducción luego se debe presionar el botón “Crear”, este crea un nuevo archivo con el nombre de la lista de reproducción en la carpeta “FileMusic” creada por el reproductor, en ese momento la lista de reproductor está vacía, para agregarle nuevas canciones esta lista debe cargarse por medio del botón “Abrir lista de Reproducción”, coloque el nombre de la lista y presione el botón, después de cargada la lista basta con agregar las canciones preferidas por medio del botón “Agregar canción” y seleccionando de la lista de canciones, cuando ya haya finalizado de agregar las canciones presione el botón “Guardar lista de Reproducción”.

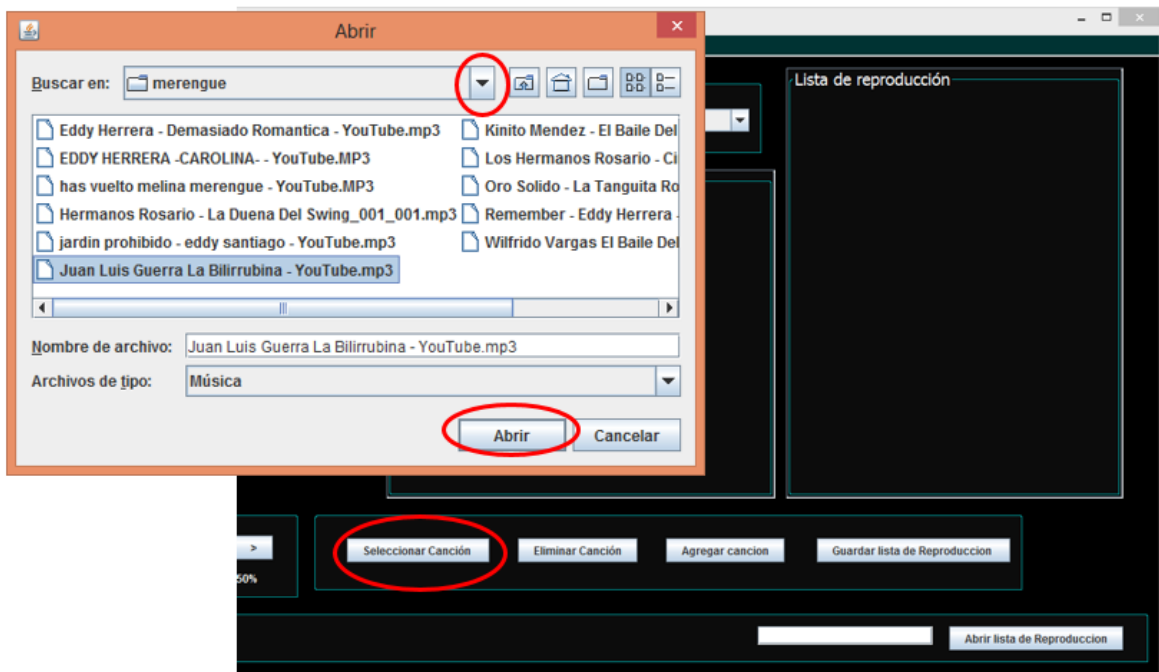


Lista de Canciones

Esta lista estará compuesta por las canciones que se encuentran en la carpeta “Música” del computador y las canciones nuevas que se irán acumulando en la carpeta “FileMusic” que crea el reproductor para guardar las canciones.

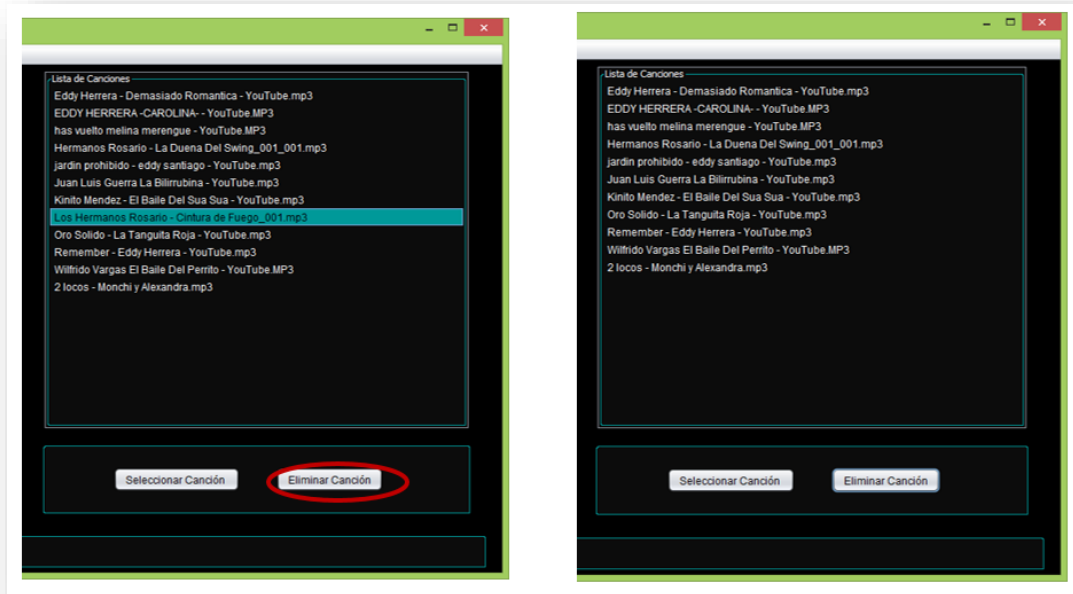
Seleccionar Canción

El botón “Seleccionar Canción” servirá para agregar nuevas pistas a la lista de canciones. Al presionarlo se abrirá el administrador de archivos en el que usted podrá seleccionar cualquier canción que desee y desde cualquier carpeta. Esta funcionalidad contiene un filtro en el que solo recibirá formatos mp3, ogg y wav. En pestaña de “Buscar en” podrá navegar entre las carpetas de su ordenador para encontrar su música preferida. Cuando ya haya elegido la canción presione el botón “Abrir” y la canción automáticamente se agregará a su lista de canciones.



Eliminar canción

El botón “Eliminar Canción” eliminará la canción seleccionada de la Lista de Canciones. Se podrá observar que la canción se ha eliminado de la lista.



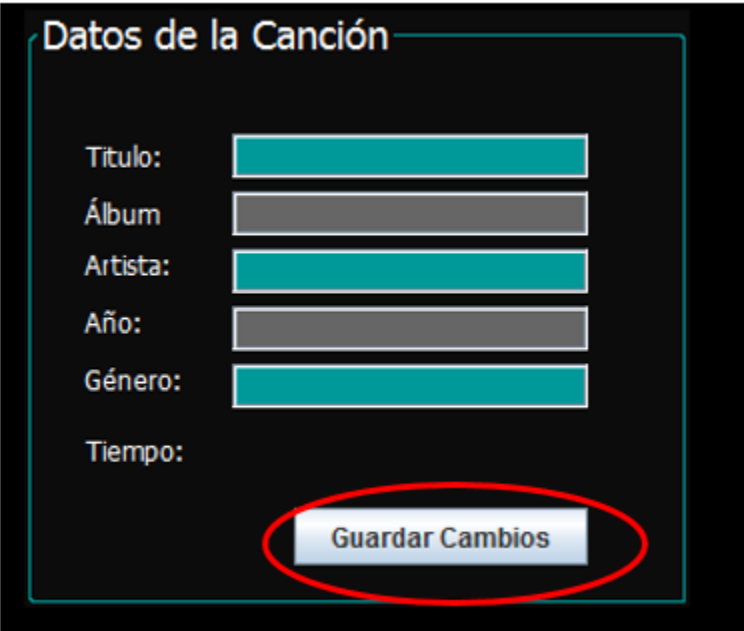
Mostrar Datos

Los datos que se mostrarán para todas las canciones son:

- Artista
- Título
- Álbum
- Año
- Género
- Fotografía del álbum
- Duración de la canción

Estos datos se mostrarán por medio de cuadros de texto a los cuales usted tendrá acceso para cambiar la información de la canción según lo desee. Mediante el botón

“Guardar cambios”, se harán validos los cambios realizados a la información de la canción y serán visibles dentro de la carpeta donde se almacena la canción.



Formulario de Datos de la Canción:

- Título:
- Álbum:
- Artista:
- Año:
- Género:
- Tiempo:

Botón: **Guardar Cambios**

Reproducción de Música

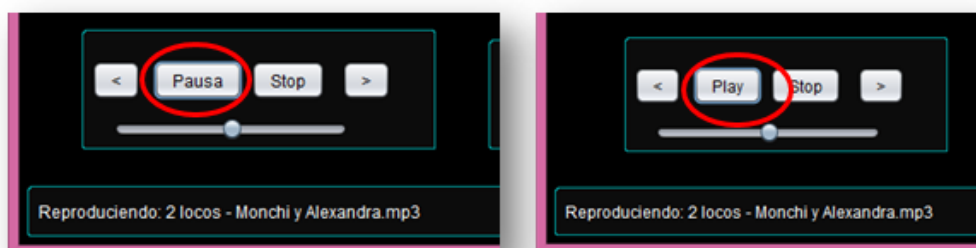
En este apartado explicaremos la funcionalidad más esencial del Reproductor, cada acción de los botones involucrados será presentada paso a paso. Es importante recalcar que el botón “Play” cambiará de estado a “Pausa” cada vez que sea presionado.

Botón Play

Para reproducir la música se debe seleccionar una canción de la lista de canciones o de la lista de reproducción y presionar el botón “Play”. En el recuadro inferior se podrá observar el nombre de la canción que está siendo reproducida. Automáticamente luego de que usted presiona el botón “Play” a la canción que previamente seleccionó de la lista de canciones, el estado del botón cambia a “Pausa”.

Botón de Pausa

Para pausar la canción presione el botón que en este momento indica “Pausa”, para detener la reproducción de la canción. De la misma manera cuando usted desee reanudar la canción debe presionar el botón “Play”, reproduciendo la canción desde el punto donde había sido detenida, el botón cambiará este al estado de “Pausa” nuevamente.



Regulador de Volumen

Con la barra que se encuentra bajo los botones de reproducción usted puede regular el volumen de la canción, este siempre iniciará con un 50% de volumen siendo 100% su máximo alcance. Para regularlo solo arrastre el indicador hacia la izquierda para reducir el volumen y a la derecha para aumentarlo.



Botón Stop

Al presionar este botón la reproducción de la canción se terminara por completo hasta que usted seleccione otra canción para reproducir. El volumen de la canción quedará donde usted lo indicó en la última reproducción.



Botón Siguiente

Al presionar este botón el reproductor se dirigirá a la canción que se encuentra a continuación dentro de la lista de canciones.



Botón Anterior

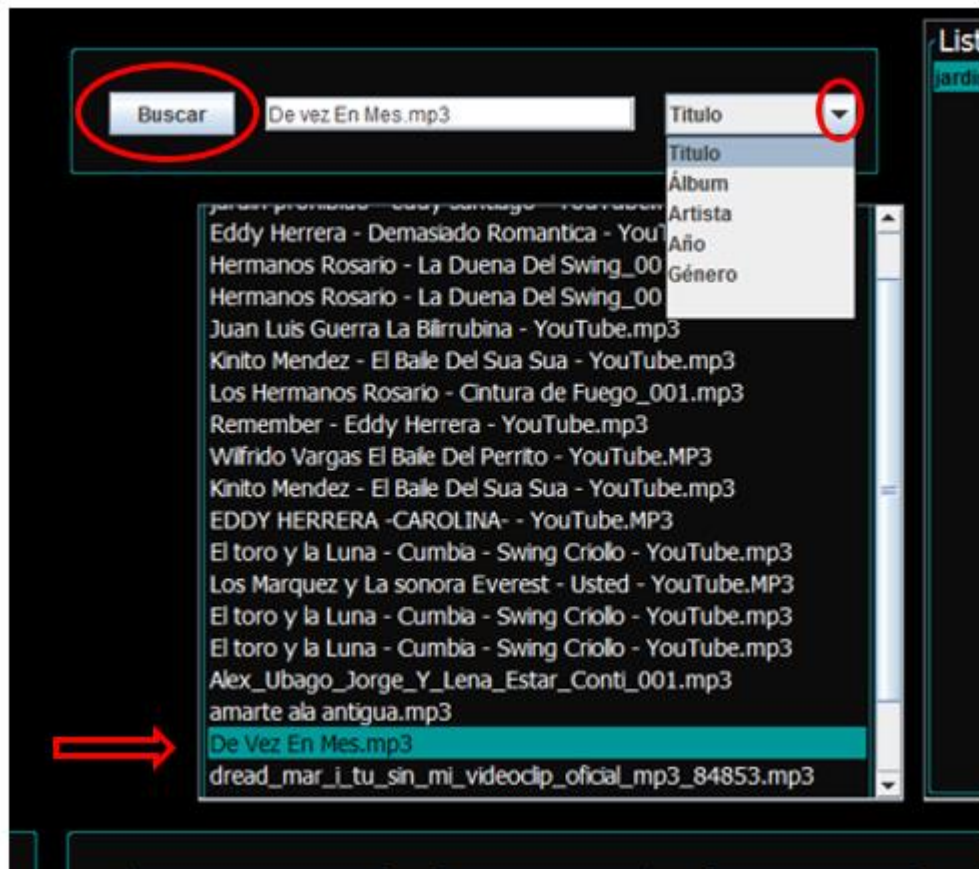
Al presionar este botón se reproducirá la pista que se encuentra antes de la canción que se está reproduciendo en ese momento.



Búsqueda Avanzada

Para buscar una canción por medio del título, el álbum, el género y el artista solo se debe colocar una expresión específica en el espacio de texto, elegir la categoría en el menú de opciones y presionar el botón buscar.

Si el buscador la encuentra, la seleccionará en la lista de canciones, y no sucederá nada si este no la encuentra.



Ejecución de reproductor en el terminal de Linux

Para poder ejecutar este programa desde la plataforma de Linux se deben seguir los siguientes pasos:

1. Dirigirse a línea de comandos (terminal) y buscar la respectiva carpeta donde se encuentra el programa.
2. En la línea de comandos y dentro de la carpeta colocar la palabra "ant".
3. Entrar a la carpeta "dist".
4. Escribir `java -jar main`.

Conclusiones

Al finalizar este proyecto, logramos comprender de manera más profunda la conceptualización de las estructuras de datos por medio de listas enlazadas.

La utilización del repositorio de Github nos deja una gran experiencia, de cómo trabajar de forma más ordenada y llevando un control continuo de las actualizaciones y cambios que se realizan a todos los archivos.

La herramienta NetBeans utilizada para la implementación de la interface nos proporcionó nuevos conocimientos sobre el lenguaje de programación y de su largo alcance.

Aparte de NetBeans se utilizó el editor de texto SublimeText2, que si bien es un poco más complicado, también sumo su cuota a nuestro aprendizaje sobre java. En muchas ocasiones surgieron problemas por falta de librerías y recursos desconocidos que no podíamos implementar directamente desde SublimeText pero gracias a que utilizamos NetBeans y este provee un fácil acceso a estas, se hizo posible.