



## Tarea Programada I

Reproductor de música para Linux

Profesor: Andrei Fuentes Leiva

Curso: Algoritmos y Estructuras de Datos

Estudiantes:

Alejandro Gonzalo Conejo, carné 201208110

Andrés Madrigal Vargas, carné 2013053178

Keiner Morales Goñi, carné 2013097325

Semestre II, 2014

## Tabla de contenidos

Descripción del problema.....	3
Diseño del Programa.....	4
Decisiones de diseño.....	4
Algoritmos usados.....	4
Diagramas Lógicos.....	6
Librerías usadas.....	8
Análisis de resultados.....	8
Manual de Usuario.....	9
Buscar la canción y seleccionarla:.....	10
Reproducir y utilizar:.....	10

## Descripción del problema.

Las estructuras de datos aplicadas en la programación nos permiten un mejor ordenamiento de la información que se quiera manejar para determinado software. Dicho esto, es muy importante conocer los tipos de estructuras de datos, sus pros y contras, y con esto desarrollar la capacidad de discernir y elegir la mejor alternativa de manejo de datos, buscando siempre que sea mediante una estructura que optimice el funcionamiento de la aplicación y de la cuál se puedan aprovechar al máximo sus características.

La primer tarea programada nos plantea el reto de diseñar un reproductor de música que soporte el formato .mp3, que posea una interfaz gráfica de usuario y que sea diseñada para correr en Linux. Además, este mismo debe ser programado en el lenguaje Java, los archivos fuente se deben entregar mediante la herramienta de Apache ANT y se debe utilizar Github. Este conjunto de aspectos técnicos ya asumen varios retos que son nuevos para el equipo de trabajo, ya que nunca hemos trabajado con la herramienta de Apache, con Github y con la generación de aplicaciones para Linux, es por esta razón que se le deba poner especial atención a estos aspectos.

Dentro de las funciones propias de la aplicación se destacan las funcionalidades básicas de cualquier reproductor de música, tales como reproducir, pausar, adelantar a la siguiente canción y devolverse a la canción anterior. Se debe tomar en cuenta también el manejo de la metainformación de la canción, la implementación de una búsqueda avanzada y la posibilidad de agregar y eliminar canciones. Dentro de lo más importante en esta parte del reto esta la selección del tipo de estructuras de datos más eficiente, y el manejo de archivos .mp3, ya que es algo que desconocemos.

Una vez planteados los principales retos y establecidos los puntos más importantes de la tarea, se puede proceder a realizar todo lo necesario para alcanzar la totalidad de los objetivos planteados. El trabajo en equipo y la prioridad que le daremos al trabajo serán las principales herramientas que vamos a utilizar para lograrlo.

# Diseño del Programa

## Decisiones de diseño

El equipo de trabajo ha dedicado un tiempo específico en la planeación del diseño tanto interno (código) como externo (interfaz) ya que se tiene una conciencia de que no sólo el producto final que se le entrega a los usuarios importa para el diseño, sino que los principios del programador apuntan a que el orden y la estética en el código son una buena práctica.

Con respecto al código y a la decisión de las diferentes estructuras de datos implementadas fueron las siguientes: La metainformación de cada canción se almacenó en una lista simple, cada nodo de la lista contiene datos diferentes (Título, Artista, Género, Album, Número de track, Dirección de la canción), así se asegura que la información de cada canción se almacena en un solo nodo de la lista simple. Dicha lista, es un elemento de los nodos de una lista doble circular, que almacena las canciones, esta estructura de datos facilita la navegación por las diferentes canciones a la hora de implementar las funciones de previous y next, ya que no se encontraron en ninguna librería de manejo de archivos mp3.

La interfaz gráfica no será de pantalla completa, sus dimensiones serán pequeñas y su diseño será natural, semejando las interfaces a las que los usuarios están acostumbrados. Dentro de los elementos principales se encuentra un área de navegación, donde se muestran las canciones y las categorías, un botón de play/pause, un botón de next, un control de volumen, un área de display de la metainformación de la canción en ejecución y una barra de búsqueda.

## Algoritmos usados

A continuación, se mostrarán segmentos de código implementados en la tarea programada, y se explicará brevemente su funcionamiento.

```

public void next() throws Exception {
    stop();
    actual = getActual();
    counter = 0;
    try {
        while (Integer.parseInt(list.obtenerElementoEnPosicion(counter).obtenerDato5().toString()) != actual + 1){
            counter++;
        }
        setSong (list.obtenerElementoEnPosicion(counter).obtenerDato1().toString() + ".mp3");
        playMP3();
    }
    catch (java.lang.NullPointerException e){
    }
}

```

*Algoritmo 1: Básicamente, se implementa mediante los recorridos de la lista doble circular, apuntando hacia al siguiente nodo en la reproducción*

```

public static ListaDoble filtrarElementos (ListaDoble list){
    int actual = 0;
    ListaDoble tracks = new ListaDoble();
    ListaDoble folders = new ListaDoble();
    for (int i = 0; i < list.LargoLista(); i++){
        //System.out.println("1");
        if (getExtensionFile(list.obtenerElementoEnPosicion0(actual).toString()).equals("mp3")){
            //System.out.println("1");
            tracks.InsFinal0(list.obtenerElementoEnPosicion0(actual));
        }
        else{
            //System.out.println("0");
            folders.InsFinal0(list.obtenerElementoEnPosicion0(actual));
        }
        actual++;
    }
    return tracks;
}

```

*Algoritmo 2: Este algoritmo se encarga de hacer la filtración de las canciones por su formato*

```

private class Nodo
{
    private Object dato1;
    private Object dato2;
    private Object dato3;
    private Object dato4;
    private Object dato5;
    private Object dato6;
    private Nodo siguiente;
}

```

*Algoritmo 3: Es importante destacar los atributos del nodo de la lista simple, ya que este es el que guarda la metainformación*

## Diagramas Lógicos

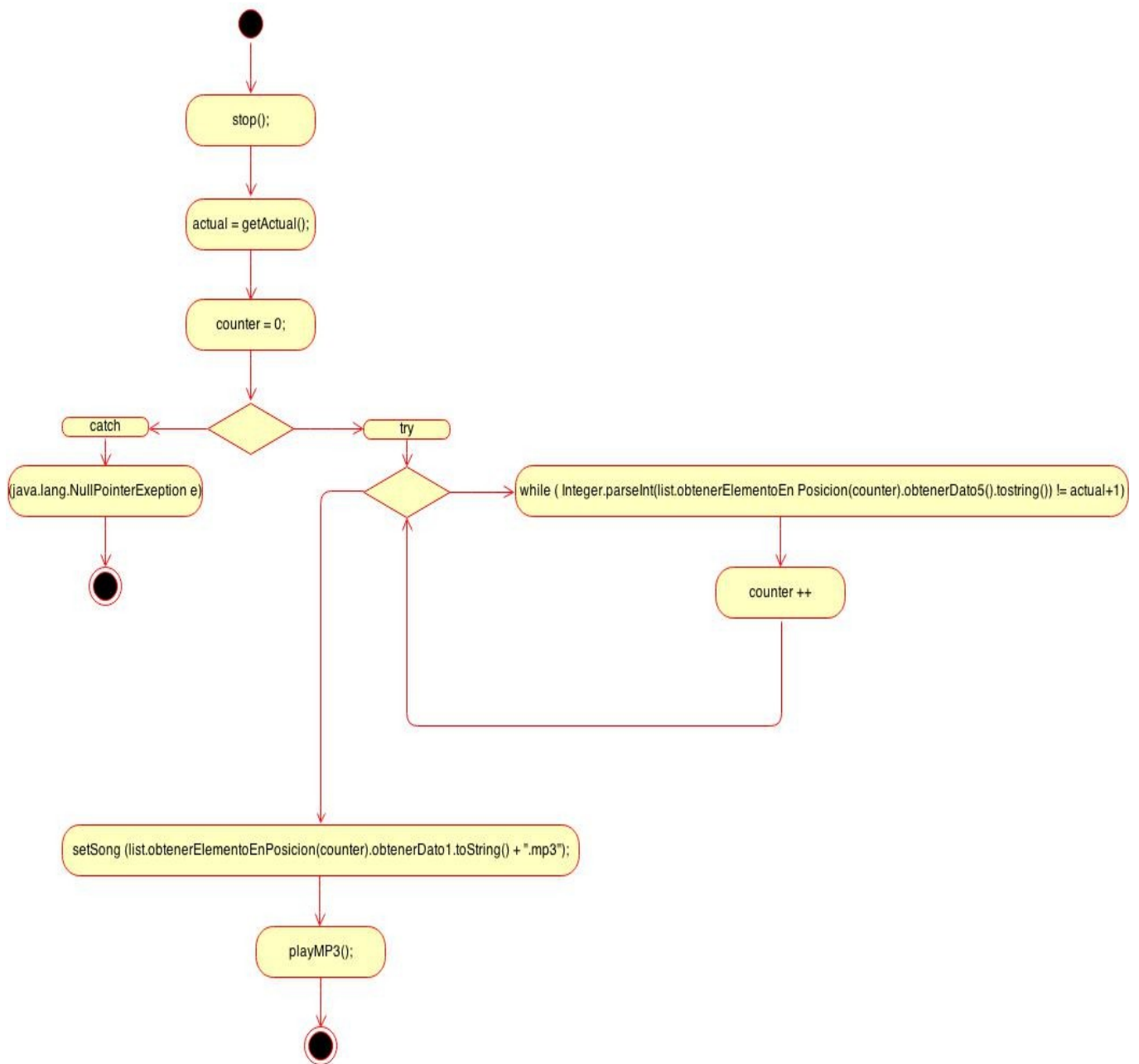
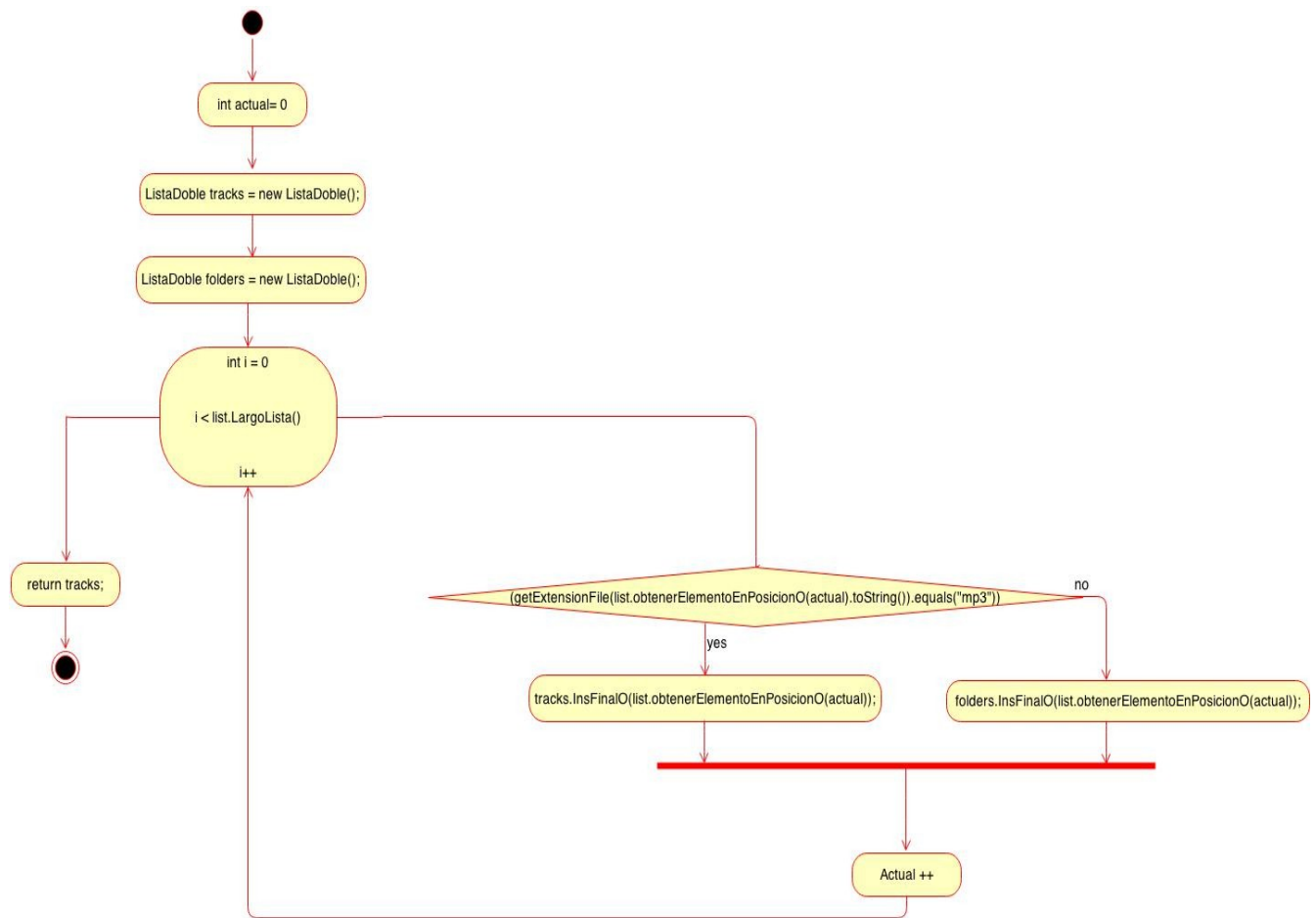


Diagrama 1: Public void next() throws Exception



Algoritmo 2: *public static ListDoble filtrarElementos (ListDoble list)*

## **Librerías usadas**

### **JAudio Tagger:**

Esta librería es utilizada para etiquetar los datos de los archivos de audio. Esta librería ofrece muchos beneficios, es muy completa y esta en constante desarrollo y revisión. Un aspecto esencial de la librería es que ofrece muchas representaciones de tipo objeto, para sustituir la representación por byte.

### **Basic Player:**

Es una clase reproductor, basada en JavaSound API, la cual implementa una interfaz del BasicController. Esta librería nos fue útil a la hora de implementar funciones de play/pause, stop, resume etc...

### **JAVA MP3 ID3**

Esta librería es otra de obtención de metainformación, la cuál extrae datos como el título de la canción, artista y álbum

## **Análisis de resultados**

Los objetivos planteados al principio del proyecto fueron muy complicados de realizar. Dentro de los mayores retos que tuvimos a la hora de realizar la tarea programada, fueron cuestiones externas a la implementación de las estructuras de datos, La generación del repositorio de Github, y la descarga e implementación de la herramienta Apache Ant. Esto se pudo deber a el hecho de estar trabajando en Linux Mint, lo que dificulta la instalación de paquetes ya que la mayoría se deben hacer por consola y la información para hacerlo es escasa. Por otro lado, la creación de código fue más sencilla y rápida gracias al conocimiento previo obtenido en las lecciones de Algoritmos y Estructuras de Datos.



# Manual de Usuario



La interfáz gráfica de usuario esta pensada para ser justo como las ya conocidas por los usuarios de reproductores de música, combinando una organización que represente una sensación de familiaridad, botones y elementos propiamente identificados que hacen que el usuario final pueda utilizar la aplicación de manera intuitiva y un tamaño que pensamos es el adecuado para la navegación, y para ser adaptado al momento de uso en el monitor del computador.

El modo de operación se compone de pasos principales:

**Buscar la canción y seleccionarla:**

Esta fase se puede realizar de 2 formas, mediante las bibliotecas previa y automáticamente diseñadas para mostrar las canciones por nombre, artista, album y género. Si quiere una búsqueda más personalizada, puede emitir un criterio de búsqueda en la barra de search.

**Reproducir y utilizar:**

Una vez buscada y seleccionada la canción, para reproducirla presiona el botón de play, esto va a ejecutar la canción y desplegar la metainformación de la misma, pulsarlo de nuevo pausará la canción. Next y previous adelantan a la siguiente canción y retroceden a la anterior respectivamente.