

Manual Técnico: Reproductor de Música

1. Información General

- Nombre del proyecto: Reproductor de Música
- Autor: Grupo III
- Descripción: Este reproductor de música reproduce canciones que ya están descargadas previamente en los datos del dispositivo y es controlado mediante un joystick que controla la interfaz del reproductor, haciendo uso de las teclas del mismo.

2. Interfaz de Usuario

El proyecto cuenta con una interfaz gráfica. Al dar doble clic sobre una canción, esta se reproduce automáticamente. La ventana de reproducción incluye los siguientes elementos:

- Botones para reproducir la canción anterior y siguiente.
- Botón de pausa y reanudar.
- Botones para subir y bajar el volumen.
- Barra para subir y bajar el volumen.
- Botón para reiniciar la canción.
- Botón para regresar a la lista de canciones.

3. Funcionalidades

- Reproducción de canciones locales en formato MP3.
- Control de volumen (subir, bajar).
- Pausar y reanudar reproducción.
- Ir a la siguiente o anterior canción.
- Reiniciar canción.
- Interacción por joystick o mediante interfaz gráfica.

4. Librerías Utilizadas

- `javazoom.jl.player.Player`
- `javazoom.jl.decoder.JavaLayerException`
- `javazoom.spi.mpeg.sampled.file.MpegAudioFileReader`
- `javax.sound.sampled.AudioFileFormat`
- `javax.swing.*`
- `java.awt.*,java.awt.event.*`
- `java.awt.event.*`
- `java.io.*`
- `java.util.concurrent.atomic.AtomicBoolean`
- `java.util.*`

5. Estructura del Proyecto

El proyecto está organizado de la siguiente manera:

- Carpeta `src`
 - └─ `controller` → Clases: JPlayer.java, VolumeAudioDevice.java
 - └─ `data` → Archivo: music.db
 - └─ `media` → Carpetas: Best Of Rock, Data, Hybrid Theory, Meteora
 - └─ `model` → Clases: Album.java, Artist.java, Composer.java, Singer.java, Song.java, SongDAO.java, Writer.java
 - └─ `ui` → Clase: GUI.java
- Archivo principal: App.java

6. Documentación Técnica

Clases Principales

- JPlayer.java: Controlador principal del reproductor. Se encarga de la reproducción, pausa, reanudación, avance, retroceso y control de volumen. También gestiona la transición automática entre canciones.

- VolumeAudioDevice.java: Subclase personalizada de `JavaSoundAudioDevice` que permite modificar el volumen en tiempo real aplicando un factor de escala a los samples de audio.
- Song.java: Clase que representa una canción, incluyendo atributos como título, duración, ruta del archivo, álbum y artista.
- SongDAO.java: Objeto de acceso a datos (DAO) para obtener canciones desde la base de datos `music.db` ubicada en `src/data`.
- Album.java: Modelo que representa un álbum musical, incluyendo año de lanzamiento, ID de artista y lista de canciones.
- Artist.java: (Abstracta) Clase base para los artistas del sistema. Incluye nombre e identificador.
- Composer.java, Singer.java, Writer.java: Subclases que extienden `Artist` para representar distintos tipos de artistas involucrados en la producción de las canciones.
- GUI.java: Interfaz gráfica del usuario. Permite seleccionar y controlar las canciones mediante botones, sliders y una lista gráfica. También muestra información como progreso y título actual.

Métodos Relevantes

- play(Song song): Inicia la reproducción de una canción y crea un nuevo hilo para manejar el flujo.
- pause() / resume(): Controlan la pausa y la reanudación de la canción.
- volumeUp() / volumeDown(): Ajustan el volumen hacia arriba o abajo en incrementos de 10%.
- repeat(): Reinicia la canción actual.
- next() / previous(): Cambia a la siguiente o anterior canción en la lista.
- setVolume(float volume): Modifica el volumen actual y lo aplica en tiempo real a través de `VolumeAudioDevice`.
- getCurrentPosition() / getDuration(): Devuelven el tiempo actual de reproducción y duración total de la canción, respectivamente.
- getDurationFromFile(File file): Calcula la duración del archivo MP3 mediante `MpegAudioFileReader`.

Base de Datos

Archivo: music.db

Contiene las tablas de canciones (`song`), artistas (`artist`) y sus relaciones (`album_artist`).

Se accede mediante consultas SQL en `SongDAO`.