**Relatório técnico: Desenvolvimento de um media player com JavaFX**

**Davi Matias Soares Genuíno, José Maia da Silva Neto**

1Instituto Metrópole Digital– Universidade Federal do Rio Grande do Norte (UFRN) Natal – RN – Brazil

davimatiassg@gmail.com, maianeto2014@gmail.com

***Abstract.*** *This technical report outlines the development of a media player using the Java language and the JavaFX, following a MVC (Model-View-Controller) architecture. The project aims to provide an efficient playing experience with an attractive graphical interface and friendly user experience.*

***Resumo.*** *Este relatório técnico aborda o desenvolvimento de um media player utilizando a linguagem Java e o JavaFX, seguindo uma arquitetura MVC (Model-View-Controller). O projeto visa proporcionar uma experiência de reprodução eficiente com uma interface gráfica atraente e amigável.*

**1. Introdução**

Este relatório descreve o desenvolvimento de um Media Player utilizando JavaFX, seguindo uma arquitetura MVC (Model-View-Controller). O projeto visa proporcionar uma experiência eficiente na reprodução de mídia, abordando desafios relacionados à interface gráfica, persistência de dados, e lógica de negócios.

**2. Descrição da abordagem de solução do problema**

O projeto desenvolvido segue o paradigma orientado a objetos. As classes foram organizadas de acordo com o padrão MVC, onde o modelo (Model) gerencia os dados, a visualização (View) cuida da interface gráfica e o controlador (Controller) coordena a interação entre o modelo e a visualização. A autenticação de usuários é implementada com diferentes níveis de acesso: usuários VIP podem criar playlists, enquanto que usuários comuns podem apenas adicionar e tocar músicas. Para esse aspecto, foi utilizado o conceito de Herança, já que os dois tipos de usuário descritos são especializações da classe Usuario (classe de modelo que representa a entidade usuário). Decisões de projeto incluíram a escolha do JavaFX para o desenvolvimento da interface gráfica de usuário devido à sua robustez e flexibilidade. A interface gráfica desenvolvida fornece uma fronteira para que o usuário adicione diretórios do seu computador que contenham músicas, para que ele possa escutá-las usando o sistema desenvolvido. Padrões de projeto como Singleton para a instância do Media Player e Observer para atualizações de interface foram aplicados para otimizar o desenvolvimento. A persistência de dados foi implementada utilizando arquivos com extensão .txt, que possuem uma estrutura previamente estabelecida, com cada linha do arquivo contendo dados relevantes para o sistema. O arquivo directories.txt possui uma lista de diretórios selecionados pelo usuário. Nesse arquivo, cada linha possui um path do sistema operacional que será lido por uma classe do pacote DAO. O arquivo songs.txt está organizando de modo que cada linha possua uma música, e na linha estejam especificados os dados dessa música, como o path, o nome da música, o artista, o álbum e outros dados relevantes. O arquivo users.txt é responsável por armazenar os dados dos usuários do sistema. Com a intenção de melhorar a coesão e o acoplamento do sistema, as responsabilidades foram divididas entre as classes. A manipulação de arquivos de dados é realizada pelas classes do pacote DAO (Data Access Object). Essas classes utilizam bibliotecas Java para leitura e escrita de arquivos e guardam os dados contidos nos arquivos além de fornecerem diversas operações, como a inserção e remoção de dados.

**3. Descrição Geral das Estruturas de Dados e Algoritmos**

A principal estrutura de dados utilizada, considerando a eficiência na manipulação de mídia, foi a ArrayList. O DAO foi implementado para separar as operações de acesso aos dados, garantindo a modularidade. Classes de serviço foram criadas para lidar com lógicas específicas, como reprodução e gerenciamento de playlists.

3.1. ArrayList

Para armazenar as músicas, playlists, diretórios e usuários foi utilizada a estrutura de dados Arraylist. Essa estrutura permite que os objetos sejam armazenados numa lista, e fornece operações essenciais, como inserção, remoção e busca. Essas operações foram utilizadas para carregar para o sistema os dados contidos nos arquivos .txt. Para as entidades do domínio, Música, Usuário e Playlist foram desenvolvidas classes DAO, em cada uma dessas classes existe um Arraylist que armazena objetos da referida classe.

**Conclusão**

O desenvolvimento deste Media player utilizando JavaFX permitiu a criação de uma aplicação robusta e modular. As decisões de projeto visaram uma alta coesão e um fraco acoplamento, bem como uma maior escalabilidade e manutenibilidade do sistema, proporcionando uma experiência de usuário aprimorada.

**References**

Schildt, H. (2014), Java The Complete Reference, McGraw-Hill Education.

Bloch, J. (2008). Effective Java, Addison-Wesley.

Sierra, K. and Bates, B. (2005). Head First Java, O'Reilly Media.

Dea, C., Grunwald, G., Pereda, J. and Phillips, S. (2014). "JavaFX 8: Introduction by Example", Apress.

Oracle Corporation. (Online). "How to Write Doc Comments for the Javadoc Tool." [[https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html](https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html)](https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html%5D(https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html)).

Oracle Corporation. (Online). "Java Documentation." [<https://docs.oracle.com/en/java/>].

Baeldung. (Online). "Java Tutorials at Baeldung." [<https://www.baeldung.com/java-tutorial>].

Oracle Corporation. (Online). "JLayer API Documentation." [<https://docs.oracle.com/javase/7/docs/api/javax/swing/JLayer.html>].

Oracle Corporation. (Online). "Java Documentation." [<https://docs.oracle.com/en/java/>].

Stack Overflow. (Online). "JavaFX Questions on Stack Overflow." [<https://stackoverflow.com/questions/tagged/javafx>].

Stack Overflow. (Online). "Java Questions on Stack Overflow." [[https://stackoverflow.com/questions/tagged/java](https://stackoverflow.com/questions/tagged/javafx)].