

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
INSTITUTO METRÓPOLE DIGITAL  
CURSO DE TECNOLOGIA DA INFORMAÇÃO

Music Player

Aluno: Wanderson A. de Oliveira

Turma: 2018.1

Disciplina: Linguagem de Programação II

Professor: Frederico Araujo da Silva Lopes

Natal, 25 de junho de 2018.

## Visão geral

1 – Resumo:

Em busca de pôr em prática o nosso aprendizado na disciplina, fomos instruídos para projetar um Music Player na linguagem Java, capaz de produzir arquivos de áudio individualmente, ou como parte de uma playlist. Nosso projeto contém quatro telas principais, que permite a interação do usuário com o programa dependendo do nível de acesso que o mesmo possui:

1. Tela de Login;
2. Tela Principal;
3. Tela de Criar Playlist;
4. Tela de Adicionar Usuário.

## Padrões de Projeto Utilizados

1. MVC: Modelo de três camadas físicas (Modelo, Visualização e Controle) para uma melhor visualização, inclusão e organização do projeto em questão.
2. DAO: Padrão de projetos onde um objeto: provê uma interface que abstrai o acesso a dados, lê e grava a partir da origem de dados e encapsula o acesso aos dados, de forma que as demais classes não precisam saber sobre isso.
3. Facade: Fornece uma interface unificada para um conjunto de interfaces do nosso subsistema e define uma interface de nível superior que facilita o uso do mesmo.
4. Singleton: Fornece uma classe-objeto capaz de ser instanciada(criada em memória pronta para seu uso) uma única vez e com visibilidade e acessibilidade global dessa instância em um determinado escopo de projeto.

## Especificações

De preferência utilizar como IDE o Netbeans (principal ferramenta utilizada na construção do projeto) para evitar erros de compilação de algumas funções específicas da mesma, como também possuir o JDK em sua última versão.

## Código Comentado

A seguir temos um pequeno resumo sobre cada função de classe participante de cada pacote do Projeto, seguindo o padrão MVC, seguem divididos em:

### I. Player Control

Control Facade: Classe responsável pela Facade do pacote de controle, que contém o construtor de todos os outros.

MP3 Player: Implementa um Mp3 Player usando programação paralela, para isso usamos a biblioteca JLayer e suas classes.

Music: Classe usada para criar uma instância de Música.

Music Control: Classe responsável pelo controle das ações do player, como parar, pausar e tocar uma ou várias músicas.

User: Representação abstrata de quais parâmetros e responsabilidades um usuário deve ter.

Normal User: Estende a classe abstrata User definindo as diferenças de privilégio de um usuário normal, um usuário normal pode reproduzir músicas de uma pasta.

Vip User: Estende a classe User definindo as diferenças de privilégio de um Usuário Vip que; pode adicionar outros usuários, criar e usar suas próprias listas de reprodução e todas as coisas que um usuário normal pode fazer.

User Control: Gerencia a maioria dos métodos relacionados a User, incluindo implementações de VipUser e NormalUser.

Playlist: Classe usada para criar uma instância de Playlist.

ProgressBar: Classe responsável por calcular o progresso da música e devolvê-lo ao usuário.

### II. Player Model

DaoInterface: Uma interface para as classes que lidam diretamente com o acesso a arquivos.

FolderDao: Uma classe que é responsável por ler o arquivo de diretório que contém o caminho para os outros arquivos responsáveis pela persistência de dados.

ModelFacade:

Uma classe projetada usando o padrão do Facade, ela encapsula métodos de outras

objetos na modelagem, a fim de torná-los muito mais fáceis de serem usados juntos.

MusicDao: Uma classe que é responsável por ler o arquivo Musics e gerencia todo o gerenciamento de dados relacionados às músicas.

PlaylistDao: Uma classe que é responsável por ler o arquivo Playlist e gerencia todo o gerenciamento de dados relacionados às playlists.

UserDao: Uma classe que é responsável por ler o arquivo Users e gerencia todo o gerenciamento de dados relacionados aos usuários.

### III. Player View

LoginScreen: Tela inicial onde o usuário insere o seu login e senha para poder acessar suas playlists.

AddUserScreen: Tela que só pode ser acessada por Vips e Admin para criar um novo usuário (seja ele Vip ou não).

MainScreen: Tela principal do programa, que possui três campos principais: Músicas da playlist, Músicas e Playlists. Fora as abas, User, File e Vip que contém funções de deslogar da conta, adicionar músicas ou diretórios e criar playlist ou novo usuário.

Playlist Screen: Tela onde podemos criar uma playlist e adicionar ou remover músicas para a mesma.

## Conclusão

Esse projeto foi importante não só para testar os conhecimentos adquiridos na disciplina mas também para mostrar o quão importante é a interdisciplinaridade com a disciplina de Estrutura de Dados Básicos II, principalmente com o uso das estruturas de árvores (Binária e Patrícia) na construção das listas correspondentes aos Usuários, Músicas, Playlists etc. Como também um exemplo prático de situação que sempre seremos colocados em nossa carreira como programadores, que o diferencial é sempre estar buscando adquirir mais conhecimento para aprimorar nossas habilidades.