

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

INSTITUTO METRÓPOLE DIGITAL

IMD0040 - Linguagem de Programação II - 2023.2

Especificação do Trabalho Final

1. Introdução

O projeto (**Media Player com Java**) consiste em um tocador de MP3 utilizando a biblioteca JLayer (<https://docs.oracle.com/javase/7/docs/api/javax/swing/JLayer.html>), capaz de produzir arquivos de áudio. É necessário que os alunos utilizem conceitos dados na disciplina de LP-II, como:

1. Organização de pacotes;
2. Interface (GUI);
3. Utilização de bibliotecas externas;
4. Documentação JavaDOC;
5. Herança;
6. Polimorfismo;
7. Classe Abstrata;
8. Interfaces;
9. Classes para tratamento de Exceções.
10. Análise sobre Coesão e Acoplamento

O tocador de áudio deve possuir uma interface gráfica. O sistema deve ser capaz de prover um *login* para seus usuários. Diferenciando níveis entre eles.

Controle de Acesso:

- Todos os usuários precisarão *logar* no sistema. Caso não haja login cadastrado, abrir opção para cadastrar usuários dos tipos: Vip e comum.
- Usuários “VIP” podem criar várias *playlists*;
- Usuários comuns podem apenas adicionar e tocar as músicas.
- O Player deve iniciar a aplicação.
 - Usuário comum:
 - O usuário comum pode adicionar diretórios para que o player exiba as músicas.
 - Usuário VIP:
 - O usuário vip terá a possibilidade de ter atrelado à **sua conta**, uma ou mais *playlists* personalizadas.
 - Importante: após o usuário adicionar diretórios de músicas ao player, a lista de músicas aparecerá no player, mesmo que o usuário feche o player, após um novo *login* as músicas devem aparecer listadas.

Salvando os dados da aplicação:

Para manter os dados da aplicação salvos, utilizaremos arquivos de texto que podem ter a extensão ***.txt** ou qualquer uma que o desenvolvedor preferir.

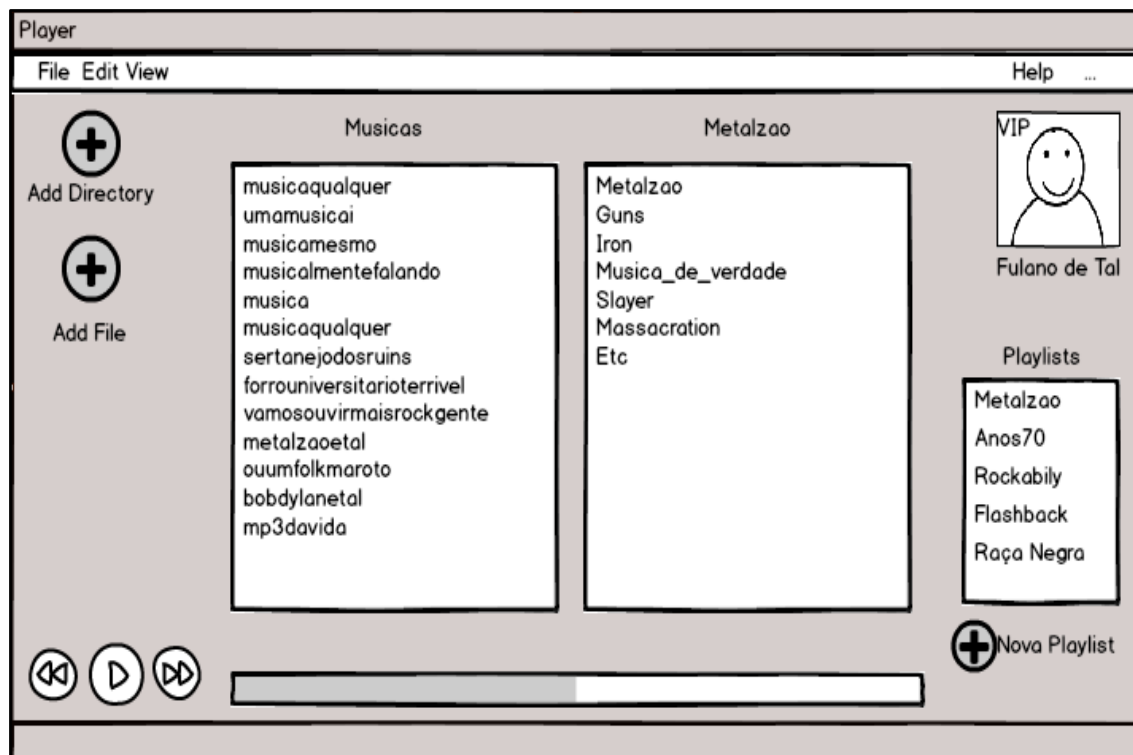
Deve-se manipular esses arquivos de modo a salvar os dados da maneira correta. Os primeiros dois (2) arquivos deverão guardar informações sobre as pastas e sobre as músicas propriamente ditas, um (1) para usuários e ***n*** arquivos para *playlists*. Estes arquivos devem ser lidos quando o programa for aberto e devem ser alterados caso os usuários realizem alterações.

- ***diretorios.txt***: este arquivo será responsável por informar ao *player* quais as pastas serão lidas quando o software for iniciado. O formato pode ser definido pelo usuário, mas recomendamos que cada linha represente uma pasta. Este arquivo deve ser lido a cada vez que o *player* for aberto e deve ser alterado quando o usuário desejar adicionar mais pastas. Quando o *player* for executado pela primeira vez, não existirão pastas cadastradas, consequentemente, não haverá lista de músicas no primeiro acesso.
- ***musicas.txt***: este arquivo salvará apenas o nome das músicas e seus respectivos caminhos. É importante que cada linha do arquivo represente o caminho completo para uma respectiva música.
- ***playlist_xxx.txt***: haverá mais de um arquivo como este, um para cada *playlist*. Como as *playlists* serão privadas por usuário (cada usuário VIP pode criar e editar suas *playlists*), é importante que os arquivos comecem com a autenticação do usuário (nome e id).
- ***usuarios***: este arquivo será responsável pelos dados do usuário. Durante a inicialização, o software carregará os usuários para a memória e poderá fazer a autenticação.

O menu de playlists:

Os usuários VIPs poderão cadastrar músicas em uma *playlist*. Essa *playlist* deverá ter um nome, podendo tal usuário ter mais de uma. A forma como a *playlist* será montada deverá ser decidida pelo programador. Pode ser arrastando de uma lista para outra, abrindo diretórios e selecionando arquivo por arquivo, etc. Essa adição pode ser feita de qualquer maneira.

Exemplo de tela:



Extras:

O aluno pode melhorar seu projeto, incluindo alguns itens extras, tais como:

- O aluno pode pesquisar uma maneira de mostrar a barra de execução da música. A biblioteca possui um método para pegar o tempo atual da música. Essa barra deve permitir ao usuário avançar e voltar.

2. Entrega e Avaliação

Seu grupo deverá submeter, via SIGAA, um arquivo compactado contendo:

- Código fonte do sistema desenvolvido, incluindo um documento README.TXT contendo instruções de como se pode compilar o código fonte.
- Apresentação em PDF (slides).
- Relatório Técnico, contendo pelo menos as seguintes seções:
 - Introdução: contendo uma breve descrição do problema abordado e do que será apresentado em seu relato;
 - Descrição da abordagem de solução do problema: Descrevendo o projeto OO (diagramas de classes), e destacando as decisões de projetos tomada, bem como identificando padrões de projeto aplicados;
 - Descrição geral das estruturas de dados utilizadas com explicação dos algoritmos utilizados;
 - Conclusão;
 - Referências;

O relatório deverá ser feito seguindo o template da Sociedade Brasileira de Computação (SBC) que pode ser encontrado no seguinte endereço:

<http://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>

Tabela de pontuação para correção:

Codificação	Classes Abstratas, Interface e Tratamento de Exceções	0,5
	Documentação JavaDOC	0,5
	Herança e Polimorfismo	1,0
	Interface Gráfica	2,0
Apresentação	Relatório	1,0
	Apresentação	2,0
Objetivos	Controle de usuários	2,0
	Armazenamento das playlist (músicas, diretório etc.)	1,0
	Plus (ponto extra)	
Total de pontos		10,0