
Sistema de Cadastro e Pesquisa de Álbuns e Músicas

Plano de Testes

Membro:

Caio Lucas

Julho/ 2020

Histórico de Revisões

| Data | Versão | Descrição | Autor |
|------------|--------|-----------------|------------|
| <Julho/20> | <1.0> | Release Inicial | Caio Lucas |

1. Introdução

1.1 Objetivos

Esse documento do Plano de Testes do teste realizado pela empresa DTI tem com os seguintes objetivos propostos:

- Identificar informações de projeto existentes e os componentes que devem ser testados.
- Listar os Requisitos a Testar.

2. Instruções para utilizar o programa.

Para utilizar o programa é necessário possuir uma IDE com Java 8 ou superior para poder compilar o devido código e executá-lo.

2.1 Instruções

- Acessar o menu de opções e escolher a opção desejada.
- Digitar a opção utilizando o console da linguagem.
- 1 – Cadastrar Álbum.
- 2 – Cadastrar Música do álbum
- 3 – Pesquisar Álbum pelo título.
- 4 – Pesquisar Álbum pelo nome
- 5 – Pesquisar Música.
- 6 – Gerar Playlist.
- 7 - Sair
- Informar os devidos dados solicitados pelo programa.

Nota: Para executar o programa é necessário compilar pela classe App ou utilizar a tecla de

atalho “F5”. Sendo não necessário a instalação de dependências para este projeto.

Nota: Para salvar a música na lista de favoritas é necessário digitar no console as seguintes entradas: “sim” ou “Sim” e “nao” e “Nao”.

3. Premissas Assumidas

Foi escolhida a linguagem Java para o desenvolvimento do sistema requerido devido a maior afinidade com a mesma. Partindo do pressuposto de um sistema de música, foram usadas técnicas de programação orientada a objetos e um pouco de AED (Algoritmos e Estrutura de dados).

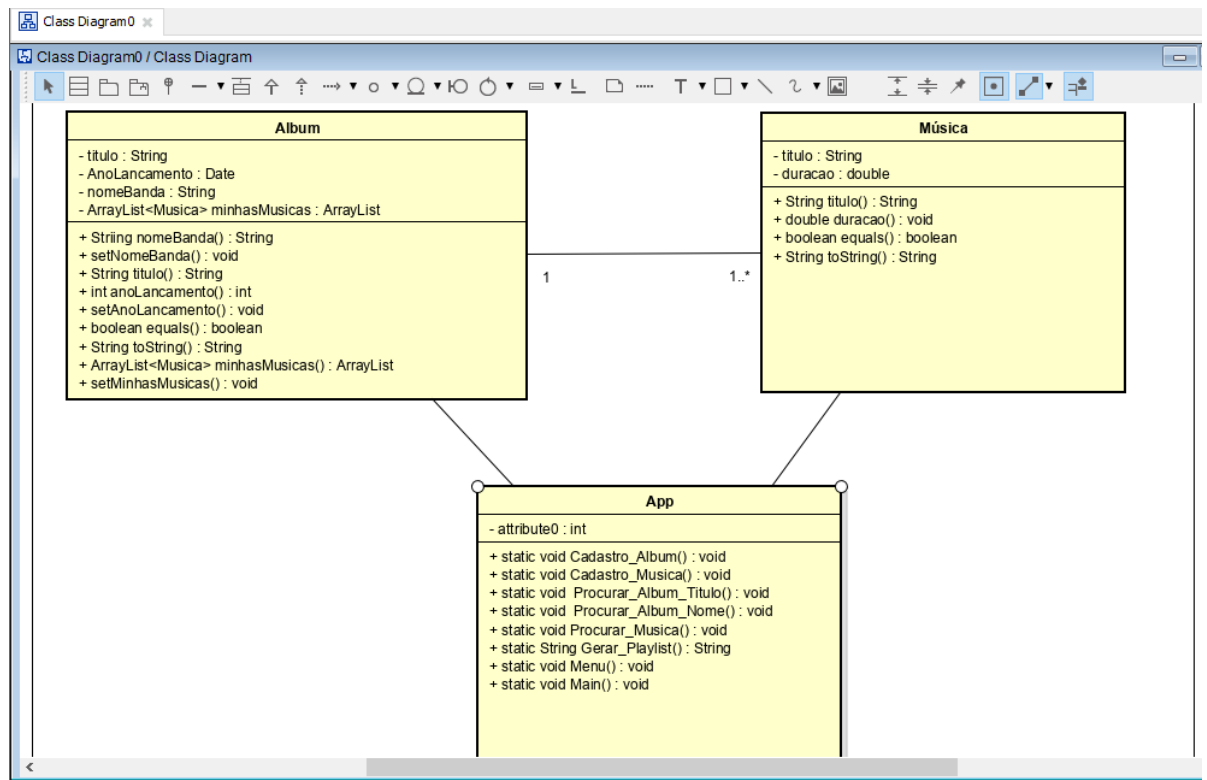
O sistema se compõe pelas seguintes classes: Álbum, Música e App, sendo não necessário a utilização de heranças, somente “override e polimorfismo”. Devido ao curto tempo, não usei o Java conectado ao banco de dados, mas sim a utilização do ArrayList do próprio Java que para o solicitado será suficiente, tendo como único problema a não persistência dos dados salvos, como ocorre em um BD (Banco de Dados).

Seguindo a regra de negócio, é necessário cadastrar um álbum e suas respectivas músicas, para que assim realize a procura das mesmas e montar um playlist.

4. Decisão de projeto

Observando o foco do sistema juntamente com a necessidade do “Billie”, o sistema foi construído em cima de técnicas de orientação a objeto para maior conversação entre os objetos e as classes. Para o programa não ficar muito acoplado, tomei cuidado em não criar grandes dependências entre os métodos das classes, para ficar mais fácil de corrigir futuros erros descoberto.

O sistema também utilizasse da reutilização de métodos já criados, como o de Cadastrar Álbum e Cadastrar Música, poupando a necessidade de novos métodos. Segue abaixo o diagrama do sistema.



5. Estratégia de Teste

5.1 Tipos de Teste

Nota: Os testes abaixo são os possíveis testes a serem realizados. Posteriormente poderão ser decididas quais técnicas de testes serão usadas e os critérios para dar o teste como finalizado, juntamente com considerações observadas acerca dos testes.

As transações abaixo se referem às “transações lógicas de negócio”. Essas transações são definidas como funções específicas que um usuário final do sistema é suposto de executar ao usar a aplicação, tais como adicionar uma dada informação.

2.1.1 Teste de Função

| | |
|--------------------------|---|
| Objetivo do Teste: | Garantir a funcionalidade apropriada do alvo do teste, incluindo navegação, entrada de dados, processamento |
| Técnica: | <p>Executar os casos de uso, fluxo de caso, testando com dados válidos e inválidos.</p> <ul style="list-style-type: none"> Resultados corretos com dados válidos. Respectivas mensagens de erro são apresentadas para dados inválidos. Regras de negócio usadas apropriadamente. |
| Critério de Finalização: | <ul style="list-style-type: none"> Todos os testes forem realizados. Todos os erros forem tratados |

| | |
|-----------------------------|--------|
| Considerações Especiais: | Nenhum |
|-----------------------------|--------|