

Relatório teste DTI

Pré-requisito:

Node 10+

Como executar o programa:

```
npm i  
npm start
```

Para o projeto, decidi implementar um modelo n-tier onde temos a parte de **model** (responsável por especificar a estrutura de dados dos álbuns, das músicas e da playlist) e temos a parte da **action** (onde se executa toda lógica da aplicação).

Action

albumAction.js - Possui os métodos de pesquisa e cadastros de albums.

cadastroAlbum() - Recebendo input do usuário com todos dados necessários para cadastrar um álbum, armazenamos a informação de um novo objeto do tipo Álbum dentro de um vetor de Álbuns.

O tipo Álbum possui o formato:

```
Album {  
  Titulo: string,  
  ano: number,  
  nome: string,  
  musicas: [ [musicas] ]  
}
```

Como um álbum possui várias músicas, decidi colocar um array do tipo Musicas dentro do objeto álbum para facilitar a manipulação.

pesquisarAlbum(albums) - O método recebe uma lista de álbuns para poder efetuar a pesquisa. Faço verificações caso essa lista esteja vazia e alerto o usuário para cadastrar albums. Daí é bem intuitivo, ao encontrar o álbum a partir da pesquisa, retorno todas informações para o usuário referente aquele álbum

pesquisarMusicas(albums) - Funciona da mesma forma que o método acima. Como não foi informado o que deve ser apresentado para o usuário, tomei duas decisões: ao filtrar por nome de música, imprimo na tela o nome da música caso presente em algum álbum) ou que não foi encontrado a música. No filtro de nome de banda, imprimo seus álbuns e suas músicas.

gerarPlaylist(albums) - Coloco todas músicas existem dentro de um array de músicas e faço ordenação por músicas favoritas. Faz cálculo duração (em segundos) e vou iterando sobre esse array. Se a música que já tenho mais a atual for menor que 1 hora, vou adicionando na playlist. Com isso minha playlist tem preferência por músicas favoritas primeiramente.

Decidi manter o sistema ativo até a opção 5 (de sair) ser chamada. Com isso, após cada interação do usuário com o sistema, o mesmo volta para o menu principal. Isso é fundamental já que não temos banco de dados e teria que cadastrar tudo novamente caso quisesse fazer outro teste.