Nome do grupo:

Caio Nogueira Cordeiro

Victor Gabriel Tenório Oliveira

Nome do projeto:

MusicManager

Gerenciador de músicas.

Repositório de código no GITHUB:

<https://github.com/caionog/MusicManager>

Diagrama UML: <https://github.com/caionog/MusicManager/blob/master/Diagrama%20UML.png>

Modelo Navegacional: <https://github.com/caionog/MusicManager/blob/master/Modelo%20Navegacional.png>  
  
Link do vídeo: <https://youtu.be/qxnRA2fXKAw>

Descrição geral do projeto

O projeto é um sistema de gestão de músicas que extrai metadados de arquivos .mp3 para armazenar em um repositório. O sistema tem um sistema do login com usuários normais e administradores. Apenas administradores podem fornecer arquivos .mp3 para serem armazenados no repositório e deletar músicas já armazenadas. Qualquer usuário pode buscar informações de uma música ou playlists, gerar um relatório com informações inferidas do repositório de música, favoritar músicas ou playlists e organizar tais músicas em playlists. As playlists podem ser públicas ou privadas, ou seja, apenas playlist públicas serão exibidas nas buscas e consideradas nas estatísticas do relatório.

O usuário possui nome, email e senha para efetuar o login. Cada música possui ID, nome, gênero, artista, duração. Na tela principal o usuário poderá entrar no seu perfil e alterar seus dados ou, ainda na tela principal, acessar as funcionalidades do sistema como o gerador de relatório ou entrar no buscador e filtrar músicas e playlists para adiciona-las em seus favoritos.  
  
Exemplos de relatório: quantas vezes uma música aparece em todas as playlist, frequência de um artista em todas as múscias ou numa playlist específica, frequência de todos os gêneros em todas as musicas ou numa playlist específica, qual música ou playlist foi mais favoritada.

Resumo das classes do projeto  
As classes bases (beans) estão localizadas em negocio.beans e são utilizadas pelos repositórios, controladores, fachada e controladores de tela. Além das 3 classes base (User, Music, Playlist), existem algumas calsses enum auxiliares como Genre, \_Visibility e UserPermission. Também existe uma classe Filter que encapsula e permita transportar as informações sobre o filtro que o usuário quer aplicar entre a tela de filtro e a tela principal.  
  
Todas as classes repositório da camada de data possuem a estratégia de singleton e uma interface para os controladores. Os repositórios armazenam dados em .txt.  
  
A classe fachada possui a estratégia singleton e está localizada no pacote negócio. A fachada guarda o usuario logado e os settings (tipo Filter que está no negocio.beans).   
  
A parte da gui está separada das camadas de negócio e repositórios, com os seus .fxml no pacote gui.view e os controladores em gui.controllers.   
  
O main do projeto é a classe applications no pacote gui e todos os arquivos e repositórios utilizados pelo projeto estão no pacote data.  
  
As bibliotecas externas utilizadas ( tika.apache e javafx) serão baixadas automaticamentes pelo maven. O seu pom.xml está na source do projeto. As entregas anteriores estão localizadas na pasta entrega contendo o modelo navegacional, diagrama de classes e documento de entrega  
  
gerador de relatório não foi implementado

Requisitos do projeto

Lista de requisitos

* O gerenciador tem um sistema de login e senha. Existe usuários normais e administradores. O sistema tem um repositório de usuários para lembrar das contas já registradas.
* O gerenciador lê arquivos do tipo .mp3 a partir de um path e extrai suas informações. Existe um repositório de musicas para armazenar os metadados.
* Apenas um usuário administrador pode adicionar ou deletar músicas no repositório de músicas.
* Qualquer usuário pode alterar seus dados (email, senha, nome), criar playlists agrupando músicas e favoritar músicas ou playlist. Existe um repositório de playlists para armazenar as músicas agrupadas. Inicialmente uma playlist criada é privada e somente pode ser vista pelo criador que pode escolher alterar a visbilidade.
* O gerenciador tem um buscador que retorna musicas ou playlists públicas.
* O gerenciador pode fornecer informações relevantes sobre uma música ou playlists a partir do buscador ou do gerador de relatório.