**1 – Banco de dados**

**Tabelas Criadas:**

**- artistas (id, nome, genero)**

id - > primary key

nome -> not null

**- histórico\_buscas (id, usuario\_id, termo\_busca)**

id - > primary key

termo\_busca -> not null

usuario\_id -> foreign key

**- musicas (id, titulo, duracao, genero, data\_lancamento, artista\_id)**

id - > primary key

titulo, duracao - > not null

artista\_id -> foreign key

**- playlist\_musicas (playlist\_id, musica\_id)**

(playlist\_id, musica\_id) - > primary key composta

playlist\_id, musica\_id - > foreign key

**- playlists (id, nome, usuario\_id)**

id - > primary key

nome -> unique

usuario\_id - > foreign key

**- usuário\_musica\_curtidas (usuario\_id, musica\_id, acao)**

(usuario\_id, musica\_id) - > primary key composta

usuario\_id, musica\_id - > foreign key

**- usuários (id, nome, login, senha)**

id - > primary key

nome, login, senha - > not null

login - > unique

**2 – Classes Criadas**

**Pacote DAO**

**- ArtistaDAO**

Possui as funções para busca de artistas no bd

**- AvaliacaoDAO**

Possui as funções para registrar avaliações (curtida ou descurtida), remover avaliações, verificar avaliações (verifica se a música passada como argumento tem alguma avaliação) e inverter avaliação (inverte curtida para descurtida e vice versa / usada quando uma música curtida é descurtida pelo mesmo usuário

**- Conexao**

Estabelece a conexão com o banco de dados

**- HistoricoDAO**

Possui as funções para retorno do histórico de buscas e histórico de musicas avaliadas

**- MusicaDAO**

Possui as funções para busca de músicas e registro de buscas no histórico

**- PlaylistDAO**

Possui as funções para criar, deletar e buscar playlists, buscar todas as músicas de uma playlist e adicionar e remover uma música de uma playlist,

**- UsuarioDAO**

Possui as funções para inserir um usuário no bd, e verificar a existência de um usuário específico no bd

**Pacote View**

**- CadastroFrame**

Tela(JFrame) de cadastro

**- LoginFrame**

Tela(JFrame) de login

**- MusicaDialog**

Tela(JDialog) exibido ao clicar em uma música, contem o nome da música, nome do artista e botões para curtir e descurtir

**- PlaylistCriacaoDialog**

Tela(JDialog) exibido ao clicar no botão de criação de playlists (localizado no PrincipalFrame na aba playlist), contéma opção para colocar o nome da playlist

**- PlaylistEdicaoMusicaDialog**

Tela(Jdialog) exibido após a view PlaylistCriacaoDialog ou ao clicar no boão de editar playlist (localizado no PrincipalFrame na aba playlist), contém um tela de buscas, uma tabela com as músicas buscadas e uma tabela com as músicas as serem adicionadas na playlist sendo criada ou editada

**- PrincipalFrame**

Contém um JTabbedPane com 3 abas:

**Home:**

A tela exibida após o login, contéma a barra de busca de músicas e a tabela para exibição das músicas, as músicas buscadas podem ser selecionadas para avaliação

**Playlists:**

Tela que exibe as playlists, além das funcionalidas para criação, edição e exclusão de playlists

**Histórico:**

Tela que contem 3 tabelas com: histórico de musicas buscadas, musicas curtidas e musicas descurtidas. Clicar em alguma das músicas buscadas redireciona o usuário para a aba Home com a busca clicada realizada, clicar em alguma das músicas curtidas ou descurtidas abre o MusicaDialog, permitindo trocar a avaliação da música

**Pacote Controller**

**-** **ControllerCadastro**

Contém a lógica do cadastro de usuários

**-** **ControllerLogin**

Contém a lógica do login de usuários

**-** **ControllerMusicaDialog**

Contém a lógica da avaliação de músicas

**-** **ControllerPlaylistCriacaoDialog**

Contém a lógica para criação de playlists no bd

**-** **ControllerPlaylistEdicaoMusicaDialog**

Contém a lógica para a associação entre playlists e músicas

**-** **ControllerPrincipalHistorico**

Contém a lógica para exibição de histórico de buscas e avaliações

**-** **ControllerPrincipalHome**

Contém a lógica para a busca de músicas

**-** **ControllerPrincipalPlaylists**

Contém a lógica para exibição de playlists, e das músicas associadas a elas

**Pacote Util**

- SessaoUsuario

Mantem armazenado o usuário atualmente logado em uma variável static para que todas as classes tenham acesso

**Pacote Main**

- Main

Início do programa