



UNIVERSIDADE DE COIMBRA

Departamento de Engenharia Informática  
Faculdade de Ciência e Tecnologia da Universidade de Coimbra

Licenciatura em Engenharia Informática

Base de Dados Meta 2

Trabalho Realizado por:

João Moreira – 2015230374

Ricardo Tavares – 2014230130

# Índice

- 1- Resumo
- 2- Manual de Instalação
- 3- Manual de Utilizador
- 4- Explicação das funções e respetivo código SQL
- 7- ER
- 7- Diagrama Físico

## **Resumo**

Este projeto tem como objetivo criar um sistema de gestão e partilha de músicas e informação relacionada com funcionalidades semelhantes aos arquivos ALLMusic.com e IMDB.com.

Cada utilizador tem acesso às músicas existentes na base de dados, podendo criar playlists, criticar álbuns com texto e pontuação. No caso do utilizador ser editor, este pode dar permissões de editor a outros utilizadores, editar conteúdos de álbuns, músicas, artistas, etc. e adicionar ficheiros (de músicas), novos álbuns, artistas e músicas.

## **Manual de Instalação**

Para utilizar, é necessário compilar o código fonte localizado na pasta “/src” e executar o ficheiro compilado “Client.java”.

Para iniciar a base de dados utilizar um cliente SQL da Oracle e executar os comandos localizados no ficheiro “Tabelas.sql”.

De forma a que a base de dados não esteja vazia inicialmente é também fornecido o ficheiro “Dados.txt” com comandos SQL com vários utilizadores, músicas, álbuns e artistas já pré-definidos.

Foi utilizada a linguagem de programação Java e a ligação com a base de dados Oracle SQL foi feita através do JDBC.

## Manual de Utilizador

Para utilizar a aplicação é fornecido um menu inicial no qual tem as opções:

- 1 – Login
- 2 – Registo
- 3 – Sair

O utilizador terá de introduzir um destes números de modo a escolher uma opção.

Escolhendo “3 – Sair” a aplicação irá terminar, “2 – Registo” o utilizador será levado para um menu no qual lhe é pedido para introduzir um “username”, “password” e se é ou não editor (introduzindo 1 no caso de ser ou 0 no caso de não ser). Após isto irá aparecer uma mensagem a dizer se o registo foi completo com sucesso ou não. Sendo bem-sucedido entra logo na aplicação.

Escolhendo “1 – Login” o utilizador, previamente registado, insere o seu utilizador e palavra chave e depara-se com o seguinte menu:

- 1 – Listar Albuns            - Irá mostrar todos os álbuns de todos os artistas.
- 2 – Pesquisa                - Será levado para o menu em que pode fazer pesquisas de artistas/álbuns.
- 3 – Adicionar               - Será levado para o menu em que pode adicionar artistas, álbuns ou músicas (no caso de ser editor).
- 4 – Gestão                  - Será levado para o menu em que pode dar privilégios a outro utilizador (no caso de ser editor).
- 5 – Playlists                - Será levado para o menu em que pode criar/ver playlists.
- 6 – Upload/Download -Será levado para o menu em que pode fazer upload/download de músicas.
- 7 – Sair                      - Sairá da aplicação.

Ao entrar no menu listar álbuns, o utilizador pode seleccionar o álbum que pretende, obtendo mais informação sobre esse mesmo álbum. Após isso pode escrever uma crítica ou editar o género musical.

No menu de pesquisa é possível pesquisar por artista e por álbum. Pesquisando por artista, obtemos a informação do artista, os álbuns associados ao mesmo e as músicas (singles) do mesmo, para além disto temos a opção de editar a biografia do artista.

Pesquisando por álbum é nos dado as informações do mesmo, além das músicas e das críticas.

No menu de “Adicionar”, terá de escolher se pretende adicionar uma música, um álbum ou um artista.

Em “Gestão”, temos a opção de dar privilégios de editor a um utilizador.

“Playlists” permite-nos listar todas as playlists de todos os utilizadores, ver o conteúdo de uma das playlists e por fim criar uma nova playlist.

No caso do “Upload/Download” a opção Upload permite passar a localização do ficheiro de música para a base de dados e a opção download obter esse mesmo caminho, dada a música pretendida.

## Explicação das funções e respectivo código SQL

### Registo

```
String query = String.format("INSERT INTO utilizador (username, password, editor) VALUES ('%s', '%s', %d)", user, pass, editor);
```

O SQL tenta criar um utilizador com os dados passados pelos utilizador. No caso de já existir um username, é devolvido um erro e o utilizador terá de escolher um novo nome de utilizador.

### Login

```
String query = String.format("SELECT USERNAME FROM UTILIZADOR WHERE USERNAME like '%s' AND PASSWORD like '%s'", user, pass);
```

É verificado na base de dados, se os campos preenchidos estão corretos.

### Adição de Música, Álbuns, Artistas, Playlists e Criticas

Na adição destas tabelas foi criado um gerador de id:

```
CREATE SEQUENCE idmusica_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE idalbum_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE idartista_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE idplaylist_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE idcritica_seq START WITH 1 INCREMENT BY 1;
```

Para adicionar uma música, álbum ou artista é necessário fazer a verificação de que o utilizador é um editor:

```
if(checkEditor(user)) {
    String query = String.format("INSERT INTO musica (idmusica, nome, data) VALUES (idmusica_seq.nextval, '%s', TO_DATE('%s', 'dd/mm/yyyy'))", name, data);
```

As adições de álbuns e artistas são semelhantes às de músicas.

A criação de playlists e criticas é feito de modo semelhante e dá-se logo a associação do utilizador que a fez e, no caso dos álbuns a associação do álbum criticado, no caso dos playlists são associadas as devidas músicas, de modo a que esta contenha as músicas que o utilizador previamente escolheu.

### Ver álbuns

```
String query = "SELECT * FROM album";
```

Esta função permite-nos ver todos os álbuns existentes na base de dados.

```
String query = String.format("SELECT a.nome NOME FROM artista art, album a, album_artista aa WHERE aa.album_idalbum = a.idalbum " +
    "AND aa.artista_idartista = art.idartista AND art.nome like '%s'", nameArtist);
```

Esta query é utilizada para quando fazemos uma pesquisa de álbuns pelo artista que o lançou.

```
String query = String.format("SELECT nome, genero, data FROM album WHERE nome like '%s'", x);
String queryReviews = String.format("SELECT a.nome, u.USERNAME, c.TEXTO, c.PONTUACAO " +
    "FROM ALBUM a, CRITICA c, UTILIZADOR u, CRITICA_ALBUM ca, CRITICA_UTILIZADOR cu WHERE ca.ALBUM_IDALBUM = a.IDALBUM " +
    "AND ca.CRITICA_IDCRITICA = c.IDCRITICA AND cu.UTILIZADOR_USERNAME = u.USERNAME " +
    "AND cu.CRITICA_IDCRITICA = c.IDCRITICA AND a.IDALBUM = %d", getAlbumID(x));
```

A obtenção de dados dos álbuns é feito com recurso a duas queries, sendo que uma acede às informações do álbum em si, enquanto que a segunda encontra as criticas feitas a este mesmo álbum.

Para além disto foi também implementada uma query que contra as músicas associadas a esse álbum.

As pesquisas para artista são efetuadas de forma semelhante às dos álbuns.

Encontrar as músicas de cada artista

```
String query = String.format("SELECT m.nome MUSICA FROM ARTISTA a, MUSICA m, MUSICA_ARTISTA ma " +  
    "WHERE m.IDMUSICA = ma.MUSICA_IDMUSICA AND a.IDARTISTA = ma.ARTISTA_IDARTISTA " +  
    "AND a.IDARTISTA = %d", getArtistID(nameArtist));
```

Este comando permite mostrar todas as músicas associadas a um determinado artista.

Tornar utilizador em editor

```
if(checkEditor(user1)){  
    String query = String.format("UPDATE UTILIZADOR SET EDITOR = 1 WHERE USERNAME like '%s'", user2);
```

Para colocar um utilizador com permissões de editor fazemos update na tabela.

Upload / Download

Como a nossa base de dados não foi desenhada para receber ficheiro apenas armazenamos o caminho do ficheiro, sendo que para fazer o “Upload” fazemos “update” a este, enquanto que o “Download” recorre a um “select” do devido ficheiro.

### Resumo de cada função:

A classe Client.java trata apenas dos inputs do cliente, passando os valores para a classe Main.java, que os traduz em código SQL e comunica com a BD, por esse motivo apresentamos apenas os métodos da classe Main.java

**public void** connect() - Função de ligação entre o cliente Java e a BD.

**public boolean** login(String user, String pass)- Procura o utilizador e respetiva password na base de dados e devolve verdadeiro caso encontre.

**public boolean** registo(String user, String pass, **int** editor) - Regista o novo utilizador na base de dados e devolve verdadeiro caso suceda.

**public** Boolean addMusic/addAlbum/addArtist - Registam o novo elemento na respetiva tabela, permitindo esta ação apenas a editores.

**public void** getAlbums() - Imprime a lista de todos os álbuns existentes na BD.

**public void** musicByAlbum(String album) - Imprime a lista das músicas contidas num dado álbum.

**public void** getAlbumsByArtist(String nameArtist) - Imprime os álbuns de um determinado artista.

**public void** getMusicByArtist(String nameArtist) - Imprime todas as músicas de um determinado artista, incluindo singles.

**public void** getMusicians(String name) - Imprime os músicos que constituem um determinado artista.

**public void** getArtistData/getAlbumData - Imprime as informações detalhadas de um Artista/Álbum.

**public** Boolean editAlbumDetails/editArtistDetails - Permite a edição dos dados de um Artista/Álbum.

**public** Boolean addReview(String user, String album, String review, **int** score) - Regista uma nova crítica a um álbum.

**public boolean** makeEditor( String user1, String user2) - Torna um utilizador num editor.

**public boolean** checkEditor(String user) - Devolve verdadeiro caso o utilizador seja editor, falso se não o for.

**public boolean** makePlaylist(String user, String nome, ArrayList<String> lista) - Cria nova playlist na base de dados.

**public void** playlists() - Imprime a lista de todas as playlists e os utilizadores que a criaram.

**public void** getPlaylistInfo(String nome) - Imprime a lista de músicas de cada playlist.

**public void** musicList() - Imprime a lista de todas as músicas da base de dados.

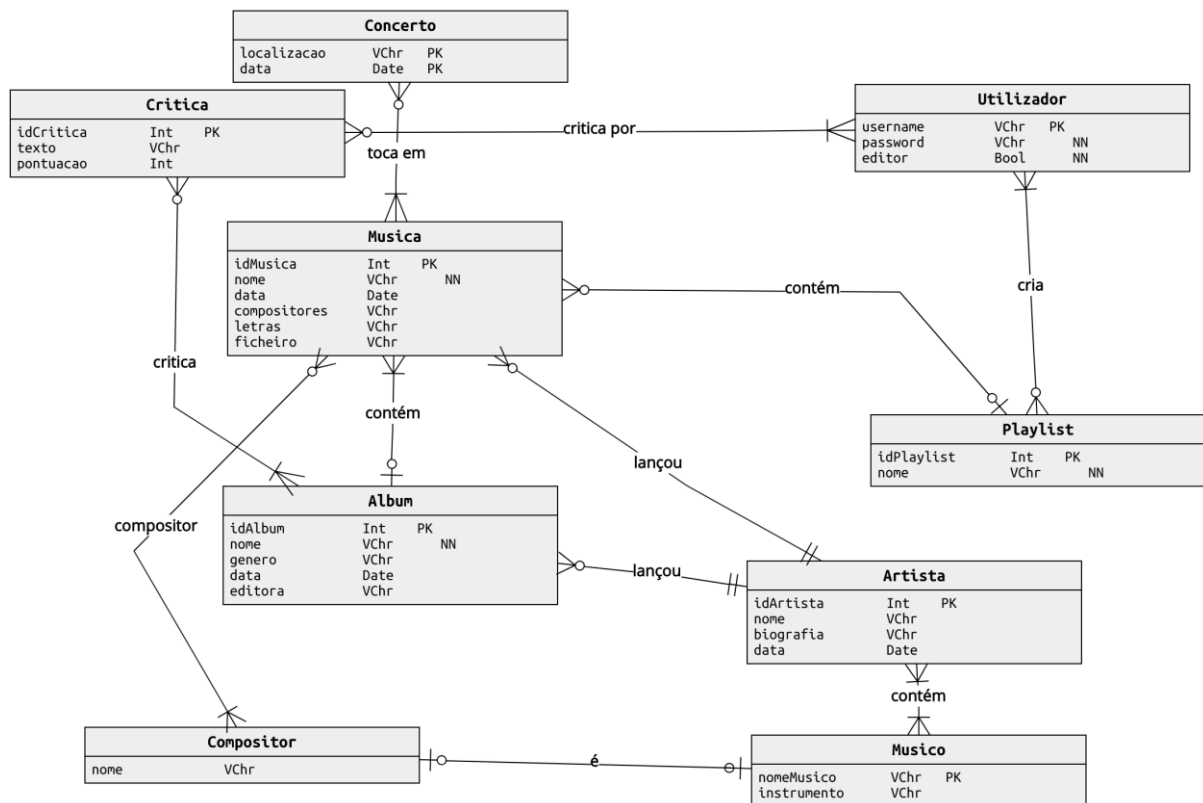
**public boolean** upload(String nome, String caminho) - Atualiza o filepath da música.

**public boolean** download(String nome) - Devolve o filepath da música escolhida.

**public int** getAlbumID/getArtistID/getMusicID/getPlaylistID - Função auxiliar para obter os IDS das tabelas.



## ER



## Diagrama Físico

