



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

DROP MUSIC

Meta2

Elaborado por:

Pedro Martins Mendonça

email:uc2016237295@student.uc.pt

Rui Miguel Freitas Rocha

email:uc2010135453@student.uc.pt

Conteúdo

Introdução.....	4
Arquitetura do Sistema	5
Diagramas Conceptual ou ER (Entidade Relacionamento) e Físico	6
Queries usadas e funcionalidades implementadas	7
Consultar álbuns por artista.....	7
Consultar álbuns por nome.....	7
Consultar musicas por nome	7
pesquisar musica por álbum.....	7
pesquisar musica por artista.....	7
Pesquisar artistas por nome	7
Pesquisar artista por álbum	8
Pesquisar artista por musica	8
Pesquisar concerto por artista.....	8
Pesquisar criticas por álbum.....	9
Listar todos os géneros musicais.....	9
Listar todas as editoras	9
Listar todos os artistas.....	9
Listar todos os artistas do tipo “musico”	9
Listar todas as musicas.....	10
Listar todos os álbuns	10
Pesquisar o ID de um álbum pelo seu nome	10
Pesquisar todos os compositores.....	10
Pesquisar ID de uma musica pelo seu nome	10
Pesquisar ID de concerto por “data”	10
Pesquisar um concerto de um artista pelo nome	11
Obter media da pontuação de um álbum	11
Obter ID de uma playlist pelo nome	11
Listar todas as playlists de um utilizador	11
Listar todas as playlists publicas de todos os utilizadores.....	11
Obter o id de um concerto por nome de artista e data	12
Actualizar pontuação de um álbum por id do álbum.....	12
Tornar uma playlist privada.....	12
tornar uma playlist publica	12
Listar musicas de uma playlist de um utilizador	12
Inserir critica	13

Inserir concerto de um artista	13
Criar um concerto	13
Associar musicas a álbuns.....	13
Criar um álbum	13
Inserir compositores	14
Inserir artistas.....	14
Inserir géneros musicais	14
Criar uma playlist.....	14
Inserir musicas em playlists	14
Mudar data de um concerto de um artista.....	14
Mudar letra de uma musica pelo id da musica	15
Mudar o caminho(path) de uma musica pelo seu ID	15
Mudar a password de um utilizador.....	15
Testes de usabilidade	16
Distribuição de tarefas	17
Manual de utilizador	18
Conclusão	19

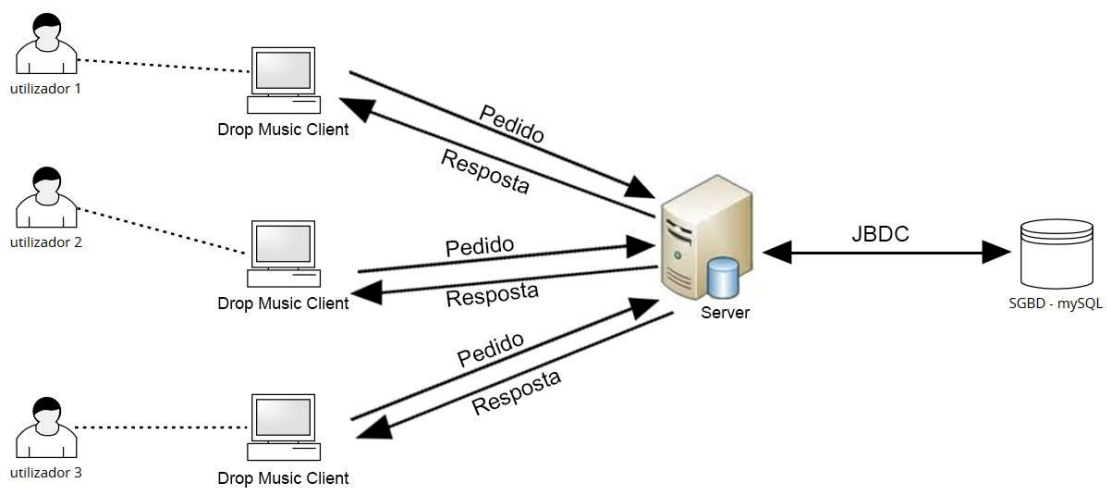
Introdução

Neste trabalho foi nos propostos, criar uma arquitetura cliente-servidor, com recurso a um SGBD (sistema de gestão de base de dados) para o sistema DropMusic. Para isso usamos a linguagem Java para fazer a aplicação, com recurso a bibliotecas externas para ligar ao banco de dados. O SGBD usado foi o mysql, para gerir e armazenar todos os dados da aplicação.

Neste relatório, também se encontra, a arquitetura mais detalhada bem como os diagramas ER (entidade Relacionamento) ou conceptual, diagrama físico final, e uma check list, com os requisitos funcionais implementados.

Arquitetura do Sistema

A arquitetura do sistema usa o princípio “request-reply” entre os clientes e o servidor. Quando um cliente faz um pedido, é enviada uma mensagem contendo a query que o servidor irá enviar à base de dados. A base de dados executa a query e retorna o resultado da mesma para o servidor. O caminho da resposta faz-se da mesma forma só que no sentido diferente. A ligação entre o servidor e o “mySQL” usa a biblioteca “mysql-connector-java-5.1.47-bin.jar”. As ligações entre os clientes e o servidor são via TCP.



Diagramas Conceptual ou ER (Entidade Relacionamento) e Físico

Na realização do trabalho pratico optamos por usar a versão dos diagramas entregues na primeira meta. No ponto de vista do grupo de trabalho, o diagrama apresentado anteriormente, apresentava todas as transações necessárias para a base de dados ficar normalizada, e assim ficar em bom funcionamento na execução final.

Diagrama ER

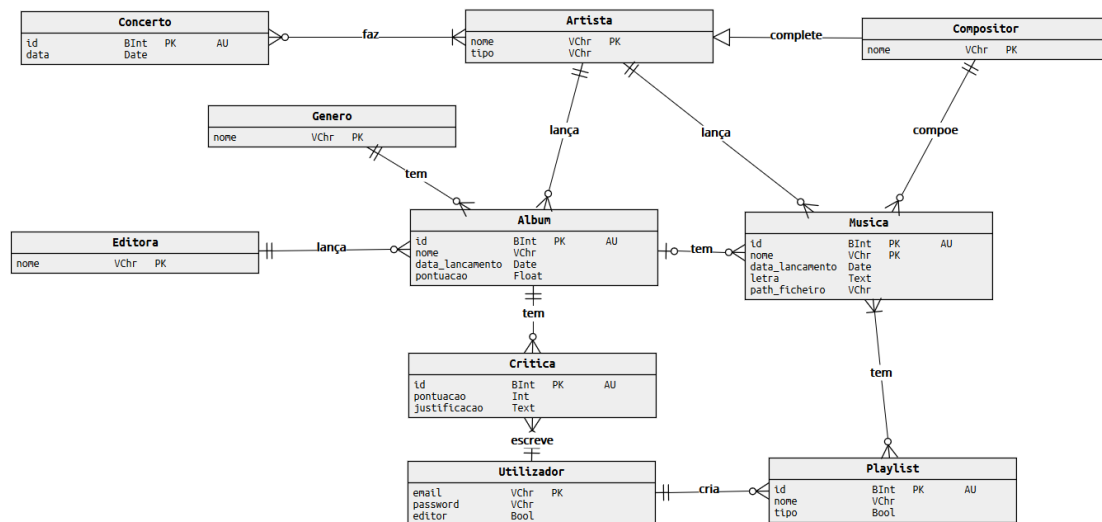
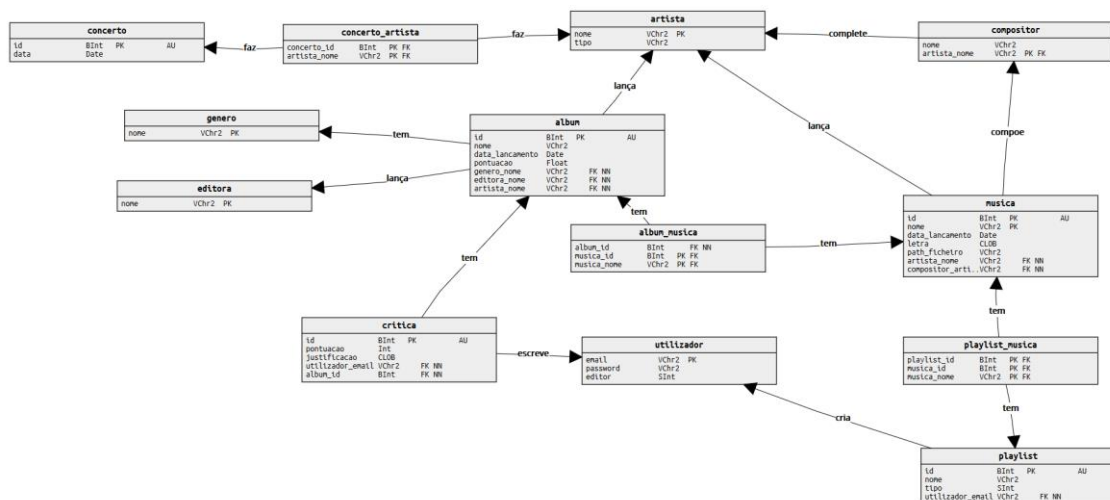


Diagrama Físico



Queries usadas e funcionalidades implementadas

Consultar álbuns por artista

```
"SELECT * FROM root.album where root.album.artista_nome = '" + nomeA+"'";
```

Consultar álbuns por nome

```
SELECT * FROM root.album where root.album.nome = '" + nome+"'";
```

Consultar musicas por nome

```
"SELECT * FROM root.musica where root.musica.nome = '" + nome+"'";
```

pesquisar musica por álbum

```
"SELECT root.album_musica.musica_nome, root.album.nome FROM  
root.album ,root.album_musica  where (root.album_musica.album_id =  
root.album.id) and root.album.nome = '" + nome+"'";
```

pesquisar musica por artista

```
"SELECT * FROM root.musica where root.musica.artista_nome = '" + nome+"'";
```

Pesquisar artistas por nome

```
"SELECT root.artista.nome, "+  
    "root.album.nome , "+  
    "root.album.pontuacao , "+  
    "root.album.genero_nome , "+  
    "root.album.data_lancamento , "+  
    "root.album.editora_nome , "+  
    "root.album_musica.musica_nome "+  
    "from root.artista, root.album, root.album_musica "+  
    "where (root.artista.nome = root.album.artista_nome) and (root.album.id =  
root.album_musica.album_id) and root.album.artista_nome = '"+nome+"'";
```

Pesquisar artista por álbum

```
"SELECT root.artista.nome, "+
    "root.album.nome , "+
    "root.album.pontuacao , "+
    "root.critica.justificacao , "+
    "root.album.genero_nome , "+
    "root.album.data_lancamento ," +
    "root.album.editora_nome ," +
    "root.album_musica.musica_nome "+
    "from root.artista, root.album, root.album_musica, root.critica "+
    "where (root.artista.nome = root.album.artista_nome) and (root.album.id =
root.album_musica.album_id) and (root.critica.album_id = root.album.id) and
root.album.nome = '" + nome+'";
```

Pesquisar artista por musica

```
"SELECT root.artista.nome, "+
    "root.album.nome , "+
    "root.album.pontuacao , "+
    "root.album.genero_nome , "+
    "root.album.data_lancamento ," +
    "root.album.editora_nome ," +
    "root.album_musica.musica_nome "+
    "from root.artista, root.album, root.album_musica "+
    "where (root.artista.nome = root.album.artista_nome) and (root.album.id =
root.album_musica.album_id) and root.album_musica.musica_nome ='" +
nome+'";
```

Pesquisar concerto por artista

```
SELECT root.concerto_artista.artista_nome, "+
    "root.concerto.data "+
    "FROM root.concerto_artista, root.concerto "+
    "WHERE (root.concerto_artista.concerto_id = root.concerto.id) and
(root.concerto_artista.artista_nome = '" + nome+'");
```


Pesquisar criticas por álbum

```
SELECT root.artista.nome , "+
      "root.album.nome , " +
      "root.album.genero_nome , " +
      "root.album.data_lancamento , " +
      "root.album.editora_nome , " +
      "root.album.pontuacao , " +
      "root.critica.justificacao , " +
      "root.critica.utilizador_email "+
"FROM root.artista, root.album, root.critica, root.utilizador "+
"WHERE (root.artista.nome = root.album.artista_nome) and "+
      "(root.critica.utilizador_email = root.utilizador.email) and " +
      "(root.critica.album_id = root.album.id) and " +
      "(root.album.nome = '" + nome+"')";
```

Listar todos os géneros musicais

```
"SELECT * FROM root.genero";
```

Listar todas as editoras

```
"SELECT * FROM root.editora"
```

Listar todos os artistas

```
"SELECT * FROM root.artista;"
```

Listar todos os artistas do tipo “musico”

```
"FROM root.artista " +
"where root.artista.tipo = 'musico'";
```

Listar todas as musicas

```
"SELECT root.musica.nome, " + "root.musica.data_lancamento, " +  
"root.musica.letra, " + "root.musica.artista_nome, " +  
"root.musica.compositor_artista_nome " +  
"FROM root.musica";
```

Listar todos os álbuns

```
"SELECT root.album.nome, "+"root.album.data_lancamento, "+  
"root.album.pontuacao, "+"root.album.genero_nome, "+  
"root.album.editora_nome, "+"root.album.artista_nome " +  
"FROM root.album";
```

Pesquisar o ID de um álbum pelo seu nome

```
"SELECT root.album.id "+  
"FROM root.album " +  
"where root.album.nome = '"+nome+"'";
```

Pesquisar todos os compositores

```
"SELECT root.compositor.nome " +  
"FROM root.compositor";
```

Pesquisar ID de uma musica pelo seu nome

```
"SELECT root.musica.id " +  
"FROM root.musica " +  
"where root.musica.nome = '"+nome+"'";
```

Pesquisar ID de concerto por “data”

```
"SELECT root.concerto.id " +  
"FROM root.concerto " +  
"WHERE root.concerto.data = '"+data+"'";
```

Pesquisar um concerto de um artista pelo nome

```
"SELECT root.concerto_artista.artista_nome, "+"root.concerto.data "+  
"FROM root.concerto_artista, root.concerto "+  
"WHERE (root.concerto_artista.concerto_id = root.concerto.id) and  
(root.concerto_artista.artista_nome = "" + nome+"");
```

Obter media da pontuação de um álbum

```
"SELECT ROUND(AVG(root.critica.pontuacao),2)" +  
"FROM root.critica "+  
"WHERE root.critica.album_id = ""+idAlbum+"";
```

Obter ID de uma playlist pelo nome

```
"SELECT root.playlist.id " +  
"FROM root.playlist " +  
"where root.playlist.nome = ""+nome+"";
```

Listar todas as playlists de um utilizador

```
"SELECT root.playlist.nome,root.playlist.tipo " +  
"FROM root.playlist " +  
"where root.playlist.utilizador_email = ""+util+"";
```

Listar todas as playlists publicas de todos os utilizadores

```
SELECT root.playlist.nome, root.playlist.utilizador_email,  
root.playlist_musica.musica_nome "+  
"FROM root.playlist,root.playlist_musica "+  
"where (root.playlist.tipo = '0') and " +"(root.playlist.id =  
root.playlist_musica.playlist_id)";
```

Obter o id de um concerto por nome de artista e data

```
"SELECT root.concerto_artista.concerto_id " +  
"FROM root.concerto_artista, root.concerto " +  
"WHERE (root.concerto_artista.concerto_id = root.concerto.id) and " +  
"(root.concerto.data = '" + data + "') and " +  
"(root.concerto_artista.artista_nome = '" + artista + "')";
```

Actualizar pontuação de um álbum por id do álbum

```
"UPDATE root.album " +  
"set root.album.pontuacao = '" + pontuacao + "'" +  
"WHERE root.album.id = '" + idAlbum + "'";
```

Tornar uma playlist privada

```
"UPDATE root.playlist " +  
"set root.playlist.tipo = '1' " +  
"where (root.playlist.nome = '" + nomePlaylist + "') and " +  
"(root.playlist.utilizador_email = '" + util + "')";
```

tornar uma playlist publica

```
"UPDATE root.playlist " +  
"set root.playlist.tipo = '0' " +  
"where (root.playlist.nome = '" + nomePlaylist + "') and " +  
"(root.playlist.utilizador_email = '" + util + "')";
```

Listar musicas de uma playlist de um utilizador

```
"SELECT root.playlist.nome, root.playlist_musica.musica_nome " +  
"from root.playlist,root.playlist_musica " +  
"where root.playlist.id = root.playlist_musica.playlist_id and  
(root.playlist.utilizador_email = '" + email + "') and (root.playlist.nome = '" +  
nomePlaylist + "'");
```

Inserir critica

```
"INSERT into
root.critica(root.critica.id,root.critica.pontuacao,root.critica.justificacao,root.critic
a.utilizador_email,root.critica.album_id) "+
"values(null,"+pontuacao+"","+justificacao+"","+util+"","+respostaAlbum+"");
```

Inserir concerto de um artista

```
"INSERT into
root.concerto_artista(root.concerto_artista.concerto_id,root.concerto_artista.arti
sta_nome) "+
"values("+idConcerto+"","+artista+"");
```

Criar um concerto

```
"INSERT into root.concerto(root.concerto.id,root.concerto.data)
values(null,"+data+"");
```

Associar musicas a álbuns

```
"INSERT into root.album_musica(root.album_musica.album_id,
root.album_musica.musica_id, root.album_musica.musica_nome) " +
"values("+respostaAlbum+"","+respostaIDMusica+"","+nomeMusica.trim()+");
```

Criar um álbum

```
"INSERT into
root.musica(root.musica.id,root.musica.nome,root.musica.data_lancamento,root
.musica.letra,root.musica.path_ficheiro,root.musica.artista_nome,root.musica.co
mpositor_artista_nome) " +
"values(null,"+nomeMusica+"","+dataLancamento+"","+letra+"","+caminho+"","
+artista+"","+compositor+"");
```

Inserir compositores

```
"INSERT INTO root.compositor(root.compositor.nome,
root.compositor.artista_nome) "+
  "values('" + nomeCompositor + "','" + nomeArtista+"");"
```

Inserir artistas

```
"INSERT INTO root.artista(root.artista.nome,root.artista.tipo) "+
  "values('" +nomeArtista + "','" +tipo+"");"
```

Inserir géneros musicais

```
"INSERT into root.genero(root.genero.nome) values('"+nomeGenero+"");"
```

Criar uma playlist

```
"INSERT into root.playlist(root.playlist.id, root.playlist.nome,
root.playlist.tipo,root.playlist.utilizador_email) "
"values(null,""+nomePlaylist+"",""+tipoPlaylist+"",""+util+"");"
```

Inserir musicas em playlists

```
INSERT into root.playlist_musica(root.playlist_musica.playlist_id,
root.playlist_musica.musica_id,root.playlist_musica.musica_nome) +
"values('"+respostaIdPlaylist+"','"+respostaIdmusica+"','"+musicalInserir+"");"
```

Mudar data de um concerto de um artista

```
"UPDATE root.concerto_artista,root.concerto " +
" set root.concerto.data = '"+ novaData+ "'" +
" where (root.concerto.id = '" + respostaID + "') and " +
" (root.concerto_artista.concerto_id = root.concerto.id) and " +
" (root.concerto_artista.artista_nome = '"+artista + "');"
```

Mudar letra de uma musica pelo id da musica

```
"UPDATE root.musica " +  
"set root.musica.letra = '" +novaLetra+ "'" +  
"where root.musica.id = '"+ respostaldMusica +"'";
```

Mudar o caminho(path) de uma musica pelo seu ID

```
UPDATE root.musica " +  
"set root.musica.path_ficheiro = '" +novoPath+ "'" +  
"where root.musica.id = '"+ respostaldMusica +"'";
```

Mudar a password de um utilizador

```
"UPDATE root.utilizador " +  
"set root.utilizador.password = '" +novaPassword+ "'" +  
"where root.utilizador.email = '" +email+ "'";
```

Testes de usabilidade

Teste	resultado
Registrar novo utilizador	Passou
Login protegido com password	Passou
Mudar password	Passou
inserir artistas	Passou
Inserir álbuns	Passou
Inserir músicas	Passou
Inserir géneros musicais	Passou
Inserir concertos	Passou
Inserir editoras	Passou
Escrever crítica sobre um álbum (com pontuação)	Passou
Consultar Álbum	Passou
Consultar Artista	Passou
Consultar Musica	Passou
Consultar concerto	Passou
Consultar editora	Passou
Consultar playlists	Passou
Criar playlists	Passou
Editar o path de uma musica	Passou
Editar a letra de uma musica	Passou
Pesquisar critica	Passou
Listar todas as playlists publicas	Passou
Tornar playlist publica	Passou
Tornar playlist privada	Passou
Alterar letra de uma musica	Passou
Alterar path de uma musica	Passou
Alterar data de um concerto	passou

Distribuição de tarefas

Requisitos feitos	Quem fez
Diagrama ER e físico	Pedro e Rui
Registar novo utilizador	Rui
Login protegido com password	Rui
Mudar password	Rui
inserir artistas	Pedro
Inserir álbuns	Pedro
Inserir músicas	Pedro
Inserir géneros musicais	Pedro
Inserir concertos	Pedro
Inserir editoras	Pedro
Escrever crítica sobre um álbum (com pontuação)	Pedro
Consultar Álbum	Rui
Consultar Artista	Rui
Consultar Musica	Rui
Consultar concerto	Rui
Consultar editora	Rui
Consultar playlists	Rui
Criar playlists	Pedro
Editar o path de uma musica	Rui
Editar a letra de uma musica	Rui
Pesquisar critica	Pedro
Listar todas as playlists publicas	Pedro
Tornar playlist publica	Pedro
Tornar playlist privada	Pedro
Alterar letra de uma musica	Pedro
Alterar path de uma musica	Rui
Alterar data de um concerto	Rui
Interface gráfica	Pedro
relatório	Rui

Manual de utilizador

Para executar o servidor, abre-se uma linha de comando escrevendo o seguinte comando:

```
Java -jar databaseServer.jar
```

Executar o cliente em linha de comando:

```
Java -jar DropMusicClient.jar <ip host>
```

Conclusão

Apesar não se ter implementado algumas funcionalidades, o objetivo principal do trabalho foi cumprido, o desenvolvimento de um sistema cliente-servidor com recurso a um SGBD. O contacto com a linguagem SQL, deu nos uma visão geral como é complexo o desenvolvimento de uma base de dados para um caso pratico, visto que tudo depende da organização das tabelas, bem como os dados vão ficar organizados na base de dados.

No nosso trabalho não foram implementadas as funcionalidades de *download/upload* de ficheiros de musica, nem a remoção de dados das tabelas, visto que a sua configuração inicial, a ligação entre o *mysql* e a biblioteca do java, tirou nos demasiado tempo na realização do trabalho. Tirando este obstáculo que tivemos, conseguimos obter a maioria das funcionalidades pedidas, nos requisitos do trabalho pratico. O trabalho em si, faz uma boa gestão dos dados do sistema.