

# Projecto Base de Dados

---

## Relatório Meta 2 (28 Novembro 2018)

---

### Contactos

---

Nome	Email	Número de Estudante
Alexandre Faria	<a href="mailto:afaria@student.dei.uc.pt">afaria@student.dei.uc.pt</a>	2014226180
Carlos Poiares	<a href="mailto:cpoiares@student.dei.uc.pt">cpoiares@student.dei.uc.pt</a>	2014226236

### Table of Contents

---

1. [Contactos](#)
2. [Instalação](#)
3. [Utilização](#)
4. [Estrutura do código](#)
5. [Funcionalidades](#)
  - i. [Registo](#)
  - ii. [Login](#)
  - iii. [Músicas](#)
6. [Diagramas](#)
  - i. [Diagrama Conceptual](#)
  - ii. [Diagrama Físico](#)

### Instalação

---

O projeto foi feito para python 3.

A configuração da ligação com a base de dados está definida em `config.py`

Instruções para correr o projecto. O único módulo externo utilizado é o `psycopg2` para a comunicação com a base de dados PostgreSQL.

```
$ python3 -m venv venv
$ source venv/bin/activate
$ pip install -r requirements.txt
$ python3 -m bdcringe.main
```

## Utilização

---

### Menu de Login / Registo de conta

- 1. Register
- 2. Login
- 3. Leave
- >

### Menu Principal

As opções marcadas com [E] estão restritas a utilizadores com privilégios de editor.

- 0. [E] Inserir editora.
- 1. [E] Tornar editor.
- 2. [E] Inserir Artista.
- 3. Gestão de grupos musicais.
- 4. Gestão de albuns.
- 5. Procurar artista.
- 6. Listar musicas de um artista.
- 7. Listar editoras.
- 8. Sair
- >

### Menu de gestão de grupos musicais

Como um álbum tem que estar necessariamente associado a um grupo musical, a inserção de um novo álbum é feito após pesquisa e seleção de um grupo musical.

- Gestão de grupos
- 0. Listar grupos.
- 1. [E] Inserir novo grupo.
- 2. Procurar grupo.
- 3. Sair
- > 2
- Procurar grupo.
- Nome: Roll
- 0. (2010-01-02 até 2010-10-01) Rolling Stones
- 1. Sair
- > 0
- 0. Listar artistas do grupo.
- 1. [E] Adicionar artista.
- 2. [E] Adicionar álbum.
- 3. [E] Adicionar concerto.
- 4. Listar Concertos.
- 5. Sair.
- >

## Menu de gestão de albuns

Gestão de albums.

0. Listar albums.

1. Procurar album.

2. Sair

> 1

Procurar album.

Nome: Dark

0. Pink Floyd - Dark Side of the Moon

1. Sair

> 0

0. Listar musicas.

1. [E] Adicionar musica.

2. Escrever critica.

3. Listar criticas.

4. Sair.

>

## Estrutura

---

O código da aplicação foi separado em 2 módulos python principais `database` e `menus`.

O módulo `database` contém vários submódulos que servem camada de abstração entre a interface do utilizador e o sql utilizado para as operações com a base de dados.

O módulo `menus` contém os menus para a interface de linha de comandos

Esta estrutura torna a transição para uma interface gráfica mais fácil.

```
|— __init__.py
|— config.py
|— main.py
|— database
|   |— __init__.py
|   |— albums.py
|   |— artists.py
|   |— groups.py
|   |— labels.py
|   |— playlists.py
|   |— songs.py
|   |— user.py
|— menus
|   |— __init__.py
|   |— albums.py
|   |— groups.py
```

## Funcionalidades

---

### Registo

Insere na base de dados um novo utilizador com o username e password fornecidos. Dá erro de inserção caso já exista algum utilizador com o mesmo username devido ao campo `username` servir de chave primária (unique).

```
bdcringe.database.user.register(username, password)
```

```
INSERT
INTO
  utilizador
(username, password)
VALUES
  (%s, %s)
returning
  username, editor
```

### Login

```
bdcringe.database.user.login(username, password)
```

Dado um nome e password de um utilizador, verifica se existe na base de dados.

```
SELECT
  username, editor
FROM
  utilizador
WHERE
  username LIKE %s and
  password LIKE %s
```

## Promover utilizador a editor

Para promover outro utilizador a editor são necessárias duas operações, uma de pesquisa e uma de atualização.

### Procurar utilizador

É feita uma listagem de utilizadores sem permissões de editor com base no nome a procurar para tornar a procura mais fácil para o utilizador.

```
bdcringe.database.user.search(username)
```

```
SELECT
  nome
FROM
  utilizador
WHERE
  username LIKE %s% and
  editor = false
```

### Tornar editor

A partir do username introduzido pelo utilizador altera os seus privilégios de editor.

```
bdcringe.database.user.make_editor(username)
```

```
UPDATE
  utilizador
SET
  editor = true
WHERE
  nome LIKE %s'
```

## Procurar Artista

Dado um nome de um artista é retornada toda a informação relativa ao artista em questão, retorna erro no caso do artista não existir.

```
bdcringe.database.artists.search(nome)
```

```
SELECT
*
FROM
artista
WHERE
nome LIKE '%s'
```

## Listar as músicas de um Artista

Dado um nome de um artista, devolve todas as músicas dele

```
bdcringe.database.artists.get_songs(nome)
```

```
SELECT
nome,
data,
historia
FROM
musica,
musica_artista
WHERE
artista_id = (
SELECT
id
FROM
artista
WHERE
nome LIKE '%s'
) and
musica_id = id
```

## Inserir um Album

Dado um nome, uma data de lançamento válida, um nome de grupo musical e o nome da editora correspondente, insere um novo album. O id da editora é obtido através de uma subconsulta.

```
bdcringe.database.albums.insert(nome, lancamento, grupo_musical, editora)
```

```
INSERT
INTO
  album
  (nome, lancamento, editora_id, grupo_musical_nome)
SELECT
  %s,
  %s,
  id,
  %s
FROM
  editora
WHERE
  nome LIKE %s
```

## Procura de um Album

Dado um nome de um album devolve toda a sua informação.

```
bdcringe.database.albums.search(nome)
```

```
SELECT
  *
FROM
  album
WHERE
  nome LIKE %(like)s ESCAPE '='
```

## Alterar o nome de um Album

Dado um nome original e um novo nome, procura e altera o album.

```
bdcringe.database.albums.update_nome(antigo, novo)
```

```
UPDATE
  album
SET
  nome = %s
WHERE
  nome LIKE %s
```



## Apagar um album

Dado o nome de um album, remove-o da base de dados.

```
bdcringe.database.albums.delete(nome)
```

```
DELETE
FROM
  album
WHERE
  nome LIKE %s
```

## Listar músicas de um album

Dado um nome de um album, devolve todas as músicas a que ele pertencem.

```
bdcringe.database.albums.songs(nome)
```

```
SELECT
  *
FROM
  musica
WHERE
  album_nome LIKE %s
```

## Criticas

---

### Escrever uma crítica a um album

É pedido ao utilizador uma pontuação e uma justificação textual para essa pontuação e é registada na tabela de críticas.

```
bdcringe.database.albums.review(nome, pontuacao, justificacao, username)
```

```
INSERT
INTO
  critica
  (album_nome, utilizador_username, pontuacao, justificacao)
VALUES
  (%s, %s, %s, %s)
```

### Mostrar todas as críticas de um album

Dado um nome de um album, mostra todas as suas críticas.

```
bdcringe.database.albums.reviews(nome)
```

```
select
  pontuacao, justificacao, utilizador_username
from
  critica
where
  album_nome like %s
```

## Inserir Editora

Dado um nome, insere uma nova editora na base de dados.

```
bdcringe.database.labels.insert(nome)
```

```
INSERT  
INTO  
  editora  
  (nome)  
VALUES  
  (%s)
```

## Procurar Editora

Dado um nome devolve toda a informação relativa a essa editora.

```
bdcringe.database.labels.search(nome)
```

```
SELECT  
  *  
FROM  
  editora  
WHERE  
  nome LIKE %s%
```

## Alterar o nome de uma editora

Dado um nome original e um novo nome, procura e altera a editora.

```
bdcringe.database.labels.update(antigo, novo)
```

```
UPDATE  
  editora  
SET  
  nome = %s  
WHERE  
  nome LIKE %s
```

## Remover Editora

Dado o nome de um album, remove-o da base de dados.

```
bdcringe.database.labels.delete(nome)
```

```
DELETE  
FROM  
  editora  
WHERE  
  nome LIKE %s
```

## Listar editoras

Lista todas a editoras existentes.

```
bdcringe.database.labels.get_all()
```

```
SELECT
*
FROM
editora
```

## Inserir Grupo Musical

Dado um nome, data de inicio e data de fim, insere um novo grupo musical.

```
bdcringe.database.groups.insert(nome, inicio, fim)
```

```
INSERT
INTO
grupo_musical
(nome, inicio, fim)
VALUES
(%s, %s, %s)
```

## Adiciona Artista ao Grupo Musical

Dado o id do artista e o nome do grupo musical, insere o artista um novo artista\_grupo\_musical.

```
bdcringe.database.groups.add_artist_id(id_artista, grupo)
```

```
INSERT
INTO
  artista_grupo_musical
  (artista_id, grupo_musical_nome)
SELECT
  %,
  nome
FROM
  grupo_musical
WHERE
  nome like %s
```

## Adiciona Concerto

Dado uma morada do concerto, uma data e o nome do grupo musical em questão, adiciona um novo concerto.

```
bdcringe.database.groups.add_show(address, date, group_name)
```

```
INSERT
INTO
  concerto
  (data, morada, grupo_musical_nome)
VALUES
  (%s,%s,%s)
```

## Listar Concertos

Dado o nome de um grupo musical, lista todos os seus concertos. `bdcringe.database.groups.list_shows(group_name)`

```
SELECT
  data, morada
FROM
  concerto
WHERE
  concerto.grupo_musical_nome like %s
```

## Listar membros de um Grupo Musical

Dado um nome do grupo musical, lista todos os artista presentes no grupo. A pesquisa é feita através do id do grupo\_musical(nome) na tabela de artista\_grupo\_musical, listando toda informação dos artistas associados ao id do grupo\_musical(nome).

```
bdcringe.database.groups.members(nome)
```

```
SELECT
*
FROM
  artista a,
  artista_grupo_musical agm
WHERE
  a.id = agm.artista_id ands
  agm.grupo_musical_nome like %s
```

## Inserir Música

Este processo é feito em dois passos, criando uma entrada na tabela musica e criando posteriormente uma entrada na tabela musica\_artista.

```
bdcringe.database.songs.insert(nome, data, historia, genero, album, id_artist)
```

### Criar Musica e inserir na base de dados

Dado um nome da música, uma data de criação válida, um resumo breve da história, género, e nome do album cria uma nova música.

```
INSERT
INTO
  musica
  (nome, data, historia, genero, album_nome)
VALUES
  ('%s', '%s', '%s', '%s', '%s')
```

### Associar a música a um artista

Dado o nome do artista a quem está associada a música, associa a música ao artista através da tabela musica\_artista. A associação é feita através dos identificadores, o que é devolvido pela subconsulta.

```
INSERT
INTO
    musica_artista
    (artista_id, musica_id)
SELECT
    a.id,
    m.id
FROM
    artista a,
    musica m
WHERE
    m.nome LIKE '%s' and
    a.nome LIKE '%s'
```

### Procurar Música

Dado um nome é retornada toda a informação geral e a lista de artistas de cada música encontrada.

```
bdcringe.database.songs.search_name(nome)
```

```
SELECT
    m.nome,
    m.data,
    m.historia,
    m.genero,
    a.nome,
    a.data_nascimento
FROM
    artista a,
    musica_artista ma,
    musica m
WHERE
    a.id = ma.artista_id
    and m.id = ma.musica_id
    and m.nome LIKE '%s%'
```

### Adicionar uma letra a uma música

Dado o id do compositor (artista), o id da música e a letra, insere uma nova letra na tabela.

```
bdcringe.database.songs.letra(musica, compositor, texto)
```

```
INSERT
INTO
    letra
    (texto, musica_id, compositor_artista_id)
VALUES
    (%s, %s, %s)
```

## Criar uma playlist

Dado um nome da playlist e o username do criador, cria uma nova playlist. `bdcringe.database.playlists.insert(nome, username)`

```
INSERT
INTO
  playlist
(nome, utilizador_username)
VALUES
  (%s, %s)
```

## Editar playlist

A edição da playlist passa por adicionar/remover músicas, listar as músicas ou alterar a sua visibilidade(publica/privada)

### Adicionar música

Dado um id da playlist e o id da música, insere a música na playlist criando uma nova entrada na tabela *playlistmusica\_*  
`bdcringe.database.playlists.add_song(id_playlist, musica_id)`

```
INSERT
INTO
  playlist_musica(playlist_id, musica_id)
SELECT
  %s,
  id
FROM
  musica m
WHERE
  m.id = %s
```

### Remover musica

Dado o id da playlist e o id da música, remove a música da playlist removendo a entrada na tabela *playlistmusica\_*.  
`bdcringe.database.playlists.remove_song(id_playlist, musica_id)`

```
DELETE
FROM
  playlist_musica
WHERE
  musica_id = (
SELECT
  id
FROM
  musica
WHERE
  id = %s
)
and
  playlist_id = %s
```

### Listar músicas de uma playlist

Dado o id da playlist retorna todas as músicas que lhe pertencem. `bdcringe.database.playlists.list_songs(id_playlist)`



```
SELECT
*
FROM
musica m, playlist_musica pm, playlist p
WHERE
pm.musica_id = m.id and
pm.playlist_id = p.id and
p.id = %s;
```

### Alterar visibilidade da playlist

Dado um id da playlist e o booleano de privacidade (True - privada | False - publica), altera a privacidade da playlist.

`bdcringe.database.playlists.set_status(id_playlist, bool)`

```
UPDATE
playlist
SET
privada = %s
WHERE
id = %s
```

### Verificar visibilidade da playlist

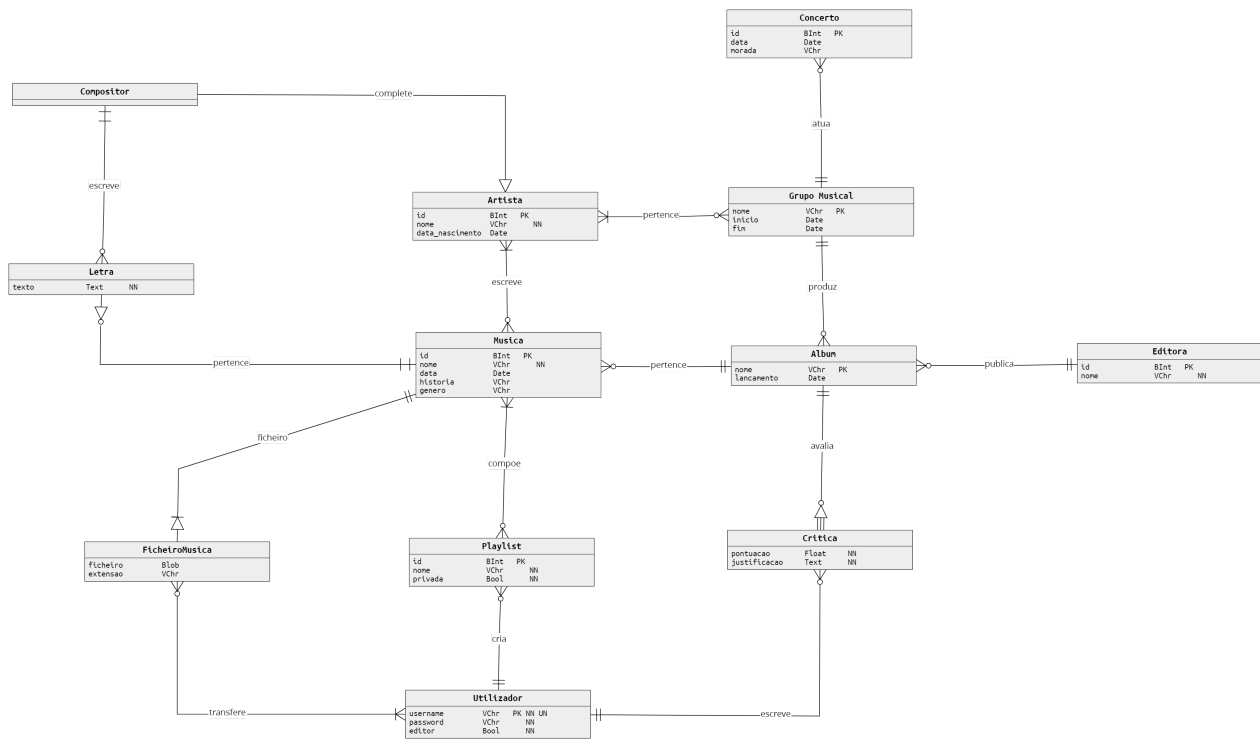
Dado um id da playlist, devolve a privacidade da playlist True - privada | False - publica)

`bdcringe.database.playlists.get_status(id_playlist)`

```
SELECT
privada
FROM
playlist
WHERE
id = %s
```

# Diagramas

## Conceptual



Físico

