



# Relatório

## Projeto

### Base de Dados

Francisco Querido - 2021221158

João Paiva - 2021216669

Salomé Costa - 2021218601

# Índice

Introdução.....	3
Principais Operações .....	3
Registo de Utilizador .....	4
Login Utilizador .....	4
Adicionar Música .....	4
Adicionar Álbum.....	5
Procurar Música.....	5
Listar detalhes de um artista .....	5
Gerar cartões pré-pagos .....	5
Inscrição no Premium.....	6
Criar Playlist.....	6
Meter música a tocar .....	6
Deixar comentário/resposta .....	6
Gerar um relatório mensal.....	7
Top 10.....	7
Manual de Instalação .....	7
Modelo conceptual .....	8
Modelo físico .....	8
Requisitos Tecnológicos .....	9
Divisão de tarefas.....	9

## Introdução

Este projeto tem como objetivo o desenvolvimento de uma plataforma de streaming de música simplificada, recorrendo a uma base dos dados que contém as tabelas com informações de utilizadores, músicas, artistas etc.

## Principais Operações

**Adicionar Utilizador**- fornecendo o nome de utilizador, email, password e tipo(administrador/consumidor/artista);

**Login de um Utilizador**- é efetuado o login usando o email e password;

**Adicionar Música**- identificando o title, duration, release\_date, genre e label\_id;

**Adicionar Álbum**- identificando o title, release\_date, e songs;

**Procurar Música**- especificando o nome da música que deseja procurar;

**Ver detalhes de Arista**- especificando o id do artista à qual deseja visualizar os detalhes;

**Inscrição no Premium**- qualquer consumidor pode tornar-se Premium, fornecendo o(s) número(s) do cartão pré-pago e o período a ser assinado como Premium;

**Criar Playlist**- Para criar uma playlist de um consumidor que esteja conectado é preciso fornecer o nome da mesma, a visibilidade e as musicas que pretende adicionar;

**Meter Música a Tocar**- o consumidor pode meter uma música a tocar fornecendo o song\_id;

**Gerar Cartões Pré-Pagos**- o administrador deve fornecer o valor do cartão (10, 25 ou 50) e o número de cartões a serem criados;

**Comentário/Feedback**- o consumidor pode deixar um comentário a uma música ou responder a comentários já existentes;

**Gerar um Relatório Mensal**- para ser possível listar o número de músicas tocadas por mês nos 12 meses após o mês inserido.

## Registo de Utilizador

- A aplicação Music4you permite o registo de um novo utilizador que é possível através do link: <http://localhost:8080/user> através do método "POST".
- Existem três tipos de utilizador: administrador, consumidor e artista.
- O administrador é o único capaz de registar artistas, e outros administradores. O tipo artista possui informações adicionais ao username, mail e password, como por exemplo, nome artístico, bio, o id do administrador que o está a criar e o id da label associada ao artista.

→ Se o utilizador foi criado com sucesso a response irá ser: {'status': StatusCode, 'results': user\_id}.

→ Em caso de erro: {'status': StatusCode, 'errors': errors}.

## Login Utilizador

- Para que um utilizador se autentique é necessário mandar um *request* para o link <http://localhost:8080/login> com o método "PUT". Para a autenticação o utilizador apenas precisa de introduzir o *email* e *password*. Caso o utilizador já exista na base de dados é criado um *token* para este.
- Exemplo de autenticação de um utilizador:

```
{  
  "email": email,  
  "pass": pass  
}
```

→ Se o login correr como o esperado irá aparecer algo como: {'status': StatusCode, 'token': token }.

→ Caso contrário irá ser mostrada uma mensagem de erro na forma de: {'status': StatusCode, 'errors': errors}.

## Adicionar Música

- Para que seja possível adicionar uma nova música é necessário que o artista mande um *request* para o link: <http://localhost:8080/song>, com a funcionalidade: "POST".
- Os dados necessários para adicionar corretamente uma música são:
  - Title
  - duration
  - release\_date
  - genre
  - label\_id

→ Se a operação ocorreu com sucesso irá aparecer: {'status': StatusCode, 'results': song\_id}.

→ Em caso de erro: {'status': StatusCode, 'errors': errors}.

(!): Apenas o artista pode adicionar uma música

## Adicionar Álbum

- Para que seja possível criar um álbum é necessário que o artista faça um *request* para o link: <http://localhost:8080/album>, com a funcionalidade: "POST".
- Os artistas podem publicar seus álbuns na plataforma.
- Os dados necessários para adicionar um álbum são:
  - title
  - release\_date
- É necessário a adição individual de músicas, com os seus respectivos parâmetros.

→ Se a operação ocorreu com sucesso irá aparecer: {'status': StatusCode, 'results': subscription\_id}.

→ Em caso de erro: {'status': StatusCode, 'errors': errors}.

(!): Apenas o artista pode criar um álbum!

## Procurar Música

- Para que seja possível procurar uma música mande um *request* para o link: <http://localhost:8080/song/{keyword}>, com a funcionalidade: "GET".
- Recupera todas as músicas que contêm a palavra-chave fornecida pelo usuário.

→ Se a operação ocorreu com sucesso irá aparecer: {'status': StatusCode, 'results': {"title": "song\_title", "artists": [{"artist\_name1", "artist\_name2", (...)], "albums": [{"album\_id1", "album\_id2", (...)]}, (...)}.

→ Em caso de erro: {'status': StatusCode, 'errors': errors}.

## Listar detalhes de um artista

- Para que seja possível consultar as informações acerca de um artista (nome, músicas, álbuns, playlists...) é necessário dar *request* em: [http://localhost:8080/artist\\_info/{artist\\_id}](http://localhost:8080/artist_info/{artist_id}) com o método "GET".

→ Se a operação ocorreu com sucesso irá aparecer: {'status': StatusCode, 'results': {"name": "artist\_name", "músicas": [{"song\_id1", "song\_id2", (...)], "álbuns": [{"album\_id1", "album\_id2", (...)], "playlists": [{"playlist\_id1", "playlist\_id2", (...)]}}.

→ Em caso de erro: {'status': StatusCode, 'errors': errors}.

## Gerar cartões pré-pagos

- Para que seja possível gerar um cartão pré-pago, é necessário mandar um *request* para o link: <http://localhost:8080/card>, com a funcionalidade "POST".
- O administrador pode gerar os cartões de um dos três valores possíveis: 10, 25 ou 50 euros.
- Também deve ser fornecido o número de cartões a serem criados.
- As informações de qual administrador criou cada cartão são armazenadas, bem como a data limite do cartão.

→ Se a operação ocorreu com sucesso irá aparecer: {'status': StatusCode, 'results': [id\_card1, (...)]}.

→ Em caso de erro: {'status': StatusCode, 'errors': errors}.

(!): Apenas o administrador pode gerar os cartões pré-pagos!

## Inscrição no Premium

- Para que seja possível um consumidor se tornar Premium é necessário dar *request* em: <http://localhost:8080/subscription> com o método “POST”.
- O consumidor fornece o(s) número(s) do cartão pré-pago e o período a ser assinado como premium. Se o consumidor tiver uma assinatura ativa, o novo período de assinatura é colocado no final da assinatura atual. A informação de quanto foi pago em cada cartão para cada período de assinatura é armazenada.

→ Se a operação ocorreu com sucesso irá aparecer: {'status': StatusCode, 'results': subscription\_id}.

→ Em caso de erro: {'status': StatusCode, 'errors': errors}.

## Criar Playlist

- Para que seja possível criar uma playlist, é necessário mandar um *request* para o link: <http://localhost:8080/playlist>, com a funcionalidade “POST”.
- Os criadores da playlist devem inserir o seu estado de visibilidade (pública ou privada), o nome e os id's das músicas.

→ Se a operação ocorreu com sucesso irá aparecer: {'status': StatusCode, 'results': playlist\_id}.

→ Em caso de erro: {'status': StatusCode, 'errors': errors}.

(!): Somente consumidores Premium podem criar playlists!

## Meter música a tocar

- Para que seja possível meter uma música a tocar, é necessário mandar um *request* para o link: [http://localhost:8080/play\\_song/{song\\_id}](http://localhost:8080/play_song/{song_id}), com a funcionalidade “PUT”.
- Os consumidores podem “reproduzir” uma música (nenhum streaming é esperado, apenas suponha que isso iria transmitir/ouvir a música).
- É usado um trigger, executado sempre que esta função é chamada, que aumenta as “visualizações” da música, algo útil para as funções Top10 e relatório mensal.

→ Em caso de erro: {'status': StatusCode, 'errors': errors}.

## Deixar comentário/resposta

- Para que seja possível realizar um/uma comentário/pergunta, o consumidor terá de mandar um *request* para o link: [http://localhost:8080/coments/{product\\_id}](http://localhost:8080/coments/{product_id}), com a funcionalidade “POST”.
- Se for para responder a uma pergunta já existente, o utilizador tem de dar *request* no link: [http://localhost:8080/coments/{product\\_id}/{parent\\_question\\_id}](http://localhost:8080/coments/{product_id}/{parent_question_id}) e com o método “POST”.
- O consumidor pode comentar/questionar algo sobre uma música(**que exista**). Esta questão pode ser respondida por qualquer tipo de utilizador (administrador, consumidor ou artista).

→ Se a operação ocorreu com sucesso irá aparecer: {'status': StatusCode, 'results': comment\_id}.

→ Em caso de erro: {'status': StatusCode, 'errors': errors}.

## Gerar um relatório mensal

- Para que seja possível obter as estatísticas é necessário dar *request* no link: [http://localhost:8080/report/{year\\_month}](http://localhost:8080/report/{year_month}) com a funcionalidade “GET”.
- É possível listar o número de músicas tocadas nos 12 meses após o mês inserido.

→ Se a operação ocorreu com sucesso irá aparecer: `{'status': StatusCode, 'results': [{"month": "month_0", "genre": "genre1", "playbacks": total_songs_played}, {"month": "month_0", "genre": "genre2", "playbacks": total_songs_played}, {"mês": "mês_1", "gênero": "gênero1", "playbacks": total_songs_played}, (...)]}`.

→ Em caso de erro: `{'status': StatusCode, 'errors': errors}`.

## Top 10

- Para que seja possível obter o Top10 músicas é necessário dar request no link: [http://localhost:8080/top\\_10](http://localhost:8080/top_10) com a funcionalidade “GET”.
- São devolvidas as 10 principais músicas do consumidor (em ordem decrescente com base no número de vezes que cada música foi ouvida).
- Usámos um trigger para manter atualizado o número de vezes que cada musica foi ouvida.

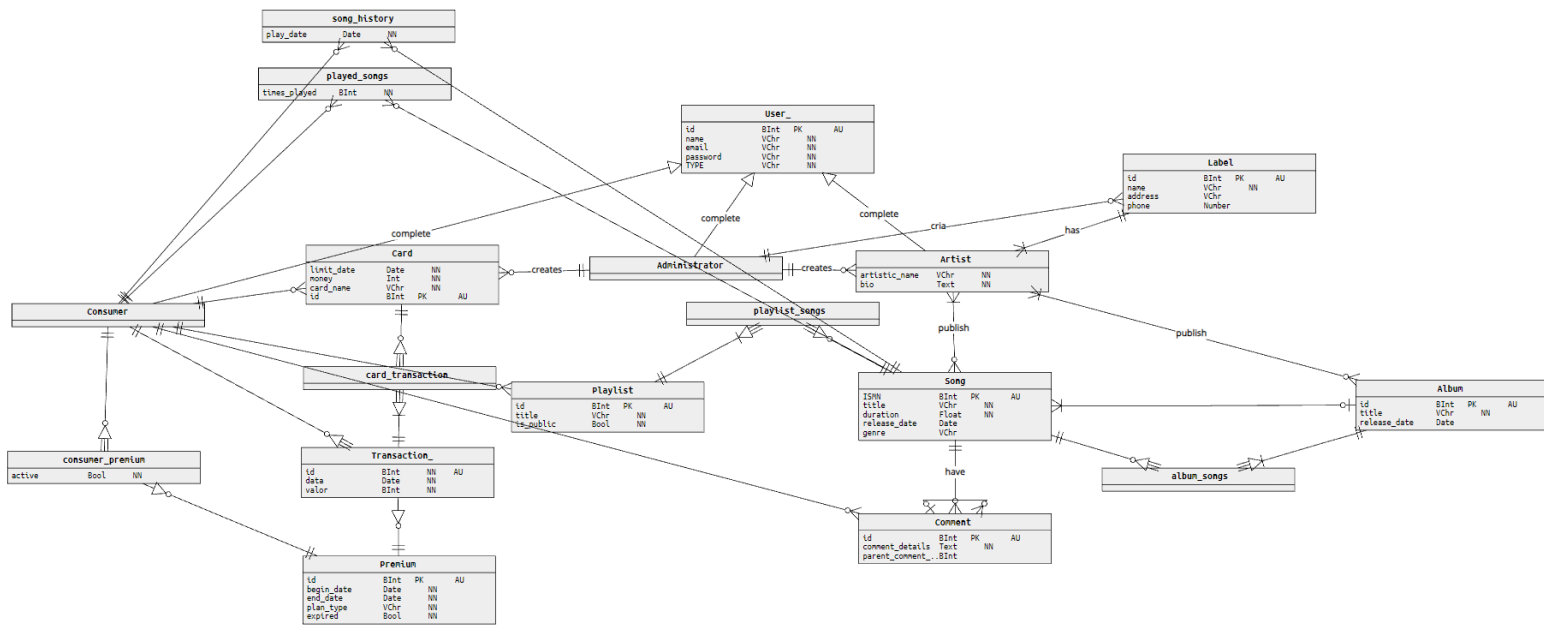
## Manual de Instalação

Para a correta utilização do código, é necessário dar correr os scripts contido nos ficheiros de configuração, Tabelas.sql, Triggers.sql numa query tool da aplicação PgAdmin4.

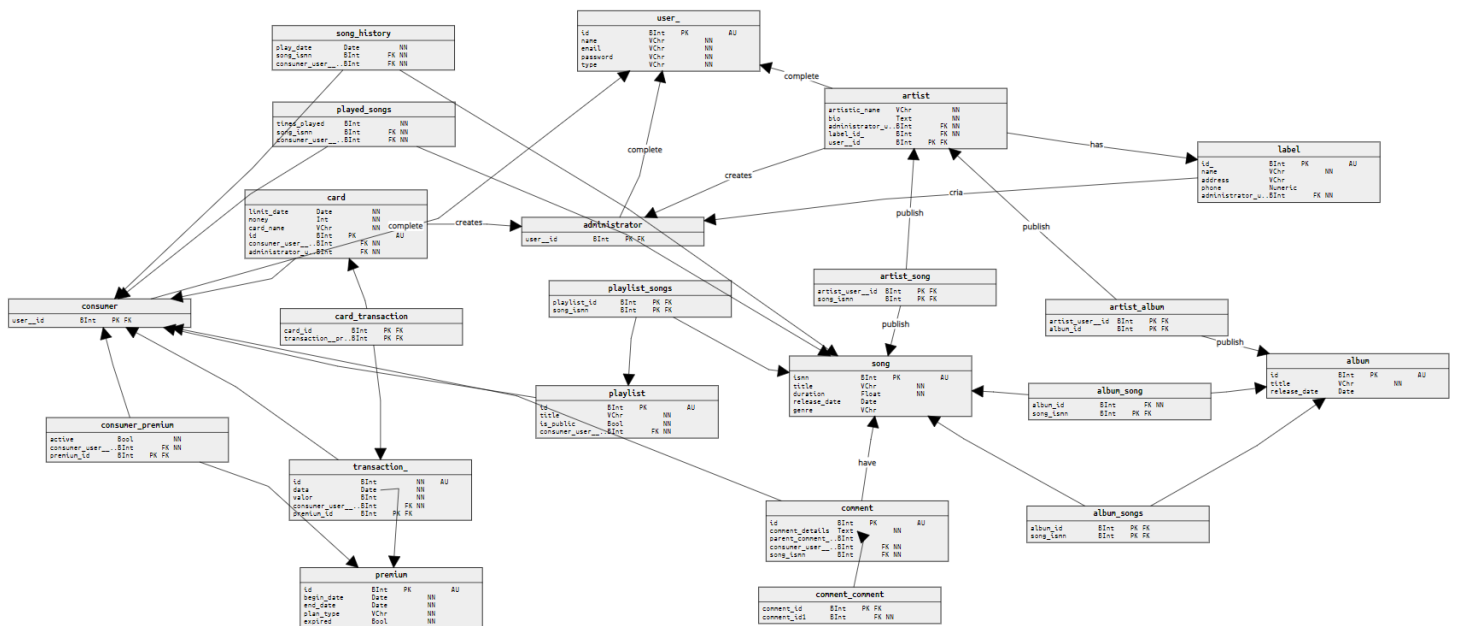
Posteriormente é necessário executar o código do ficheiro Projeto.py para ativar a receção dos requests e acessar à base de dados. Assim, é possível usar o Postman para enviar os requests para a API, e o software está pronto a ser testado.

Os requests têm de ter payloads necessários e cabeçalhos associados à informação guardada no JSON file. O código é todo testado através de localhost, desde que o link do endpoint testado no Postman seja o correto.

## Modelo conceptual



## Modelo físico





## Requisitos Tecnológicos

**Linguagens:** Python SQL

**DBMS:** PostgreSQL

ONDA e Postman

## Divisão de tarefas

Este projeto foi realizado pelos 3 membros do grupo de forma contínua e o mais uniforme possível. Tanto a realização do código como o relatório e o diagrama final foram discutidos entre os 3 de forma a obtermos o melhor resultado.