



Relatório de Projeto

Manual de utilização

Base de Dados

2022/2023

Alunos:

Marco Lucas	2021219146
Rui Coelho	2021235407
Bruno Almeida	2021237081

Introdução

O objetivo do projeto é a administração de um banco de dados de um aplicativo de música. Ele utiliza a estrutura Flask para criar uma API que lida com várias operações relacionadas ao registro do usuário, autenticação, adição de músicas e álbuns, busca de músicas e outras funcionalidades. O código SQL inclui a definição de esquema para várias tabelas que armazenam informações relacionadas a usuários, músicas, álbuns, rótulos, listas de reprodução, comentários, assinaturas e muito mais.

REGISTRO DO USUARIO:

O terminal `/user/` com o método POST é usado para registro do usuário. Quando uma solicitação é feita para esse endpoint, a função `register_user()` é chamada. Ele recupera a carga útil (dados do usuário) da solicitação, estabelece uma conexão com o banco de dados e cria um objeto de cursor para executar instruções SQL.

A função gera um ID de usuário exclusivo e verifica se ele já existe no banco de dados. Se um ID duplicado for encontrado, um novo ID será gerado. As informações do usuário são então inseridas na tabela de usuários usando uma instrução SQL INSERT. Esta informação é armazenada na tabela "PLATFORM_USERS".

LOGIN DO USUARIO:

O terminal `/user/` com o método PUT é usado para autenticação do usuário. A função `login_user()` lida com esse endpoint. Ele recupera a carga útil (dados de login do usuário) da solicitação, estabelece uma conexão com o banco de dados e cria um objeto de cursor.

A função codifica as credenciais do usuário (nome e senha) em um JSON Web Token (JWT) usando a função `jwt.encode()`. O JWT codificado é armazenado na tabela do usuário para fins de autenticação. Esta informação é armazenada na tabela "PLATFORM_USERS".

ADICIONAR MUSICAS E ALBUNS;

O ponto de extremidade `/songs/` com o método POST é usado para adicionar uma música. A função `add_song()` é responsável por lidar com esse endpoint. Ele recupera a carga útil (dados da música)

da solicitação, estabelece uma conexão com o banco de dados e cria um objeto de cursor.

A função gera um ISMN (International Standard Music Number) único para a música e verifica se já existe no banco de dados. Se um ISMN duplicado for encontrado, um novo ISMN será gerado. As informações da música são inseridas na tabela de “SONGS” usando uma instrução SQL INSERT.

Da mesma forma, o ponto de extremidade /album/ com o método POST é usado para adicionar um álbum. A função add_album() lida com esse endpoint. Ele executa operações semelhantes à adição de uma música, mas também valida o JWT fornecido para garantir que o usuário esteja autorizado a adicionar um álbum.

PESQUISAR MUSICA:

O ponto de extremidade /song/<keyword> com o método GET é usado para pesquisar uma música por uma palavra-chave. A função search_song() lida com esse endpoint. Ele recupera a palavra-chave dos parâmetros de URL, estabelece uma conexão com o banco de dados e cria um objeto de cursor.

A função executa uma consulta de pesquisa na tabela “SONGS” com base na palavra-chave usando uma instrução SQL SELECT.

DETALHES DO ARTISTA:

O ponto de extremidade /artists/<id> com o método GET é usado para recuperar detalhes do artista por seu ID. A função get_artist_details() lida com esse endpoint. Ele recupera o ID do artista dos parâmetros de URL, estabelece uma conexão com o banco de dados e cria um objeto de cursor.

A função realiza uma consulta na tabela “ARTISTS” para buscar os detalhes do artista. Ele também verifica a validade do JWT fornecido para garantir que o usuário esteja autorizado a acessar os detalhes do artista.

ASSINAR PREMIUM :

Esta função lida com a solicitação POST para assinatura do serviço premium do aplicativo de música. Ele registra a solicitação e recupera a carga útil (dados JSON) da solicitação. Estabelece uma conexão com o banco de dados e cria um objeto de cursor para executar consultas SQL.

CRIAR PLAYLIST :

Esta função lida com a solicitação POST para criar uma nova lista de reprodução. Recupera a carga útil (dados JSON) da solicitação. Estabelece uma conexão de banco de dados e cria um objeto de cursor.

TOCAR MÚSICA :

Esta função lida com a solicitação PUT para tocar uma música específica identificada por seu ID. Recupera a carga útil (dados JSON) da solicitação. Estabelece uma conexão de banco de dados e cria um objeto de cursor.

GERAR CARTÕES PRÉ-PAGOS:

Esta função trata da solicitação POST para geração de cartões pré-pagos. Recupera a carga útil (dados JSON) da solicitação. Estabelece uma conexão de banco de dados e cria um objeto de cursor.

DEIXE SEU COMENTÁRIO:

Esta função lida com a solicitação POST para deixar um comentário em uma música específica identificada por seu ID. Recupera a carga útil (dados JSON) da solicitação. Estabelece uma conexão de banco de dados e cria um objeto de cursor.

DEIXAR UM FEEDBACK :

Esta função lida com a solicitação POST para deixar feedback sobre o comentário de uma música específica identificada pelo ID da música e pelo ID do comentário principal. Recupera a carga útil (dados JSON) da solicitação. Estabelece uma conexão de banco de dados e cria um objeto de cursor.

ADICIONAR LABEL :

Esta função lida com a solicitação POST para criar uma nova Label. Recupera a carga útil (dados JSON) da solicitação. Estabelece uma conexão de banco de dados e cria um objeto de cursor. Gera um ID exclusivo para Label e o insere junto com o nome da mesma no banco de dados, na tabela "LABELS".

ADICIONAR ARTISTA :

Esta função lida com a solicitação POST para criar um novo artista. Recupera a carga útil (dados JSON) da solicitação. Estabelece uma conexão de banco de dados e cria um objeto de cursor. Gera um ID exclusivo para o artista e insere os detalhes do artista (nome, nome artístico, e-mail, senha, endereço, idade e ID da gravadora) no banco de dados, tá tabela "ARTISTS".

Definições de tabela:

PLATFORM_USERS: esta tabela armazena informações sobre os usuários da plataforma, incluindo seu ID exclusivo, nome de usuário e senha.

ADMINS: Esta tabela representa os administradores do aplicativo de música. Ele tem uma referência de chave estrangeira para a tabela PLATFORM_USERS.

CLIENTES: Esta tabela armazena informações sobre os clientes do aplicativo de música, incluindo seu ID exclusivo, e-mail, endereço e idade. Ele também possui uma referência de chave estrangeira para a tabela PLATFORM_USERS.

SUBSCRIPTION_PREMIUM: Esta tabela contém dados relacionados a assinaturas premium. Inclui detalhes como ID da assinatura, plano, preço, data de início e data de término.

SONGS: Esta tabela armazena informações sobre músicas, incluindo o ISMN (International Standard Music Number), título, duração, data de lançamento, editora e contador de reprodução.

LABELS: Esta tabela representa as gravadoras. Inclui o ID e o nome do rótulo.

ARTISTAS: Esta tabela contém dados sobre artistas. Inclui o ID de usuário da plataforma do artista, nome artístico, ID da gravadora e ID do administrador. Ele possui referências de chave estrangeira para as tabelas PLATFORM_USERS, LABELS e ADMINS.

ALBUMS: Esta tabela armazena informações sobre álbuns, incluindo o ID do álbum, nome, data de lançamento e editora. Ele tem uma referência de chave estrangeira para a tabela ARTISTS.

PRE_PAID_CARDS: Esta tabela representa cartões pré-pagos para assinaturas premium. Inclui o ID do cartão, valor, data de validade e ID do administrador.

COMENTÁRIO: Esta tabela armazena os comentários feitos pelos usuários. Inclui o ID do comentário, ID do usuário da plataforma, ISMN, texto do comentário e data de criação.

PLAYLISTS: Esta tabela representa as playlists criadas pelos usuários. Inclui o ID da lista de reprodução, nome, data de criação e tipo de lista de reprodução.

SONG_COLLECTION: Esta tabela armazena a coleção de músicas em uma lista de reprodução. Inclui o ID da coleção, posição da música, ISMN e ID da lista de reprodução.

ARTISTS_LABELS: Esta tabela representa a relação entre artistas e gravadoras. Inclui o ID do artista e o ID da gravadora.

COMMENTS_COMMENTS: Esta tabela representa o relacionamento entre os comentários. Ele inclui dois IDs de comentário.

PLAYLISTS_SUBSCRIPTION_PREMIUM: Esta tabela representa a relação entre listas de reprodução e assinaturas premium. Inclui o ID da lista de reprodução e o ID da assinatura.

PRE_PAID_CARDS_SUBSCRIPTION_PREMIUM: Esta tabela representa a relação entre cartões pré-pagos e assinaturas premium. Inclui o ID do cartão e o ID da assinatura.

SONGS_ALBUMS: Esta tabela representa a relação entre músicas e álbuns. Inclui o ISMN e o ID do álbum.

ARTISTS_SONGS: Esta tabela representa a relação entre artistas e músicas. Inclui o ID do artista e o ISMN.

População de dados

O código SQL fornecido também inclui instruções INSERT para preencher algumas das tabelas com dados de amostra. Por exemplo, ele insere registros nas tabelas PLATFORM_USERS, ADMINS, LABELS e ARTISTS.

Divisão de tarefas e tempo de trabalho

Decidimos dividir as tarefas da seguinte forma:

- Marco e Rui - rest api;
- Bruno – relatório, diagrama e sql file;
- grupo - pequenas melhorias.

Com um total de cerca de 45 horas de todos os envolvidos.

CONCLUSÃO

O trabalho define o esquema do banco de dados para um aplicativo de música. Inclui tabelas para armazenar informações sobre usuários, músicas, álbuns, rótulos, listas de reprodução, comentários, assinaturas e muito mais. O código pode ser usado como base para implementar a funcionalidade de banco de dados em um aplicativo de música. Outras melhorias e

modificações podem ser feitas para atender a requisitos específicos.