



UNIVERSIDADE FEDERAL DE SERGIPE

DEPARTAMENTO DE COMPUTAÇÃO

BANCO DE DADOS I

TURMA 01

ANDREY FELLIPE ALMEIDA CHAGAS

ARTHUR HENRIQUE SILVA DE LIMA

CRISTIAN DI GIOVANNI

Rotten Caju

Projeto para Disciplina de Banco de Dados - UFS/2022.2

SÃO CRISTÓVÃO

MAIO, 2023

Sumário

Rotten Caju.....	1
Sumário.....	2
Apresentação do Projeto.....	3
Visões de Projeto.....	4
Visão conceitual.....	4
Visão Lógica.....	7
Visão Prática.....	8
API em Python, Flask e SQLAlchemy.....	9
Tecnologias para criação e uso do banco de dados.....	10
Conclusão.....	11
Bibliografia.....	12

Apresentação do Projeto

O projeto é definido como uma aplicação back-end para criação e gerenciamento de um banco de dados Postgres com intuito de gerar um sistema(API) para um possível website que trabalhe com o ambiente de críticas, filmes, séries e prêmios.

Os conceitos abordados no cenário proposto variam entre características técnicas das obras, como produtoras e elenco, até comentários gerados por usuários da plataforma, que, inclusive, podem dar notas às obras cadastradas no banco.

O projeto utiliza as tecnologias Python, SQLAlchemy, Flask, PGAdmin, Postman e serviços AWS para prover o desenvolvimento do projeto, possibilitando a criação do banco de dados, de uma API e de um script capaz de criar requisições como cadastros de usuários, de críticas e afins.

O projeto está disponível em um diretório na plataforma da ferramenta github. O acesso é feito através do seguinte link: <https://github.com/CristianDG/projeto-bd> . Nesse diretório, o projeto guarda os modelos de tabelas, possíveis dependências, configurações adicionais e a própria API.

Dessa forma, o projeto busca integrar informação ao aspecto democrático que as artes permitem, propiciando um ambiente no qual usuários são livres para expressar suas opiniões acerca das obras catalogadas.

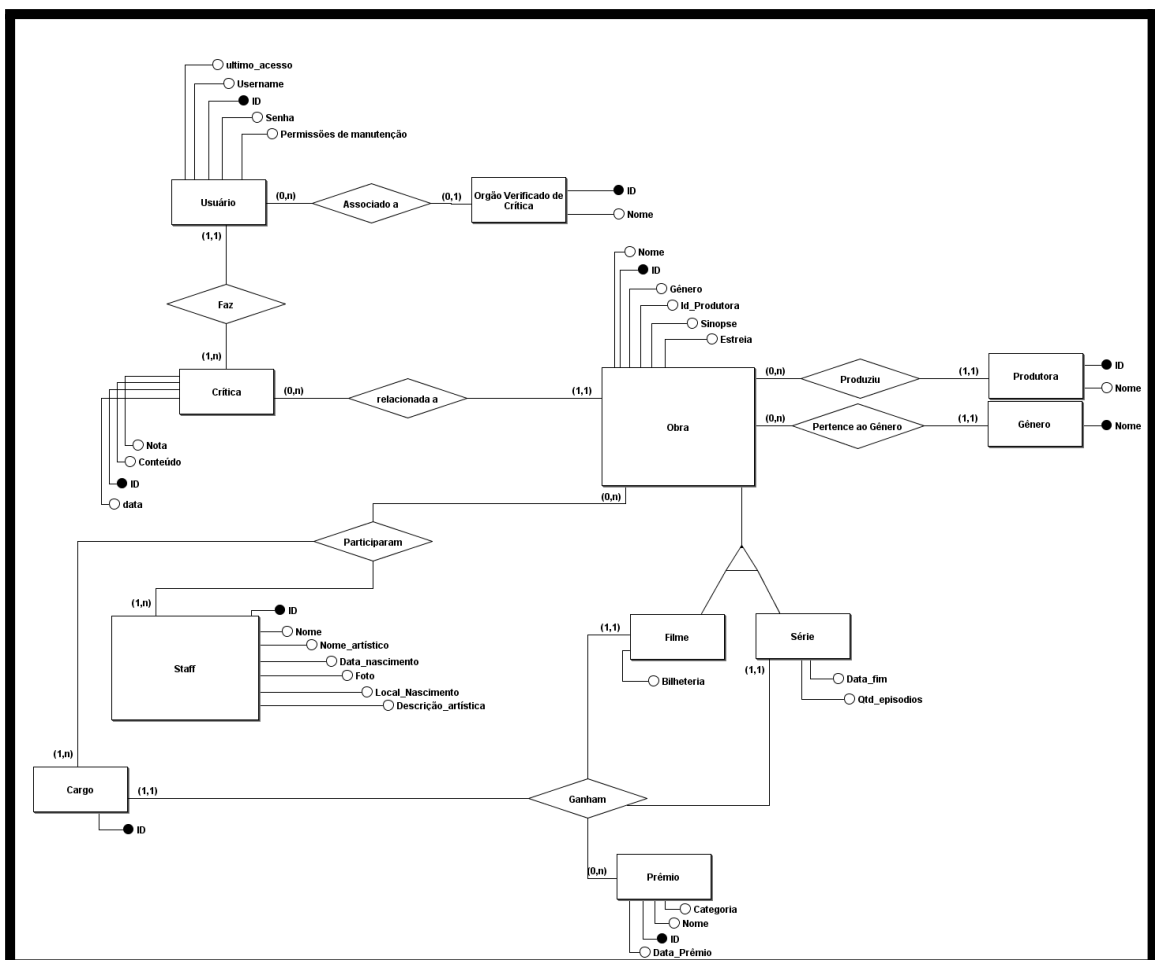
Visões de Projeto

A equipe responsável pelo projeto desenvolveu as etapas para criação do sistema nos seguintes eixos:

- Conceitual: Definir os conceitos básicos do projeto proposto.
- Lógico: Iniciar os aspectos práticos do projeto com a modelagem das tabelas.
- Prático: Criação da API e dos scripts que interferem no banco de dados.

Visão conceitual

A primeira etapa para desenvolvimento do projeto foi o estabelecimento conceitual da ideia. A ferramenta utilizada para criação do modelo conceitual foi o software BrModelo, disponível gratuitamente na web. Com esse software, foi possível fazer uma primeira abordagem do cenário proposto, definindo características básicas do projeto, como as possíveis entidades que ilustram o caso e quais características definem essas entidades.



Dessa forma, o projeto conceitual foi idealizado da seguinte forma:

Para um melhor entendimento do modelo conceitual, é necessário definir formalmente o que as entidades representam.

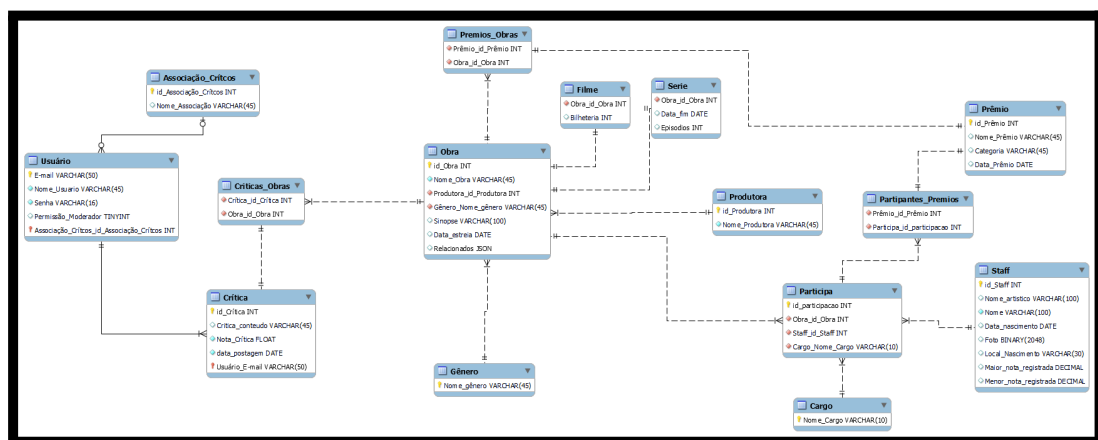
- Usuário: Executor final da aplicação; Autor de críticas; Em certos casos, nos quais o usuário é moderador, é capaz de gerenciar exclusões de dados do banco;
 - Possui os seguintes atributos:
 - Um nome de usuário;
 - Um ID definido pelo e-mail cadastrado;
 - Uma senha para acesso à aplicação;
 - Permissão de moderador
- Órgão Verificado de Crítica: Associação que endossa a crítica de determinado usuário; Não é obrigatório ao usuário ser associado a um Órgão de crítica para poder exercê-la;
 - Possui os seguintes atributos:
 - Um ID;
 - Um nome;
- Crítica: Ato exercido pelo usuário que demonstra sua opinião acerca de determinada obra;
 - Possui os seguintes atributos:
 - Um ID;
 - Um valor que representa a nota dada pelo usuário àquela obra;
 - Um possível texto associado à crítica, uma espécie de comentário sobre a obra;
 - A data de postagem da crítica;
- Obra: Todo o material artístico armazenado no banco de dados; Possui a especificação entre dois tipos básicos: Filmes e Séries;
 - Possui os seguintes atributos:
 - Um nome;
 - Um ID;
 - Um Gênero(Ação, Romance, Drama, Terror...);
 - Uma produtora responsável pela obra;
 - Uma sinopse que sintetize a obra;
 - A data de estreia da obra;
- Filme: Especialização possível para a obra;
 - Possui o seguintes atributo:
 - Bilheteria;
- Série: Especialização possível para a obra;
 - Possui os seguintes atributos:
 - Uma data que representa o último ano de criação ativa da série;
 - A quantidade de episódios;
- Produtora: Empresa responsável pela produção da obra;
 - Possui os seguintes atributos:
 - Um ID;

- Um nome;
- Gênero: Gênero que sintetiza as características nas quais as obras se encaixam;
 - Possui o seguintes atributo;
 - Um nome;
- Staff: Todos os profissionais cadastrados no banco de dados;
 - Possui os seguintes atributos:
 - Um ID;
 - Um nome real;
 - Um nome artístico(caso exista);
 - Data de nascimento;
 - Uma foto para seu perfil;
 - Local de Nascimento
 - Uma descrição do Profissional
- Cargo; Todos os possíveis cargos existentes no banco de dados;
 - Possui o seguinte atributo:
 - Um ID que identifica o cargo;
- Prêmio: Todos os prêmios cadastrados no banco de dados;
 - Possui os seguintes atributos:
 - Um ID;
 - Uma categoria;
 - Um nome;
 - A data do prêmio;

Visão Lógica

Após a conclusão do modelo conceitual, a versão lógica do projeto foi iniciada. O objetivo dessa etapa é tornar factível o modelo de tabelas que o banco de dados irá exercer. Definindo quais serão as possíveis visões do banco e os atributos definitivos das entidades. Tornando transparente as relações de chave estrangeira entre as tabelas.

Logo, o modelo lógico definiu as seguintes diretrizes:



Como a versão lógica aplica em prática as características definidas no modelo conceitual, algumas relações tiveram de ser estabelecidas para organizar as ligações entre as entidades, definindo a ordem dessas ligações nos aspectos possíveis: ligações muitos para muitos, um para um, um para muitos etc .

Sendo assim, surgem tabelas intermediárias como 'Participa' ou 'Criticas_Obras', que relacionam entidades através de suas chaves primárias, ou estrangeiras do ponto de vista da tabela resultante.

Visão Prática

A última visão do projeto aborda o contexto prático, de modo que compreende os códigos escritos, a API implementada e a fase de testagem do projeto. A equipe desenvolveu os códigos em conjunto por meio de pesquisas de documentação e conhecimentos prévios das tecnologias adotadas.

A escrita dos códigos da API se resume em definições de classes e métodos que implementam movimentos básicos de uma API como criação, listagem, exclusão e busca de objetos no banco de dados.

A fase de testagem do projeto abrange duas etapas: a testagem do banco de dados de forma pura e a testagem da API. A testagem pura do banco de dados pode ser feita a partir da ferramenta para gerenciamento de bancos de dados Postgres PGAdmin, possibilitando executar comandos padrão de consulta em bancos de dados relacionais, o SQL. Já a API foi testada executando suas funções e analisando os desdobramentos do banco, verificando os códigos de retorno do requerimento HTTP junto à ferramenta Postman.

API em Python, Flask e SQLAlchemy

A API desenvolvida para o projeto envolve as operações básicas de leitura, escrita, consulta e listagem utilizando as tecnologias: Python, SQLAlchemy e Flask. O Conjunto dessas tecnologias tornou possível desenvolver uma ferramenta de acesso a uma base de dados através de uma interface web.

Além disso, a API implementa um controle de acesso com gerenciamento de login e criação de token para autenticação do usuário simulando o aspecto de segurança que a aceitação de login requer em uma aplicação real.

A API é capaz de lidar com dados em diversos formatos, tais como JSON, XML e HTML, contudo, como o projeto foi desenvolvido em um contexto estrito de back-end, o único tipo de dado adotado é o formato JSON.

A API possui definições de classes representantes das entidades e métodos que interagem com essas classes. Dessa forma, a API propõe uma ferramenta concisa para gerenciamento do banco de dados determinando as ações que o possível website que utilizasse a API usaria.

Tecnologias para criação e uso do banco de dados

A criação do banco de dados seguiu etapas, de modo que o processo fosse planejado e conciso. A primeira etapa se resume à criação do banco de dados na ferramenta da AWS ACADEMY, definindo os parâmetros da instância com base no conteúdo gratuito disponibilizado pela parceria da UFS com a AWS.

O segundo passo foi a criação do esquema que compreende as tabelas projetadas no modelo lógico, a criação do esquema foi automática de acordo com o modelo padrão fornecido pela ferramenta da AWS.

A terceira etapa consiste na aplicação da API na ferramenta Postman para simular os possíveis acessos e usos da API, além da configuração da API em uma máquina EC2 proveniente do serviço AWS, possibilitando o serviço da API em nuvem.

Conclusão

Ao final do trabalho, o projeto é finalizado com o cenário proposto já desenvolvido, a API implementada, testada e aplicada à instância EC2 no serviço AWS.

Dessa forma, o projeto é definido para o usuário como uma aplicação, porém a nível de documentação possui toda a modelagem conceitual, lógica, prática e virtual com uso da tecnologia SWAGGER para promover uma explicação detalhada para bom uso da ferramenta pelo usuário.

Bibliografia

Flask Documentation. Flask. User's Guide. Disponível em:
<<https://flask.palletsprojects.com/en/2.3.x/>> Acesso em: 10/05/2023

Python 3.11.3 Documentation. Python. Documentation. Disponível em:
< <https://docs.python.org/3/> > Acesso em: 10/05/2023

SQLAlchemy 2.0 Documentation. SQLAlchemy. Documentation. Disponível em:
<<https://docs.sqlalchemy.org/en/20/>> Acesso em: 10/05/2023

Swagger Documentation. Swagger. Documentation. Disponível em:
<<https://swagger.io/docs/>> Acesso em: 10/05/2023