

Processo seletivo CAM Tecnologia

Solução web de lista telefônica

Documento descritivo de projeto

Lincoln Abreu Barbosa

Brasília, 14 de Abril de 2021.

Índice

Índice	2
Solução web de lista telefônica	3
Objetivos	3
Descrição de funcionamento	4
Banco de Dados	4

Solução web de lista telefônica

1. Objetivos

Desenvolver uma solução web de lista telefônica. Tal solução precisa oferecer uma interface web simples que permita o CRUD das informações da lista. Requisitos

- A solução precisa ter uma interface Web que permita CRUD dos dados;
- A solução precisa ser implantada em contêineres Docker (versão 19.03.6 ou superior);
- O banco de dados utilizado precisa ser relacional (MariaDB, MySQL ou PostgreSQL);
- O banco de dados e a aplicação não podem estar no mesmo contêiner;
- Toda a implantação precisa ser automatizada via Docker-Compose;
- Todos os artefatos (código, arquivos de configuração, desenho da arquitetura...) desenvolvidos precisam estar disponíveis em um repositório de versionamento de código acessível pelo time de avaliadores;
- A solução Web pode ser desenvolvida em Python ou JavaScript.

Observações

- Os requisitos são requisitos funcionais e não funcionais da solução, mas a criatividade pode ser exercitada. A inclusão de outras funcionalidades é encorajada (autenticação, acesso seguro, submissão de foto são exemplos).
- Comentários no código são sempre bem vindos;
- Google, StackOverflow são seus amigos. Contudo, essa é uma atividade individual. Qualquer evidência de plágio ou cópia de artefatos online será motivo de exclusão do processo seletivo.

2. Descrição de funcionamento

A aplicação pode ser preparada e iniciada utilizando docker-composer, os arquivos dockerfile e docker-compose.yml descrevem as bibliotecas necessárias e os parâmetros de configuração dos contêineres.

O ponto de entrada de execução é o ponto abaixo no arquivo main.py:

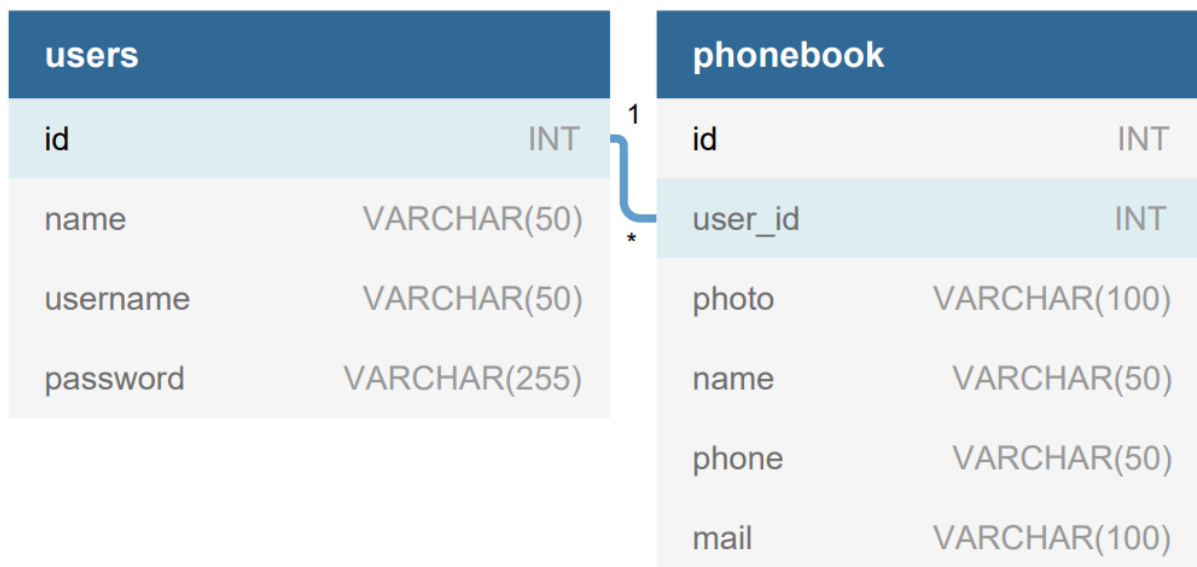
```
if __name__ == "__main__":  
    app.run(port=5000, debug=True, host='0.0.0.0')
```

App é instância de Flask e as rotas são declaradas em main.py, as rotas declaradas são: `/`; `/signup`; `/edit_user`; `/delete_user/`; `/logout`; `/phonebook`; `/add_phone`; `/edit_phone/`; `/update_phone/` e `/delete_phone/`.

Conforme as rotas são requisitadas os arquivos de template são carregados nas requisições `GET` e as requisições `POST` levam a interações de código e com o banco de dados.

3. Banco de Dados

O banco de dados foi modelado conforme o diagrama a seguir:



A relação entre a tabela users e a tabela phonebook é tal que para cada um usuário pode possuir várias entradas em phonebook e a remoção de um usuário implica na remoção de todas as entradas referentes a ele em phonebook.

4. Melhorias a serem realizadas

Entre as várias melhorias a serem realizadas para que a aplicação possa entrar em produção estão:

- Implementação de uma interface de usuário mais agradável e interativa;
- Gerenciamento das entradas do banco de dados sem utilizar auto-incremento;
- Armazenar as senhas em hash e realizar comparações de hash no login.