

Trabalho final de redes 2024/1

Fazer download das bases de dados e descompactar elas

Usar as **libs** de **sqlite3** e **sqlalchemy** para conectar aos arquivos de db

Criar um servidor **multithread** usando a biblioteca **multiprocessing** – **OBRIGATÓRIO** que forneça dados completos recebendo um NOME ou CPFs

O servidor não precisa ter interface, porém deve ter as seguintes buscas

Buscar todos dados por CPF ou NOME – na busca por nome use a primitiva LIKE para procurar nomes parecidos

Buscar por nome na base de CNPJs – para saber se o indivíduo selecionado é sócio de empresas

O servidor deve retornar para o cliente o resultado das buscas no formato JSON.

O cliente deve conter interface e deve ter os seguintes campos

IP

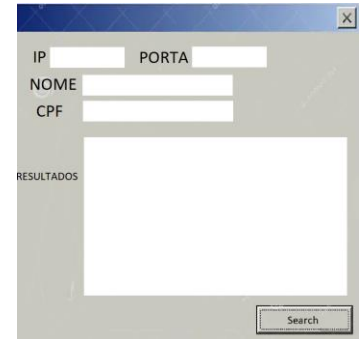
PORTA

NOME

CPF

BOTÃO DE SEARCH

CAIXA DE TEXTO PARA EXIBIR OS RESULTADOS



Interface mínima

O cliente deve receber os dados no formato **JSON** e exibir no formato de **texto padrão** na caixa de texto da interface

Exibir **todos os dados** da pesquisa no cliente

Avaliação

1. SERVER

- Receber os dados de nome do cliente e buscar por nome na basecpf = 5%
- Receber os dados de nome do cliente e buscar por nomes similares basecpf = 5%
- Receber os dados de CPF e buscar por CPF na basecpf = 5%
- Enviar dos dados de resultado no formato JSON para o cliente = 5% - **OBRIGATÓRIO**
- Busca dados de todos CNPJs por nome = 10%
- Busca dados de todos CNPJs por nome e CPF = 20%

2. CLIENTE

- Interface do cliente em pyqt – **OBRIGATÓRIO** = 10%
- Exibir os dados da basecpf = 20%
- Exibir os dados das empresas de qual se é sócio = 20%

EXTRAS

- Fazer um bom **FRONT-END WEB** para o **CLIENTE** usando **VueJS + Flask** – 1,0 na média final
- Criar no servidor uma **API** para fazer consultas por **NOME** e outra para buscar por **CPF** – 0,5 na média final
- Permitir que o usuário selecione 1 dentre os **N NOMBES** encontrados e usar esse NOME SELECIONADO para pesquisar os dados da baseCNPJ – 0,5 na média final