Trabalho de Redes: Desenvolvimento de um Cliente-Servidor Multi-thread

Objetivo:

O objetivo deste trabalho é desenvolver um cliente e um servidor TCP com uso de **multi-thread**. O cliente se conectará ao servidor através de IP e PORTA de comunicação e realizará o envio de dados de **nome** ou **CPF** no formato JSON. O servidor, por sua vez,, aceitará várias conexões (entre 100 e 1000) receberá os dados enviados pelo cliente, criará uma fila de consultas e realizará a consulta por nome ou CPF num banco de dados previamente fornecido. Além disso, o servidor retornará o resultados desses dados no formato JSON para cada cliente conectado.

Requisitos funcionais:

Cliente:

- Deve ter uma GUI (Graphic User Interface) recomendo o uso de PyQt5;
- Deve se conectar ao servidor usando sockets TCP. (IP e PORTA);
- Deve ser capaz de realizar múltiplas consultas simultâneas (use multi-threads);
- Deve enviar Nome ou CPF para o servidor num formato JSON;
- O cliente deve receber os dados do servidor (caso a consulta retorne N respostas, todas devem ser apresentadas na interface do cliente;

Servidor:

- Deve ter GUI (Graphical User Interface) para configurações de interface, porta, pasta do arquivo do db;
- Deve utilizar múltiplas threads **obrigatoriamente**, penalidade de **50%** da nota do trabalho se não funcionar adequadamente; (Certifique-se que está funcionando e utilizando N cores da CPU)
- Deve aceitar várias conexões de clientes (entre 100 e 1000);
- Deve conectar ao banco de dados basecpf.db (https://drive.google.com/file/d/1E3SxFxE20_-ASptS3UIOZvDT1t-yYtYA/view?usp=drive_link), que compactado possui 12.55GB, e realizar as consultas conforme solicitadas pelos clientes;

Observações adicionais:

- O cliente e o servidor devem ser implementados em arquivos separados;
- O servidor pode ser implementado usando um pool de threads para gerenciar as conexões com os clientes;

Testes:

- Testar o cliente e o servidor com diferentes cenários.
- Testar a performance do servidor com várias conexões simultâneas.

Documentação:

Criar um documento que descreva o funcionamento do cliente e do servidor.

O documento deve incluir:

- Estrutura do código;
- Instruções de uso;
- Resultados dos testes.

Recomendações:

Utilize preferencialmente python, gere o executável portável do seu código (Windows e Linux);
Para o uso de múltiplas threads utilize a lib multiprocessing (as outras não são 100% funcionais);
Para utilizar o banco de dados utilize pyodbc, sqlalchemy e sqlite3;
Para o envio de dados padronizados via web-service utilize a biblioteca json;
Para criação de interfaces utilize PyQt5 ou TK, não serão aceitas interfaces WEB;

Critérios de Avaliação:

Funcionalidade do cliente e do servidor. (70%)
Interface gráfica (10%)
Performance do servidor com várias conexões simultâneas. (10%)
Qualidade da documentação apresentada. (10%)

Extras na Avaliação: (1,0 ponto extra na média final)

Adicionar criptografia (SSL entre cliente e servidor) sem necessidade de novas interfaces (0,5 pontos);
Criação de uma segunda interface, agora a versão web que utilize o protocolo https (0,5 pontos).