



UNIVERSIDADE ESTADUAL DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA- CAMPUS I
BACHARELADO EM SISTEMA DE INFORMAÇÃO

LUCAS DOS SANTOS DIAS

COMUNICAÇÃO TCP ENTRE MÁQUINAS NA REDE LOCAL

Salvador

2025

LUCAS DOS SANTOS DIAS

COMUNICAÇÃO TCP ENTRE MÁQUINAS NA REDE LOCAL

Projeto apresentado ao Componente Curricular Linguagem de programação III, do Curso de Bacharelado em Sistemas de Informação do Departamento de Ciências Exatas e da Terra (DCET)- Campus I, da Universidade do Estado da Bahia (UNEB), como requisito avaliativo.

Professor: Jose Grimaldo da Silva Filho

Salvador

2025

1 INTRODUÇÃO

Este trabalho apresenta o desenvolvimento de uma aplicação cliente-servidor em Python utilizando comunicação via sockets TCP. O sistema permite múltiplas conexões simultâneas, autenticação de usuários, bem como envio, recebimento e gerenciamento de arquivos.

2 OBJETIVOS

- Implementar um servidor multithread que aceite conexões simultâneas de múltiplos clientes.
- Garantir a criação e autenticação de contas de usuários.
- Permitir o envio , recebimento, listagem e remoção de arquivos por parte dos clientes.
- Implementar a exclusão total de contas com remoção de dados vinculados.

3 METODOLOGIA

3.1 Estrutura do Sistema

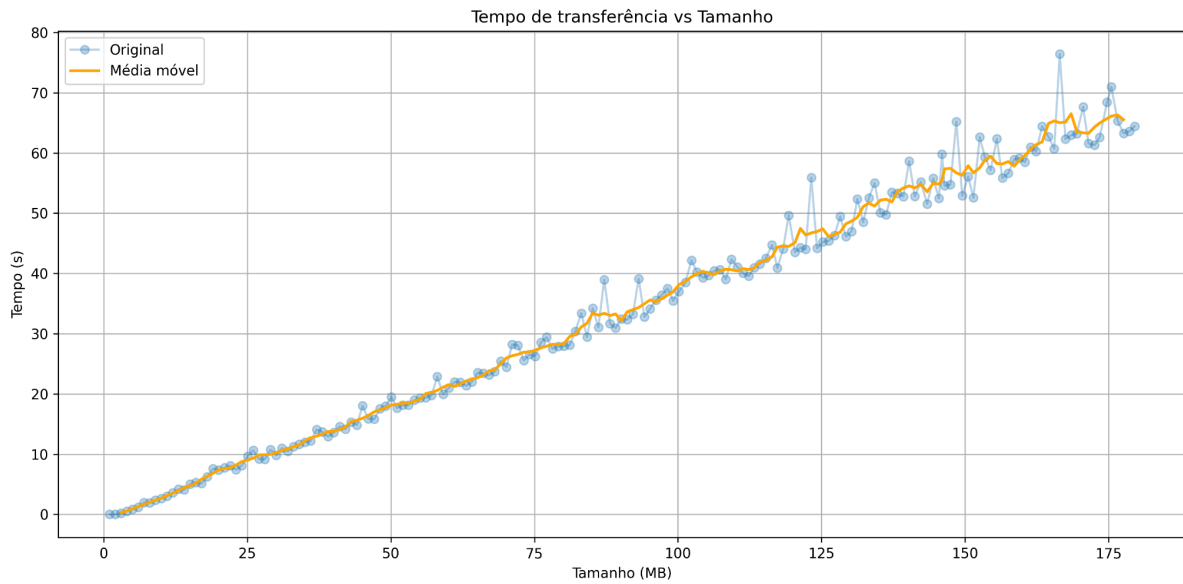
- Servidor (servidor.py): Inicializa um socket TCP, escuta por conexões e trata comandos enviados pelos clientes. Realiza autenticação, cria diretórios para cada usuário e coordena ações via controladores específicos.
- Cliente (cliente.py): Interface CLI que permite login, registro e execução dos comandos suportados: UPLOAD, DOWNLOAD, LIST, DELETE, DELETE_ACCOUNT e QUIT.
- Banco de Dados (db.py): Implementado com SQLite e SQLAlchemy, armazena credenciais de usuários e o identificador do diretório associado.
- Gerenciamento de Arquivos (fileController.py): Responsável por salvar, listar, enviar e remover arquivos no servidor.
- Comunicação (until.py, utils.py): Comunicação baseada em comandos de texto e separadores por linha, com leitura byte a byte em certos pontos para sincronização precisa. Os arquivos são enviados em blocos de até 64 KB.
- O servidor utiliza threads para cada conexão, garantindo suporte simultâneo a vários clientes, Além disso o mesmo cliente consegue se conectar na mesma conta em dispositivos diferentes..

3 Resultados

As seguintes funcionalidades foram implementadas com sucesso:

- Registro e login de usuários com criação automática de diretórios exclusivos.
- Upload e download de arquivos entre cliente e servidor.
- Listagem dos arquivos disponíveis no diretório do usuário.
- Exclusão de arquivos específicos e remoção completa da conta com seus dados.
- Interface clara com mensagens de retorno para cada operação.
- Suporte a múltiplos clientes conectados simultaneamente e realizando ações de forma concorrente.

A aplicação foi testada com arquivos variando entre pequenos (1 MB) e grandes (até 180 MB), simulando cenários reais de uso. O tempo de upload foi registrado para cada caso, e os dados obtidos foram utilizados para gerar o gráfico a seguir:



Observa-se uma tendência linear crescente entre o tamanho do arquivo e o tempo de envio, como esperado. A linha azul representa os dados reais coletados em testes unitários, enquanto a linha laranja representa a média móvel, que suaviza as flutuações pontuais provocadas por sobrecarga de rede ou variações no sistema.

Observações:

- A média móvel confirma um crescimento estável do tempo com o aumento do tamanho.
- Flutuações esparsas em arquivos maiores indicam gargalos pontuais no envio, possivelmente relacionados ao uso da rede local ou ao disco.
- O sistema comportou-se de forma estável até cerca de 170 MB, com pequenos aumentos de latência previsíveis.

