

BigFS

Cliente/servidor em socket
(Entrega 1)

Visão Geral do Sistema de Arquivos

- O sistema é composto por um arquivo `client.py` e um `server.py` que se comunicam por meio de sockets
- O servidor armazena arquivos e os disponibiliza para download, remoção, cópia e listagem, para os clientes que se conectarem a ele.

Arquitetura do Sistema

- **Client.py**

- Conecta-se ao servidor via socket
- Traduz comandos de texto em requisições JSON
- Gerencia diretório local de downloads (“/downloads”)

- **Server.py**

- Escuta na porta 9999 (padrão)
- Gerencia diretório de dados (“/data”)
- Processa comandos dos clientes
- Implementa multithreading para suportar múltiplos clientes simultâneos

Protocolos de Comunicação

- Protocolo: JSON via TCP/IP.

- **Cliente** envia comandos JSON:

- ls - Listar arquivos e diretórios.
- rm - Remover arquivo ou diretório.
- cp - Copiar arquivo ou diretório.
- get - Baixar arquivo ou diretório.

- **Servidor** responde com JSON contendo:

- Status da operação (success ou error).
- Mensagens detalhadas sobre o resultado da operação.
- Exemplo de comunicação:
- Cliente envia:
`{"command": "ls", "path": "/documents"}`
- Servidor responde:
`{"status": "success", "message": "Contents of /documents", "files": [...]}`

Comandos e Respostas

- **Cliente** possui os seguintes comandos:
 - ls [path]: Lista arquivos/diretórios.
 - rm [path]: Remove arquivo/diretório.
 - cp [source] [destination]: Copia arquivo/diretório.
 - get [path]: Baixa arquivo/diretório.
 - help: Exibe ajuda.
 - exit: Encerra a sessão.
- **Servidor** sempre responderá com:
 - sucesso: Retorna status: "success", com dados específicos como lista de arquivos ou confirmação de ação.
 - erro: Retorna status: "error", com a mensagem explicativa.