

# Documentação para o protocolo de comunicação cliente-servidor

Este documento fornece uma visão geral do protocolo de comunicação entre o cliente e o servidor no código Python fornecido. O sistema facilita operações de manipulação de arquivos, como listar arquivos, fazer upload, baixar e excluir arquivos no servidor.

## Server Side (server.py)

### 1. Configuração do Servidor:

- Endereço IP e porta:
  - O servidor está configurado para escutar um endereço IP específico (SERVER\_IP) e uma porta (SERVER\_PORT).
  - Endereço IP: '192.168.56.1'
  - Porta: 12345

### 2. Inicialização do servidor:

- A função `start_server` inicializa um soquete TCP, vincula-o ao IP e à porta especificados e começa a escutar conexões de entrada.
- O servidor usa multithreading para lidar com múltiplas conexões de clientes simultaneamente.

### 3. Tratamento de conexão do cliente:

- Ao aceitar uma conexão, o servidor envia uma mensagem de conexão estabelecida ao cliente.
- Cada conexão do cliente é então tratada em um thread separado pela função `handle_client`.

### 4. Solicitações suportadas:

- O servidor suporta as seguintes solicitações de clientes:
  - LIST: Retorna a lista de arquivos no servidor.
  - DELETE [nome do arquivo]: Deleta o arquivo especificado no servidor.
  - DOWNLOAD [nome do arquivo]: Envia o arquivo especificado ao cliente.

- **UPLOAD [nome do arquivo]:** Recebe um arquivo do cliente e salva no diretório específico (BASE\_DIR).

## **Client Side (client.py)**

### **1. Configuração do cliente:**

- **Endereço IP e porta:**
  - O cliente está configurado para se conectar ao servidor usando o mesmo endereço IP e porta configurados no servidor.

### **2. Estabelecimento de conexão:**

- O cliente estabelece uma conexão com o servidor usando a função `establish_connection`.
- O cliente imprime uma mensagem de conexão estabelecida no terminal, gerada pelo cliente; assim como imprime uma mensagem de confirmação de conexão estabelecida gerada e enviada pelo servidor.

### **3. Comandos suportados:**

- O cliente suporta os seguintes comandos:
  - `list`: Lista os arquivos no servidor.
  - `delete`: Deleta um arquivo do servidor.
  - `download`: Realiza o download de um arquivo do servidor.
  - `upload`: Realiza o upload de um arquivo para o servidor.
  - `exit`: Encerra o cliente.

### **4. Execução de Comandos:**

- A função `run_command` interpreta a entrada do usuário e executa as ações correspondentes.
- Quando o usuário entra com o comando `'list'`, o programa retorna a listagem de arquivos disponíveis no diretório do servidor.

- Quando o usuário entra com o comando 'delete', o programa solicita ao usuário que insira o nome do arquivo que deseja deletar do diretório do servidor.
- Quando o usuário entra com o comando 'download', o programa solicita ao usuário que insira o nome do arquivo que deseja baixar.
- Quando o usuário entra com o comando 'upload', o programa solicita ao usuário que insira o nome do arquivo que deseja fazer upload ao diretório do servidor.

## **Notas Gerais:**

- A constante BUFFER\_SIZE define o tamanho dos blocos de dados transferidos entre o cliente e o servidor durante as operações de arquivo.
- Tanto o cliente quanto o servidor tratam de erros durante operações de arquivo e exibem mensagens apropriadas.
- A comunicação é encerrada normalmente após a entrada do usuário 'exit'.