

# Protocolo de Comunicação Cliente-Servidor

## Visão Geral

A comunicação cliente-servidor implementada permite realizar operações específicas:

- sinalização de que a conexão foi estabelecida com sucesso;
- listagem dos arquivos disponíveis no servidor;
- download de algum arquivo do servidor;
- deletar algum arquivo no servidor;
- upload de algum arquivo para o servidor.

## Estrutura da Mensagem

As mensagens entre o cliente e o servidor seguem um formato JSON para facilitar a leitura e a análise. As mensagens do cliente possuem um par de chave-valor específico para o tipo de requisição e, quando necessário, um par de chave-valor para o nome do arquivo. As respostas do servidor possuem um par de chave-valor referente ao status – indicando se a operação foi concluída com sucesso ou falhou – e um par de chave-valor com uma mensagem que descreve informações adicionais sobre o sucesso ou fracasso da operação.

## Tipos de Mensagens

### 1. Sinalização de que a conexão foi estabelecida com sucesso

Requisição do cliente:

```
{'tipo_requisicao': 'ESTABLISHING_CONNECTION'}
```

Resposta do servidor:

```
{'status': 'success',  
'message': 'Connection established.'}
```

### 2. Listagem dos arquivos disponíveis no servidor

Requisição do cliente:

```
{'tipo_requisicao': 'LIST'}
```

Resposta do servidor:

```
{'status': 'success',  
'files': files}
```

(retorna um dicionário Python contendo a lista de arquivos)

Ou

```
{'status': 'error',  
'message': 'No files found.'}
```

### 3. Download de algum arquivo do servidor

Requisição do cliente:

```
{'tipo_requisicao': 'DOWNLOAD',  
'nome_arquivo': filename}
```

Resposta do servidor:

```
{'status': 'success',  
'message': 'File found. Transference started.'},
```

além de blocos de dados do arquivo especificado, de tamanho especificado pela constante 'BUFFER\_SIZE', lidos no modo de leitura binário ('rb') em um loop até não existirem mais dados a serem lidos do arquivo.

Ou

```
{'status': 'error',  
'message': 'File not found.'}
```

#### **4. Deletar algum arquivo no servidor**

Requisição do cliente:

```
{'tipo_requisicao': 'DELETE',  
'nome_arquivo': filename}
```

Resposta do servidor:

```
{'status': 'success',  
'message': 'File deleted successfully.'}
```

ou

```
{'status': 'error',  
'message': 'File not found.'}
```

#### **5. Upload de algum arquivo para o servidor.**

Requisição do cliente:

```
{'tipo_requisicao': 'UPLOAD',  
'nome_arquivo': filename}
```

Resposta do servidor:

```
{'status': 'success',  
'message': 'Receiving file...'}
```

ou

```
{'status': 'error',  
'message': 'Error during transfer of file.'}
```

### **Conexão**

A comunicação utiliza uma conexão TCP na porta 12345.