



CENTRO UNIVERSITÁRIO DE EXCELÊNCIA
Curso de Sistemas de Informação
Disciplina: Projetos de Redes de Computadores
Professor: Júlio César Andrade Silva

OAT1: Drive de Arquivos com Sockets TCP em Java

O objetivo dessa atividade é criar um sistema simples de gerenciamento de arquivos usando **sockets TCP**, permitindo que o cliente se conecte a um servidor, faça login com um usuário e senha, visualize a lista de arquivos e faça o download ou upload de arquivos. Os arquivos serão armazenados em diferentes pastas, organizadas por **usuário** e por **tipo de arquivo** (PDF, JPG e TXT).

1. Funcionalidades:

1. Servidor:

- O servidor deve ser capaz de **escutar conexões** em uma porta específica.
- Deve **validar o login** do cliente, utilizando usuários e senhas armazenados em um **array simples**.
- Após o login, o servidor deve:
 - **Criar uma pasta para cada usuário**, caso não exista.
 - **Organizar os arquivos em pastas por tipo** (PDF, JPG e TXT) dentro da pasta do usuário.
 - **Listar os arquivos** presentes na pasta do usuário.
 - Permitir que o cliente:
 - **Baixe** um arquivo específico.
 - **Envie** novos arquivos para seu diretório pessoal.
- **Armazenamento organizado:**
 - Cada usuário terá sua própria pasta no servidor.
 - Os arquivos serão classificados por tipo de arquivo (**pdf**, **jpg**, **txt**).

2. Cliente:

- Deve se conectar ao servidor utilizando **sockets TCP**.
- Realizar **login** com um nome de usuário e senha.
- Após o login, o cliente poderá:
 - **Visualizar a lista de arquivos** em suas pastas (PDF, JPG, TXT).
 - **Escolher um arquivo para download**.
 - **Enviar um novo arquivo**, que será salvo na pasta correspondente no servidor.

2. Estrutura de Pastas no Servidor:

armazenamento/

└─ usuario1/

│─ pdf/

│─ jpg/

└─ txt/

└─ usuario2/

│─ pdf/

│─ jpg/

└─ txt/

3. Resumo das funcionalidades principais:

1. Login:

- O cliente envia nome de usuário e senha.
- O servidor verifica as credenciais em um array e autentica o cliente.
- Se o usuário for autenticado pela primeira vez, o servidor cria as pastas necessárias.

2. Listagem de Arquivos:

- O servidor lista os arquivos presentes nas pastas do usuário.
- A lista é enviada ao cliente para visualização.

3. Download de Arquivos:

- O cliente seleciona um arquivo da lista.
- O servidor envia o arquivo binário.
- O cliente salva o arquivo localmente.

4. Upload de Arquivos:

- O cliente seleciona um arquivo local para envio.
- O cliente envia o arquivo binário para o servidor.
- O servidor salva o arquivo na pasta correta (por tipo e usuário).

4. Desafios Adicionais (Opcional):

- Implementar **controle de tamanho máximo de arquivo**.
- Adicionar **validação de tipo de arquivo** para evitar uploads inválidos.
- Implementar um **log de atividade** para registrar operações de upload e download

Barema de Avaliação

Total: 20 pontos

1. Funcionalidades Básicas (6 pontos)

- **Conexão Cliente-Servidor (2 pontos)**
 - **Descrição:** Cliente conecta ao servidor e mantém comunicação.
- **Validação de Login (2 pontos)**
 - **Descrição:** Somente usuários autenticados podem acessar os arquivos.
- **Estrutura de Pastas por Usuário (1 pontos)**
 - **Descrição:** Cria pasta para cada usuário com subpastas (pdf, jpg, txt).
- **Listagem de Arquivos (1 pontos)**
 - **Descrição:** Exibe lista de arquivos organizados por tipo.


2. Funcionalidades de Arquivo (10 pontos)

- **Upload de Arquivos (5 pontos)**
 - **Descrição:** Envia arquivos do cliente para o servidor.
- **Download de Arquivos (5 pontos)**
 - **Descrição:** Baixa arquivos organizados por tipo.

3. Organização e Qualidade do Código (4 pontos)

- **Estrutura e Modularização (2 pontos)**
 - **Descrição:** Código organizado em métodos claros.
- **Comentários e Nomes Significativos (2 pontos)**
 - **Descrição:** Comentários explicativos e nomes claros.

Número de pessoas por grupo: 1 a 2

 O professor poderá fazer perguntas direcionadas a um membro específico da dupla, e respostas incorretas impactarão a nota de ambos.