Especificação de Trabalho – Disciplina de Sistemas Distribuídos

Tema: Desenvolvimento de um Chat Distribuído (Modelo WhatsApp) com Comunicação via Socket

1. Objetivo

O objetivo deste trabalho é aplicar os conceitos de **comunicação em sistemas distribuídos** por meio do desenvolvimento de uma aplicação de chat estilo "WhatsApp", utilizando **sockets** para permitir:

- Troca de mensagens de texto entre dois usuários (chat privado).
- Envio de mensagens para um grupo de usuários.
- Envio e recebimento de arquivos entre dois usuários ou entre grupos.

O projeto deverá explorar aspectos de **concorrência**, **sincronização** e **tratamento de falhas** típicos em sistemas distribuídos.

---

2. Requisitos Funcionais

1. **Conexão Cliente-Servidor**
    - A arquitetura deve seguir o modelo cliente-servidor.
    - O servidor será responsável por gerenciar as conexões dos clientes, identificar usuários e rotear mensagens/arquivos.
    - O cliente deverá permitir ao usuário autenticar-se, iniciar conversas privadas e interagir com grupos.
2. **Mensagens Privadas**
    - Permitir que um usuário envie uma mensagem de texto a outro usuário conectado.
    - O receptor deve receber em tempo real a mensagem, desde que esteja conectado.
3. **Mensagens em Grupo**
    - O sistema deve suportar a criação de grupos de usuários.
    - Mensagens enviadas a um grupo devem ser distribuídas automaticamente a todos os membros.
4. **Envio de Arquivos**

- O sistema deve permitir o envio de arquivos (texto, imagem ou PDF, por exemplo).
- O arquivo enviado deve ser recebido corretamente pelo(s) destinatário(s).

5. **Gerenciamento de Sessão**
    - Usuários devem ser identificados por um nome único (username).
    - O servidor deve registrar quem está conectado e garantir a entrega de mensagens apenas aos destinatários corretos.

---

3. Requisitos Não Funcionais

- **Tecnologia:** Implementação obrigatória com **sockets** (TCP ou UDP, a critério do grupo, com justificativa técnica).
- **Concorrência:** O servidor deve ser **multithreaded** ou usar mecanismos assíncronos para gerenciar múltiplas conexões simultâneas.
- **Persistência (opcional):** O sistema pode armazenar mensagens/arquivos enviados quando um usuário estiver offline, para entrega posterior.
- **Interface:**
    - Não é obrigatório desenvolver interface gráfica. Uma interface de terminal (CLI) já satisfaz os requisitos mínimos.
    - Se o grupo desejar, pode implementar interface gráfica para melhorar a experiência de uso.

---

4. Entregáveis

1. **Código Fonte:** Projeto bem estruturado e comentado, com instruções de execução.
2. **Relatório Técnico (até 10 páginas):**
    - Introdução ao problema.
    - Arquitetura do sistema (cliente-servidor, diagrama de fluxo de mensagens, etc.).
    - Explicação das principais decisões técnicas (uso de TCP ou UDP, tratamento de concorrência, etc.).

- Descrição dos testes realizados.
- Limitações conhecidas e possíveis melhorias.

3. **Demonstração em Sala:**
    - Cada grupo deverá demonstrar o funcionamento do sistema com pelo menos 5 clientes conectados simultaneamente.

---

5. Critérios de Avaliação

- **Corretude (40%)** – Se todas as funcionalidades especificadas foram implementadas corretamente.
- **Qualidade do Código (20%)** – Organização, modularidade, clareza e documentação.
- **Relatório Técnico (20%)** – Clareza na descrição da arquitetura e justificativas técnicas.
- **Demonstração Prática (20%)** – Funcionamento do sistema em ambiente controlado.

---

6. Observações

- O trabalho pode ser realizado em grupos de até **3 alunos**.
- É permitido o uso de qualquer linguagem de programação que ofereça suporte a sockets (C, Java, Python, Go, etc.).
- Bibliotecas de terceiros podem ser utilizadas, desde que não substituam a implementação via socket.
- Prazo de entrega: **22/9/2025**