UNIVERSIDADE DO ESTADO DA BAHIA
Departamento de Ciências Exatas e da Terra, Campus I
Colegiado de Sistemas de Informação / **Semestre:** 2025.2
Disciplina: Linguagem de Programação III / **Professor:** Vagner Fonseca
Aluno: _____

Atividade de Fixação

Implemente um sistema de chat simples em Java, usando sockets e threads. O servidor deve aceitar múltiplos clientes simultaneamente, cada um rodando em sua própria thread. Sempre que um cliente enviar uma mensagem, o servidor deve repassá-la a todos os outros clientes conectados.

Requisitos

1. Servidor:
    a. O servidor deve escutar em uma porta fixa (ex.: 12345).
    b. Para cada cliente conectado, o servidor deve criar uma nova thread dedicada ao tratamento das mensagens.
    c. As mensagens enviadas por um cliente devem ser retransmitidas a todos os outros clientes (broadcast).
    d. Deve gerenciar uma lista de sockets ativos (sincronização necessária).

2. Cliente:
    a. O cliente conecta ao servidor e pode enviar mensagens pelo teclado.
    b. No momento da conexão o cliente deve informar seu nickname.
    c. As mensagens recebidas do servidor devem ser exibidas em tempo real (thread separada para leitura).

3. Concorrência e Sincronização:
    a. Use threads para garantir que múltiplos clientes sejam atendidos ao mesmo tempo.
    b. Garanta acesso thread-safe à lista de clientes conectados (ex.: Collections.synchronizedList, CopyOnWriteArrayList ou synchronized).

Desafio Extra:

- Reimplementar a parte do servidor usando um ExecutorService (Executors.newFixedThreadPool) em vez de criar threads manualmente.
- Adicionar um comando especial: se o cliente digitar exit, ele deve se desconectar do chat sem encerrar o servidor.

O que precisa ser completado (checklist rápido)
Servidor:

- Implementar broadcast corretamente e remoção do cliente ao sair.
- Tratar o comando exit (encerrar só o cliente, não o servidor).
- (Extra) Migrar “new Thread(...).start()” para ExecutorService e explicar por que isso melhora a gestão de recursos do servidor.

Cliente:

- Garantir envio de mensagens do teclado para o servidor.
- Sair de forma limpa ao digitar exit (fecha socket, deixa a thread de leitura terminar).