Nome do Campus: Polo Cruzeiro-SP

Nome do Curso: Desenvolvimento FullStack

Nome da Disciplina: Por que não paralelizar

Número da Turma: 2025.1

Semestre Letivo: Primeiro Semestre

Nome do Aluno: Moniza de Oliveira Silva Santos Pelegrini

Matrícula: 202401190829

1. **Objetivo da Prática**

Desenvolver um sistema cliente-servidor em Java utilizando Sockets, JPA e Threads. A prática visa: - Criar servidores baseados em Socket com múltiplos clientes paralelos.

- Criar clientes síncronos e assíncronos.

- Utilizar Threads para processos concorrentes.

- Integrar com banco de dados SQL Server utilizando JPA.

2. **Desenvolvimento da Solução**

O projeto foi dividido em três partes:

- CadastroServer: servidor Java que utiliza JPA e Threads para aceitar múltiplas conexões.

- CadastroClient: cliente síncrono que se conecta ao servidor, autentica e solicita a listagem de produtos.

- CadastroClientV2: cliente assíncrono com interface gráfica Swing e Thread de leitura.

O servidor utiliza JPA para acessar as entidades Usuario e Produto no banco de dados SQL Server. A autenticação ocorre via login e senha, e a listagem de produtos é retornada com comando 'L'.

3. **Resultados Obtidos**

Ao executar o servidor e os clientes:

- O cliente síncrono foi capaz de autenticar e exibir produtos com sucesso.

- O cliente assíncrono permitiu múltiplas interações via terminal e exibiu mensagens em tempo real na interface.

Todas as funcionalidades especificadas no enunciado foram implementadas com sucesso.

4. **Análise e Conclusão**

1. Como funcionam as classes Socket e ServerSocket?

- ServerSocket escuta conexões em uma porta específica. Socket representa a conexão estabelecida entre cliente e servidor.

2. Qual a importância das portas?

- Portas identificam serviços diferentes em um mesmo IP, permitindo múltiplas aplicações de rede rodando simultaneamente.

3. Para que servem ObjectInputStream e ObjectOutputStream?

- Permitem enviar/receber objetos Java pela rede. Os objetos precisam ser serializáveis para serem transmitidos.

4. Como foi possível garantir o isolamento do banco de dados no cliente?

- As entidades JPA foram copiadas para o cliente, mas sem incluir o persistence.xml ou qualquer conexão JPA, evitando acesso direto ao banco.

5. Como as Threads são usadas para respostas assíncronas?

- A ThreadClient lê continuamente os dados enviados pelo servidor sem bloquear o fluxo principal do cliente.

6. Qual o papel do método invokeLater?

- Garante que atualizações na interface Swing ocorram na thread correta (Event Dispatch Thread), evitando erros de concorrência.

7. Comparação entre clientes síncronos e assíncronos:

- O cliente síncrono aguarda a resposta do servidor antes de continuar.

- O cliente assíncrono processa comandos e respostas simultaneamente, melhorando a experiência e permitindo múltiplas ações em paralelo.