

**Universidade:[Estácio]**

**Campus: [1685 Polo Anhaia Melo - São Paulo - SP]**

**Curso: [Desenvolvimento Full Stack]**

**Disciplina: [Nível2: Vamos manter as informações? ]**

**Turma: [9001]**

**Semestre Letivo: [2024.3]**

**Integrantes da Prática: [Fernando Garcia Farias]**

Relatório discente de acompanhamento

**Criando o Servidor e Cliente de Teste**

1. Objetivo da Prática;

O objetivo é desenvolver uma aplicação cliente-servidor utilizando Java, onde o cliente envia objetos para o servidor através de sockets. O servidor persiste esses objetos em um banco de dados utilizando JPA.

2.Todos os códigos solicitados neste roteiro de aula;

Foram implementados códigos para o servidor e o cliente. O servidor utiliza ServerSocket para esperar por conexões, Socket para comunicação com o cliente, e JPA para persistência dos dados. O cliente utiliza Socket para se conectar ao servidor e enviar objetos serializados.

3.Os resultados da execução dos códigos também devem ser apresentados;

Os resultados mostram que os objetos enviados pelo cliente são recebidos e persistidos corretamente pelo servidor no banco de dados. As operações de leitura e escrita entre cliente e servidor funcionam conforme esperado, utilizando fluxos de entrada e saída de objetos.

4.Análise e Conclusão:

 As classes Socket e ServerSocket permitem a comunicação entre cliente e servidor através de uma conexão TCP/IP. ServerSocket aguarda por conexões enquanto Socket representa a conexão estabelecida.

 Portas são cruciais para identificar serviços específicos em um servidor, permitindo múltiplas conexões simultâneas de diferentes clientes de forma organizada.

 As classes ObjectInputStream e ObjectOutputStream permitem a leitura e escrita de objetos Java através de fluxos de entrada e saída. Os objetos devem ser serializáveis para serem convertidos em bytes e transmitidos pela rede.

 Mesmo utilizando JPA no cliente, o isolamento do acesso ao banco de dados é garantido porque todas as operações de persistência são centralizadas no servidor. O cliente envia apenas dados serializados para o servidor, que então manipula esses dados e interage com o banco de dados de forma segura e controlada.

https://github.com/Futureforworld/M5