

Missão Prática | Nível 4

Mundo 3

Objetivos da Prática:

1. Criar servidores Java com base em Sockets;
2. Criar clientes síncronos para servidores com base em Sockets;
3. Criar clientes assíncronos para servidores com base em Sockets;
4. Utilizar Threads para implementação de processos paralelos;
5. No final do exercício, o aluno terá criado um servidor Java baseado em Socket, com acesso ao banco de dados via JPA, além de utilizar os recursos nativos do Java para implementação de clientes síncronos e assíncronos. As Threads serão usadas tanto no servidor, para viabilizar múltiplos clientes paralelos, quanto no cliente, para implementar a resposta assíncrona;

Análise e Conclusão:

Como funcionam as classes Socket e ServerSocket?

O uso das duas juntas permitem uma comunicação entre dispositivos, através do uso do protocolo TCP/IP. A ServerSocket atua como o lado servidor, e a Socket, como o lado cliente.

Qual a importância das portas para a conexão com servidores?

Elas permitem o acesso à múltiplas aplicações/serviços de forma simultânea. Os números diferentes das portas permitem a diferenciação, portanto, melhor gerenciamento, entre essas aplicações (além de sua identificação).

Para que servem as classes de entrada e saída ObjectOutputStream e ObjectOutputStream, e por que os objetos transmitidos devem ser serializáveis?

Elas servem para a serialização e desserialização de objetos, ou seja, torná-los “armazenáveis” de forma que seja possível guardá-los e depois recuperá-los. É importante que o objeto seja serializável (através da implementação da interface Serializable) para que a máquina reconheça que esse objeto é capaz de ser convertido em uma sequência de bytes (guardado) e desconvertido (recuperado).

Por que, mesmo utilizando as classes de entidades JPA no cliente, foi possível garantir o isolamento do acesso ao banco de dados?

Através do modelo de camadas, que permite a divisão de tarefas da aplicação, deixando o acesso ao banco de dados isolado do cliente.

Como as Threads podem ser utilizadas para o tratamento assíncrono das respostas enviadas pelo servidor?

Elas permitem a continuação da comunicação entre o servidor e o cliente, ou seja, mesmo que a aplicação esteja esperando uma resposta do servidor, ela pode executar outras tarefas.

Para que serve o método invokeLater, da classe SwingUtilities?

Para que a manipulação de um componente Swing só aconteça em uma EDT, uma thread sequencial que garante a segurança do processamento de seus dados.

Como os objetos são enviados e recebidos pelo Socket Java?

Através das classes ObjectOutputStream e ObjectInputStream.

Compare a utilização de comportamento assíncrono ou síncrono nos clientes com Socket Java, ressaltando as características relacionadas ao bloqueio do processamento.

No tratamento síncrono, há o bloqueio ao processador (ou seja, da capacidade de fazer outras tarefas) enquanto não certa resposta do servidor não é recebida). No tratamento assíncrono, isso não ocorre; o acesso ao processador e a capacidade de realizar outras tarefas não é bloqueado.