



UNIVERSIDADE ESTÁCIO DE SÁ  
FULLSTACK

## **Mundo 03 - Nível 05**

**Por que não paralelizar**

**Servidores e clientes baseados em Socket,  
com uso de Threads tanto no lado cliente  
quanto no lado servidor,  
acessando o banco de dados via JPA.**

Herval Rosano Dantas  
Matrícula 202205119203

RIO DE JANEIRO – RJ  
2023

## **Objetivo da Prática**

- Criar servidores Java com base em Sockets.
- Criar clientes síncronos para servidores com base em Sockets.
- Utilizar Threads para implementação de processos paralelos.

Na conclusão deste trabalho, terá sido 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.

## **Desenvolvimento da prática**

- Criar o Servidor e Cliente de Teste.
- Criar a camada de persistência em CadastroServer.
- Criar a camada de controle em CadastroServer.
- No pacote principal, cadastroserver, adicionar a Thread de comunicação, com o nome CadastroThread.
- Implementar a classe de execução (main), utilizando as características que são apresentadas a seguir.
- Criar o cliente de teste, utilizando o nome CadastroClient, do tipo console, no modelo Ant padrão, para implementar a funcionalidade apresentada a seguir.
- Configurar o projeto do cliente para uso das entidades.

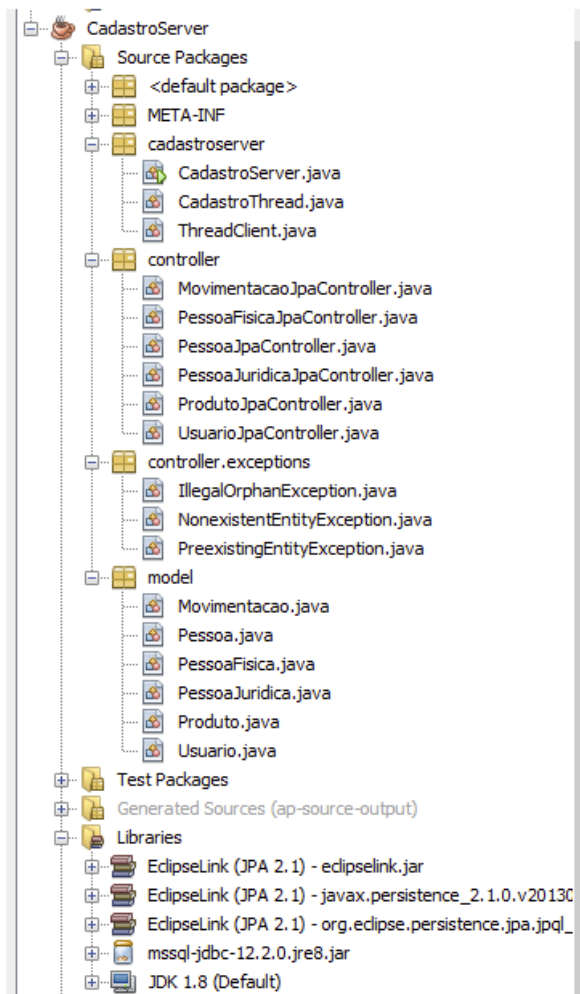
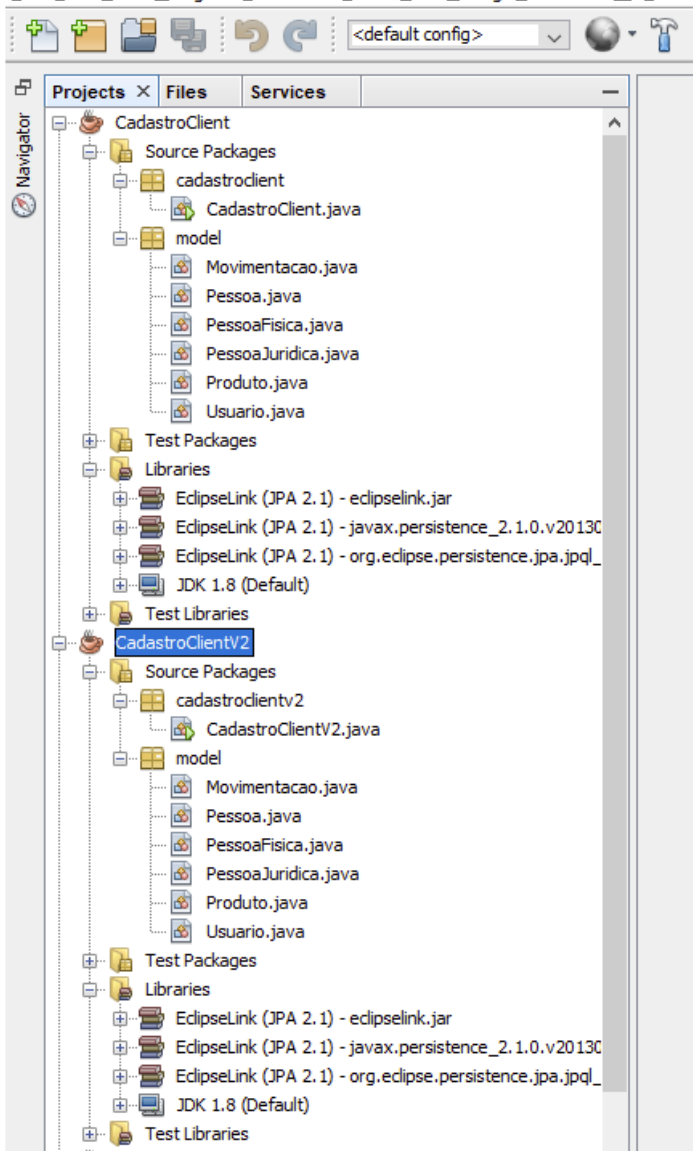
## Prints das configurações

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

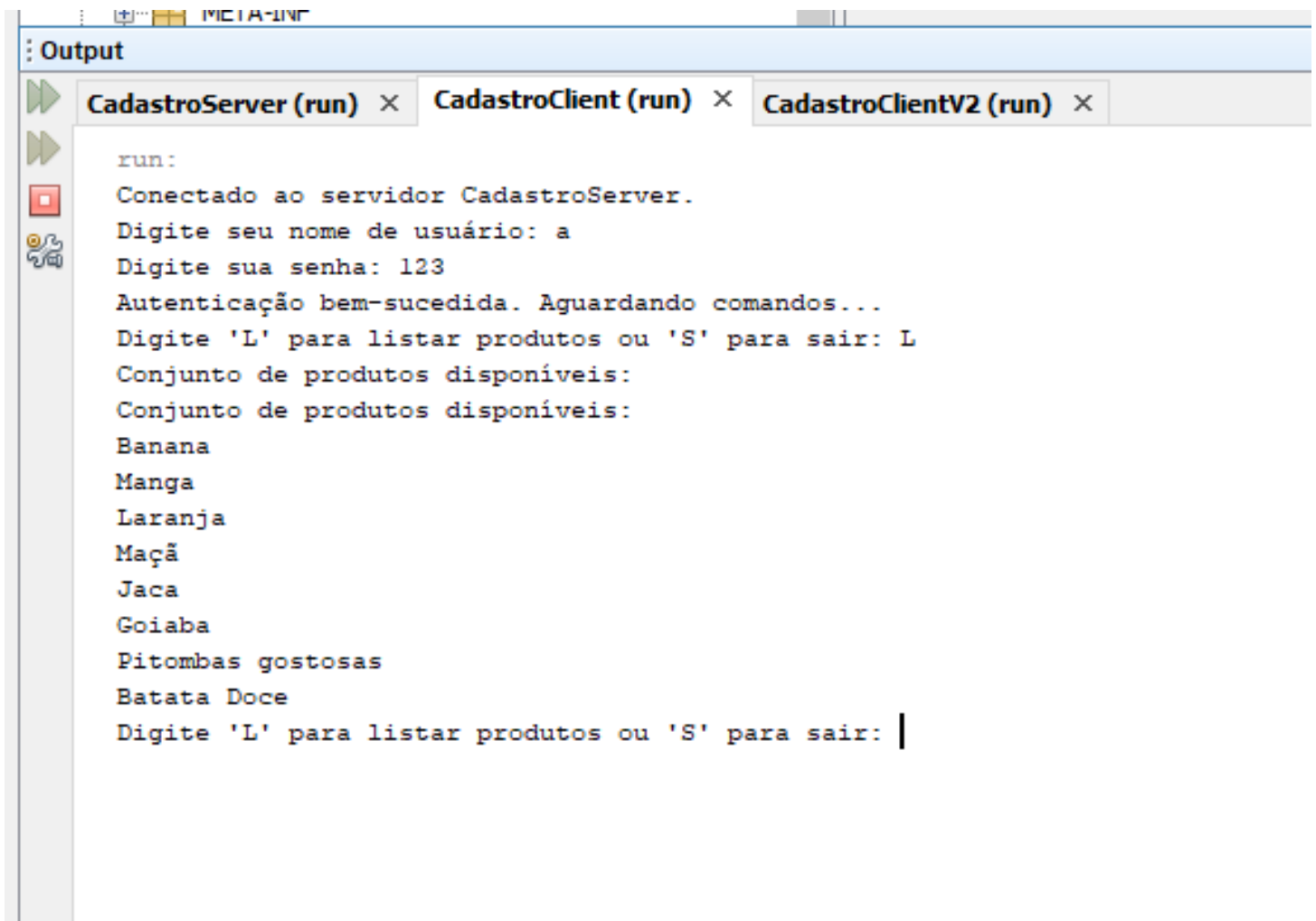
The screenshot displays the Eclipse IDE interface. The **Project Explorer** on the left shows a project named **loja** with a **dbo** database. The **Package Explorer** on the right shows the database schema, including tables (**Movimentacao**, **Pessoa**, **PessoaFisica**, **PessoaJuridica**, **Produto**, **Usuario**), views, procedures, and other schemas. The **Output Console** at the bottom shows the following log messages:

```
Java DB Database Process x GlassFish Server 4.1.1 x
Tue Sep 19 08:49:23 BRT 2023 : Security manager installed using the Basic server security policy.
Tue Sep 19 08:49:23 BRT 2023 : Apache Derby Network Server - 10.10.2.0 - (1582446) started and ready to accept connections on port 1527
```

File Edit View Navigate Source Refactor Run Debug Profile Team Tools



## Print das execuções



```
run:
Conectado ao servidor CadastroServer.
Digite seu nome de usuário: a
Digite sua senha: 123
Autenticação bem-sucedida. Aguardando comandos...
Digite 'L' para listar produtos ou 'S' para sair: L
Conjunto de produtos disponíveis:
Conjunto de produtos disponíveis:
Banana
Manga
Laranja
Maçã
Jaca
Goiaba
Pitombas gostosas
Batata Doce
Digite 'L' para listar produtos ou 'S' para sair: |
```

## Exibindo através de um Menu - Listar

```
CadastroServer (run) × CadastroClient (run) × CadastroClientV2 (run) ×  
  
run:  
Conectado ao servidor CadastroServer.  
Digite seu nome de usuário: a  
Digite sua senha: 123  
Autenticação bem-sucedida. Aguardando comandos...  
Menu:  
L - Listar  
X - Finalizar  
E - Entrada  
S - Saída  
Escolha uma opção: L  
Conjunto de produtos disponíveis:  
Conjunto de produtos disponíveis:  
Banana  
Manga  
Laranja  
Maçã  
Jaca  
Goiaba  
Pitombas gostosas  
Batata Doce  
..
```

## Exibindo através de um Menu –

```

Menu:
L - Listar
X - Finalizar
E - Entrada
S - Saída
Escolha uma opção: E
Informe o Id da pessoa: 1
Informe o Id do produto: 3
Informe a quantidade: 500
Informe o valor unitário: 20.55

```

4	4	2	3	3	Menu:	52US	6.4
5	5	2	5	4		380 E	4.9
6	6	1	5	4		200 E	6.9
7	7	1	4	7		100 S	3.5
8	8	1	4	5		80 S	7.33
9	9	2	7	1		450 E	4.44
10	10	2	9	2		550 S	5.55
11	11	<NULL>	1	<NULL>		300 E	55.5
12	12	<NULL>	2	<NULL>		700 E	45.0
13	13	<NULL>	2	<NULL>		60 S	50.0
14	14	<NULL>	1	<NULL>		25 E	8.0
15	15	<NULL>	1	<NULL>		500 E	20.55



## **Análise e Conclusão:**

### **1 - Como funcionam as classes Socket e ServerSocket?**

ServerSocket: Usado no lado do servidor para aguardar conexões de clientes em uma porta específica.

Socket: Usado tanto no lado do cliente quanto no lado do servidor para estabelecer uma conexão de rede entre dois pontos. Permite a comunicação entre eles por meio de entrada e saída de dados.

### **2 - Qual a importância das portas para a conexão com servidores?**

As portas são números usados para identificar serviços em servidores, rotear dados corretamente, garantir a segurança e permitir conexões múltiplas em um único servidor. Elas desempenham um papel crucial na comunicação eficaz entre computadores em uma rede.

### **3 - Para que servem as classes de entrada e saída ObjectInputStream e ObjectOutputStream, e por que os objetos transmitidos devem ser serializáveis?**

ObjectInputStream e ObjectOutputStream são usados em Java para serializar (converter em bytes) e desserializar (reconstruir a partir de bytes) objetos.

Eles permitem que objetos sejam transmitidos através da rede, armazenados em arquivos e compartilhados entre sistemas.

Os objetos transmitidos devem ser serializáveis, o que significa que suas classes devem implementar a interface Serializable, para que possam ser convertidos em bytes de forma segura. Isso é importante para a comunicação e persistência de dados em Java.

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

O isolamento do acesso ao banco de dados ao usar classes de entidades JPA é garantido através do suporte a transações, controle de transação, gerenciamento de conexão e mapeamento de entidades fornecidos pela JPA. Isso permite operações seguras e isoladas, mesmo em ambientes concorrentes.

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

Threads podem ser usadas para tratar respostas assíncronas do servidor permitindo que várias operações sejam processadas simultaneamente. Isso pode ser feito criando threads individuais, usando um pool de threads, callbacks, programação reativa ou async/await, dependendo dos requisitos do aplicativo. Certifique-se de gerenciar a concorrência com cuidado para evitar problemas.

## **6 - Para que serve o método invokeLater, da classe SwingUtilities?**

O método invokeLater da classe SwingUtilities em Java é usado para executar uma tarefa (geralmente uma operação na interface gráfica do usuário - GUI) de forma assíncrona na thread de despacho de eventos Swing (também conhecida como a thread EDT - Event Dispatch Thread). A EDT é a thread principal responsável pela atualização da interface gráfica em aplicativos Swing.

A principal finalidade do invokeLater é evitar problemas de concorrência ao atualizar a interface gráfica em um programa Swing. Isso ocorre porque a maioria das operações de GUI deve ser executada na EDT para garantir a consistência e a segurança da interface do usuário. Quando você chama o invokeLater, a tarefa especificada é enfileirada na EDT e executada em segundo plano, permitindo que a EDT continue processando eventos de GUI e evitando bloqueios na interface do usuário.

## **7 - Como os objetos são enviados e recebidos pelo Socket Java?**

Em Java, objetos são enviados e recebidos por meio de sockets usando ObjectOutputStream para enviar e ObjectInputStream para receber. Os objetos são automaticamente serializados em bytes ao enviar e desserializados ao receber, mas eles devem ser serializáveis (implementar Serializable) e ambos os lados devem usar a mesma classe para desserialização correta.

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

Comportamento síncrono em clientes de Socket Java bloqueia a thread principal, tornando a GUI não responsiva, mas é mais simples de implementar. Comportamento assíncrono não bloqueia a thread principal, permitindo uma GUI responsiva, mas pode ser mais complexo de implementar e é mais escalável para lidar com várias conexões simultâneas. A escolha depende dos requisitos do aplicativo.