

# CAMPUS POLO AUSTIN - NOVA IGUAÇU – RJ DESENVOLVIMENTO FULL STACK

Nível 5: Por Que Não Paralelizar?

RPG0018 9001

2024/2

SALOMÃO ISAAC CARVALHO GARCIA

Implementação de Comunicação via Socket com Persistência JPA

Nova Iguaçu 15/09/2024

# Implementação de Comunicação via Socket com Persistência JPA

**Objetivo da Prática**: O objetivo principal desta prática é desenvolver uma aplicação de comunicação cliente-servidor utilizando sockets para comunicação de rede e JPA (Java Persistence API) para persistência de dados. A prática visa proporcionar uma compreensão profunda dos seguintes conceitos e técnicas:

# 1. Comunicação via Sockets:

- Implementar um servidor que utiliza ServerSocket para escutar conexões de clientes e Socket para estabelecer comunicação bidirecional.
- Desenvolver um cliente que se conecta ao servidor e envia e recebe mensagens, utilizando streams de entrada e saída para transmissão de dados.

#### 2. Persistência de Dados com JPA:

- Configurar uma camada de persistência usando JPA para gerenciar o armazenamento e recuperação de dados em um banco de dados relacional.
- Utilizar entidades JPA para representar e manipular dados de forma eficiente e abstraída.

# 3. Integração Cliente-Servidor:

 Garantir a integração entre a comunicação via sockets e a persistência de dados, onde o cliente envia comandos ao servidor que interage com o banco de dados e retorna resultados para o cliente.

# 4. Segurança e Isolamento:

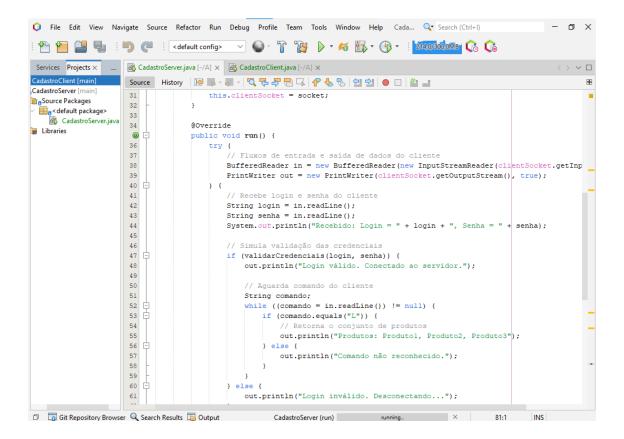
- Implementar um sistema básico de autenticação para validar as credenciais do usuário antes de permitir o acesso às funcionalidades do servidor.
- Assegurar que o acesso ao banco de dados seja isolado, de modo que o cliente não tenha acesso direto às operações de persistência, garantindo segurança e integridade dos dados.

A prática visa não apenas a implementação técnica desses componentes, mas também a compreensão de como a comunicação em rede e a persistência de dados interagem em um ambiente cliente-servidor. O resultado esperado é um sistema funcional que demonstre a habilidade de combinar esses dois aspectos de forma eficiente e segura.

https://github.com/SaloGarcia/trabalhon5.git

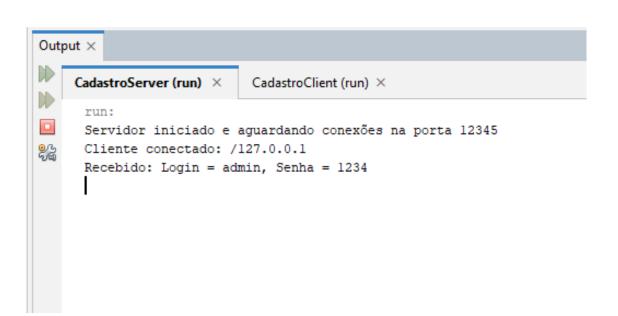
# CÓDIGOS

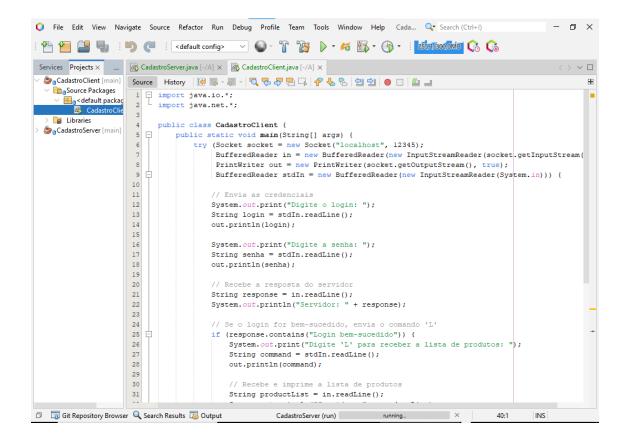
```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... 🗣 Search (Ctrl+I)
                                                                                                                        - o ×
|| 🚰 🚰 📮 || 🦃 🏴 || (4efault config> 🔻 🚳 - 🚡 🚳 - 🏇 - 🚯 - | 2004/3000000 🔥 💪
Services Projects × _ GadastroServer.java [-/A] × GadastroClient.java [-/A] ×
                     Source History | 🔀 🖟 🔻 🔻 🗸 🗸 🖓 🖶 🖫 | 😤 😂 | 💇 💇 | 💿 🗆 | 💯 🚅
                                                                                                                                 -
CadastroServer [main]
                       import java.io.*;
import java.net.*;
maSource Packages
 equit package
   CadastroServer.java
                            public class CadastroServer {
Libraries
                                // Porta que o servidor vai escutar
private static final int PORT = 12345;
                                public static void main(String[] args) {
                                    try (ServerSocket serverSocket = new ServerSocket(PORT)) {
                       10
                                         System.out.println("Servidor iniciado e aguardando conexões na porta " + PORT);
                       11
                       12
                                         while (true) {
                       13
                                             // Espera uma conexão do cliente
                       14
                                             Socket clientSocket = serverSocket.accept();
                                             System.out.println("Cliente conectado: " + clientSocket.getInetAddress());
                       15
                       16
                       17
                                             // Cria uma nova thread para lidar com a conexão do cliente
                       18
                                             new ClientHandler(clientSocket).start();
                       19
                       20
                       21
                                     } catch (IOException e) {
                                        e.printStackTrace();
                       23
                       24
                       25
                                 // Classe interna para lidar com a comunicação do cliente em uma thread separada
                      27 =
<u>Q</u>
                                 private static class ClientHandler extends Thread {
                                    private Socket clientSocket;
                       30
                                     public ClientHandler(Socket socket) {
                       31
                                        this.clientSocket = socket;
☐ Git Repository Browser Q Search Results 🕫 Output
                                                        CadastroServer (run)
                                                                                 running...
```

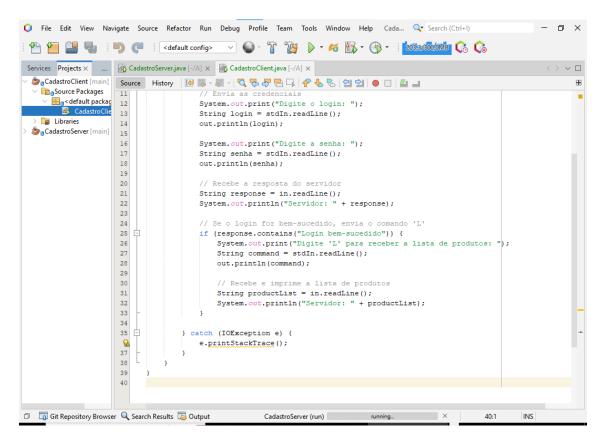


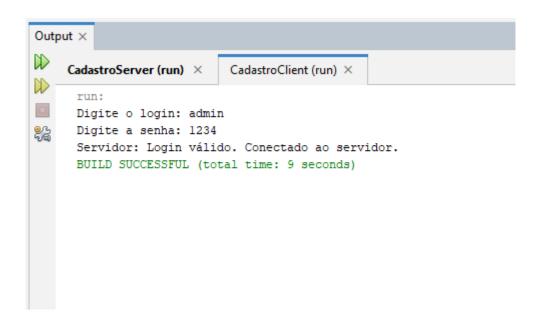
```
- 🗇 ×
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... 🗣 Search (Ctrl+I)
: 😷 🚰 🛂 🤚 : 🍏 🏴 : <default config> 🔻 🚳 ະ 🚡 ፣ 🍇 🖫 ፣ 🦠 ፣ 🕦 ፣ : 194/3000MB 🔥 💪
Services Projects \times _ _ \bigcirc CadastroServer.java [-/A] \times \bigcirc CadastroClient.java [-/A] \times
Source History 🖟 🐺 🔻 🗸 🖓 🖶 📮 🖟 😓 🖭 💇 💿 🗆 🏰 🚉
                                                                                                               4
CadastroServer [main]

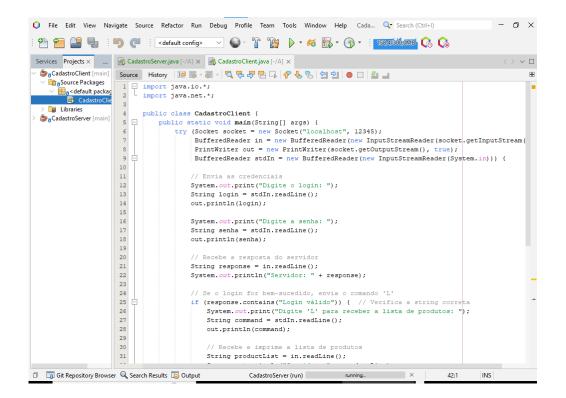
Source Packages
                   55
                                                 out.println("Produtos: Produto1, Produto2, Produto3");
                   56
                                              } else {
   GadastroServer.
                                                 out.println("Comando não reconhecido.");
                   58
 Libraries
                   59
                   60
                                      } else {
                   61
                                          out.println("Login inválido. Desconectando...");
                   62
                   63
                   64 =
66 =
                                  } catch (IOException e) {
                                      e.printStackTrace();
                                   } finally {
                   67 E
68 E
                                      try {
                                          clientSocket.close();
                                      } catch (IOException e) {
                   71
72
                                         e.printStackTrace();
                   73
                   74
75
                               // Simula validação de login e senha
                   76
                               private boolean validarCredenciais(String login, String senha) {
                   77
78
                                   return "admin".equals(login) && "1234".equals(senha); // Exemplo simples de valida
                   80
                   81
INS
```





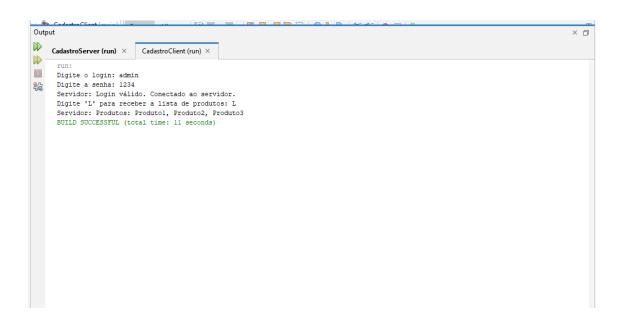






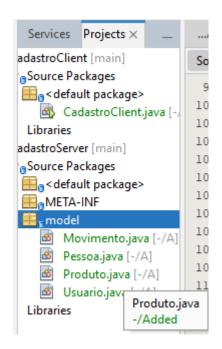
```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... 🗣 Search (Ctrl+I)
                                                                                                                          - o ×
Services Projects × _ _ GadastroServer.java [-/A] × GadastroClient.java [-/A] ×
                                                                                                                                 v 🗆
 4
  ✓ □ Source Packages
                       19
      edefault packag

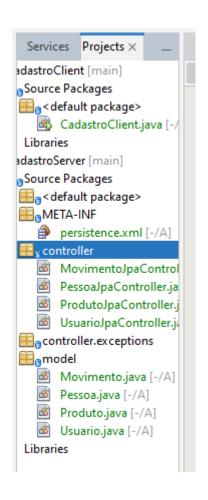
CadastroClie
                      20
                                          // Recebe a resposta do servidor
                       21
                                          String response = in.readLine();
   Libraries
                       22
                                          System.out.println("Servidor: " + response);
 SadastroServer [m
                      23
24
                                          // Se o login for bem-sucedido, envia o comando 'L'
                                         if (response.contains("Login válido")) { // Verifica a string correta
   System.out.print("Digite 'L' para receber a lista de produtos: ");
   String command = stdIn.readLine();
                       25
                      26
27
                                             out.println(command);
                      29
30
                                             // Recebe e imprime a lista de produtos
String productList = in.readLine();
System.out.println("Servidor: " + productList);
                       31
                      32
33
                                          } else {
                       34
                                             System.out.println("Falha no login. Desconectando...");
                      35
36
                       37
<u>Q</u>
39
                                     } catch (IOException e) {
                                         e.printStackTrace();
                       40
                       41
                       42
```

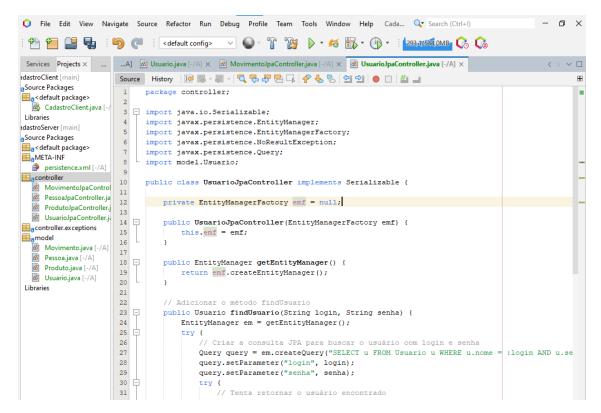


```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... 🗣 Search (Ctrl+I)
                                                                                                                     - o ×
: 😷 🚰 🛂 🤚 : 🎙 🥒 : (<default config> 🔻 🚳 - 🔐 🎉 👂 - 🍇 🐘 - 🕦 - : : 📆 😘 🗞 🐍
Services Projects × _ ....A] 🎒 persistence.xml [-/A] × 🔞 Movimento.java [-/A] × 🔞 Pessoa.java [-/A] × 🔞 Produto.java [-/A] × 🔞 Usuario.java [-... < > > 🗆
 Source Packages
    * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this l
* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
  > la Libraries
  CadastroServer [main]
                             package model;
    Source Packages
    7 import java.io.Serializable;
                             import java.math.BigDecimal;
         persistence.x
                             import java.util.Date;
      model
                             import javax.persistence.Basic;
        Movimento.
                             import javax.persistence.Column;
         Pessoa.java [
                             import javax.persistence.Entity;
         Produto.java
                             import javax.persistence.Id;
         ■ Usuario.java
                             import javax.persistence.JoinColumn;
  🗦 📴 Libraries
                             import javax.persistence.ManyToOne;
                             import javax.persistence.NamedQueries;
                             import javax.persistence.NamedQuery;
                             import javax.persistence.Table;
                       19
                             import javax.persistence.Temporal;
                       20
                             import javax.persistence.TemporalType:
                       21
                          - /**
                       22
                       23
                             * @author user
                       25
                       26
                             @Entity
                       27
                             @Table(name = "Movimento")
                       28
                             @NamedOueries({
                                 @NamedQuery(name = "Movimento.findAll", guery = "SELECT m FROM Movimento m"),
                       29
                                 ONamedQuery(name = "Movimento.findByIdMovimento", query = "SELECT m FROM Movimento m WHERE

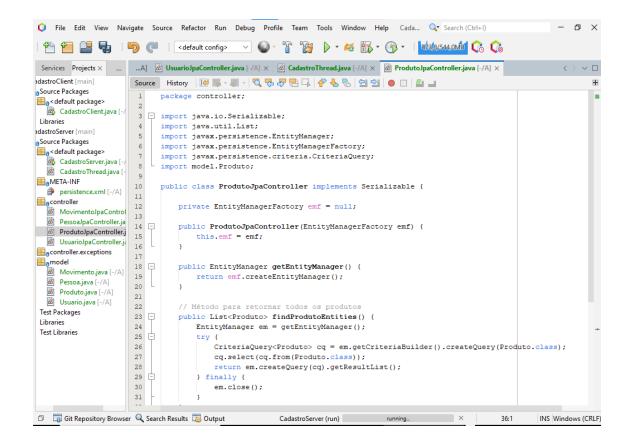
@NamedQuery(name = "Movimento.findByQuantidade", query = "SELECT m FROM Movimento m WHERE m
                       31
☐ ☐ Git Repository Browser Q Search Results ☐ Output CadastroServer (run) running. × 1:1
```



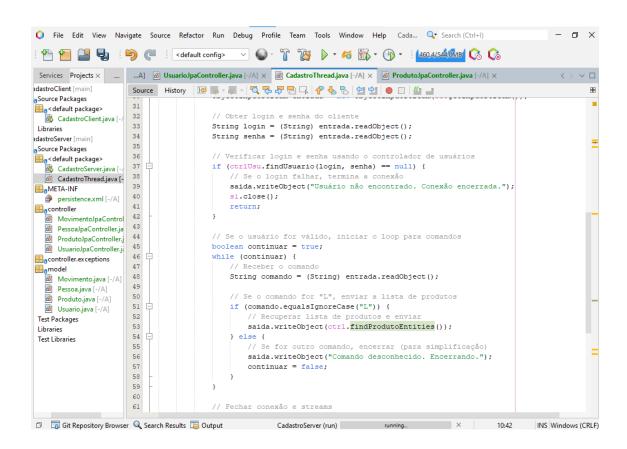




```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... 🗣 Search (Ctrl+1)
                                                                                                                       - o ×
Services Projects × _ ....A] 🙆 Usuario.java [-/A] × 🙆 MovimentoJpaController.java [-/A] × 🙆 UsuarioJpaController.java [-/A] ×
adastroClient [main]
                      4
Source Packages
                      16
Bo<default package
  CadastroClient.iava [-
                      18 📮
                                public EntityManager getEntityManager() {
Libraries
                      19
20
                                   return emf.createEntityManager();
adastroServer [main]
Source Packages
                      21
= < default package>
                      22
                                // Adicionar o método findUsuario
⊞<sub>0</sub>META-INF
                      23 🖃
                                public Usuario findUsuario(String login, String senha) {
  persistence.xml [-/A]
                                    EntityManager em = getEntityManager();
                      24
                      24
25 =
Recontroller ...
                                    try {
  MovimentoJpaControl
PessoaJpaController.ja
                                         .
// Criar a consulta JPA para buscar o usuário com login e senha
                      26
                                        Query query = em.createQuery("SELECT u FROM Usuario u WHERE u.nome = :login AND u.se
  ProdutoJpaController.j
                                        query.setParameter("login", login);
query.setParameter("senha", senha);
                      28
  ■ UsuarioJpaController.j.
                      29
■gcontroller.exceptions
                                        try {
⊞@model
                                            // Tenta retornar o usuário encontrado
                      31
     Movimento.java [-/A]
                                            return (Usuario) query.getSingleResult();
                      32
  Pessoa.java [-/A]
Produto.java [-/A]
Usuario.java [-/A]
                      33
                                        } catch (NoResultException e) {
                      34
                                            // Retorna null se não houver resultado
                                            return null;
                      35
Libraries
                         ψ.
                      37
                                    } finally {
                                        em.close();
                      38
                      40
                      41
                            }
🗇 🌀 Git Repository Browser 🔍 Search Results 🗔 Output Cadastro Server (run) running... × 12:45 INS Windows (CRLF)
```



```
Q File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... Q▼ Search (Ctrl+I)
                                                                                                                                                                                                                                                                  - o ×
: 😷 🚰 🛂 🚦 : 🥱 🧖 : <default config> 🔻 🚳 - 🔐 🎉 👂 - 🍇 🐘 - 🕦 - : : 👊 2,5/,544 ôm 🖟 📞
 Services Projects × _ ...A] 🖄 UsuarioJpaController.java [-/A] × 🕅 CadastroThread.java [-/A] × 🗗 ProdutoJpaController.java [-/A] ×
                                                Source History | 164 | 155 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 157 | 
dastroClient [main]
                                                                                                                                                                                                                                                                                     P
 Source Packages
                                                  1 = import controller.ProdutoJpaController;
                                                                                                                                                                                                                                                                                      e-< default package>
                                                             import controller.UsuarioJpaController;
     CadastroClient.java [-
                                                             import java.io.ObjectInputStream;
 Libraries
                                                             import java.io.ObjectOutputStream;
adastroServer [main]
                                                            import java.net.Socket;
Source Packages
Ba<default package>
                                                             public class CadastroThread implements Runnable {
     CadastroServer.java
    CadastroThread.java [-
                                                                     private ProdutoJpaController ctrl;
META-INF
                                                                      private UsuarioJpaController ctrlUsu;
     persistence.xml [-/A]
                                                  <u>Q.</u>
                                                                      private Socket sl;
⊞<sub>@</sub>controller
      MovimentoJpaControl
                                               13 📮
     PessoaJpaController.ja
                                                                      * Construtor da classe CadastroThread.
                                                14
     ProdutoJpaController.j
                                                                       * @param ctrl Controlador de produtos.
     UsuarioJpaController.j.
                                                16
                                                                       * @param ctrlUsu Controlador de usuários
econtroller.exceptions
                                                                      * @param sl Socket para comunicação.
⊞@model
     Movimento.iava [-/A]
                                                19 🖃
                                                                      public CadastroThread(ProdutoJpaController ctrl, UsuarioJpaController ctrlUsu, Socket sl) {
     Pessoa.java [-/A]
                                                20
                                                                              this.ctrl = ctrl;
     Produto.java [-/A]
                                                                               this.ctrlUsu = ctrlUsu;
                                                21
     Usuario.iava [-/A]
                                                22
                                                                               this.sl = sl;
  Test Packages
                                                23
 Libraries
                                                24
 Test Libraries
                                                25
                                                                      @Override
                                                 (1) [-]
                                                                      public void run() {
                                                                            try {
                                                28
<u>Q</u>
                                                                                       // Encapsular os canais de entrada e saída do socket
ObjectOutputStream saida = new ObjectOutputStream(sl.getOutputStream());
                                                                                        ObjectInputStream entrada = new ObjectInputStream(sl.getInputStream());
                                                 31
🗇 🏮 Git Repository Browser 🔍 Search Results 😇 Output CadastroServer (run) running... 💢 68:10 INS Windows (CRLF)
```



```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... 🗣 Search (Ctrl+1)
: 🚰 🚰 📮 : 🥱 🎮 : (default config> 🔻 🎧 - 📅 👺 👂 - 🗯 🚯 - 🕦 - : 🙉 834.8/544.0MB. 🗘 🕻
Services Projects × _ ....A] 🚳 UsuarioJpaController.java [-/A] × 🚳 CadastroThread.java [-/A] × 🚳 ProdutoJpaController.java [-/A] ×
adastroClient [main]
                       Source History | № 🖟 🔻 🔻 🗸 🗸 🞝 🖶 🖫 | 春 😓 | 🕮 💇 | • □ | 😃 📑
Source Packages
                        46
                                           while (continuar) {
Eg<default package>
                        47
                                                // Receber o comando
  CadastroClient.java [-/
                                               String comando = (String) entrada.readObject();
                        48
Libraries
                        49
dastroServer [main]
                                               // Se o comando for "L", enviar a lista de produtos
Source Packages
                        51
                                               if (comando.equalsIgnoreCase("L")) {

——
«default package»

                        52
                                                    // Recuperar lista de produtos e enviar
  CadastroServer.iava [
                                                    saida.writeObject(ctrl.findProdutoEntities());
                       53
 CadastroThread.java [-
■ META-INF
                                                   // Se for outro comando, encerrar (para simplificação)
                        55
  persistence.xml [-/A]
                        56
                                                    saida.writeObject("Comando desconhecido. Encerrando.");
⊞<sub>@</sub>controller
                                                    continuar = false;
  MovimentoJpaControl
  PessoaJpaController.ja 58

■ ProdutoJpaController.j

  ■ UsuarioJpaController.ja
                       61
                                           // Fechar conexão e streams
econtroller.exceptions
                       62
                                           entrada.close():
⊞<sub>@</sub>model
                                           saida.close();
  Movimento.java [-/A]
                       63
                                           sl.close();
  Pessoa.java [-/A]
  Produto.java [-/A]
                                       } catch (Exception e) {

■ Usuario.iava [-/A]

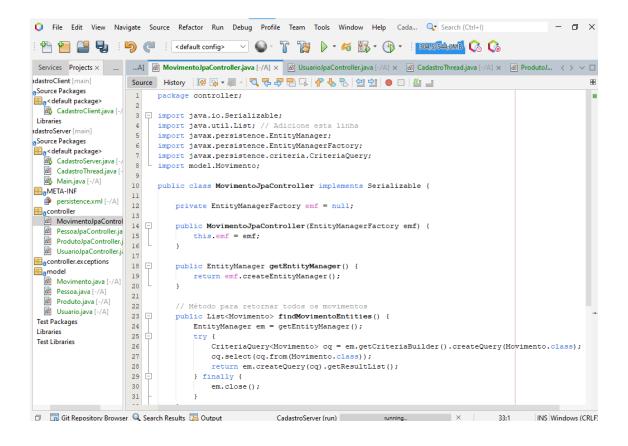
                                           e.printStackTrace();
Test Packages
                       68
Libraries
                       69
Test Libraries
```

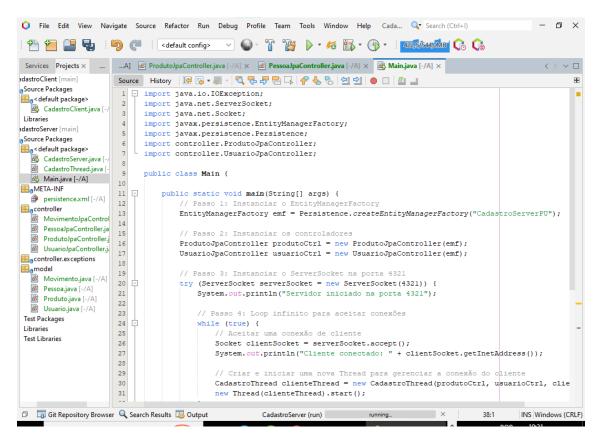
```
Q File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... Q▼ Search (Ctrl+I)
                                                                                                                                                                                                                                                             - 0
: 🚰 🚰 📮 : 🥱 🎮 : (default config) 🗸 🚳 - 🔐 🔯 👂 - 🍇 🚮 - (1) - : | 4422/544.0MB| 🗘 💪
Services Projects X _ ...A] 🚳 CadastroThread.java [-/A] X 🚳 ProdutoJpaController.java [-/A] X 🚳 PessoaJpaController.java [-/A] X 🚳 Main.java [-/A]... < > >
dastroClient [main]
                                              Source Packages
                                                           package controller;
default package>
    CadastroClient.iava [-/
                                                Libraries
                                                           import java.util.List; // Importar a classe List
adastroServer [main]
                                                           import javax.persistence.EntityManager;
Source Packages
                                                           import javax.persistence.EntityManagerFactory;
A compare the compare of the comp
                                                          import javax.persistence.criteria.CriteriaQuery;
     CadastroServer.java [-/
                                                     import model.Pessoa;
    Main.java [-/A]
                                              10
                                                           public class PessoaJpaController implements Serializable {
⊞

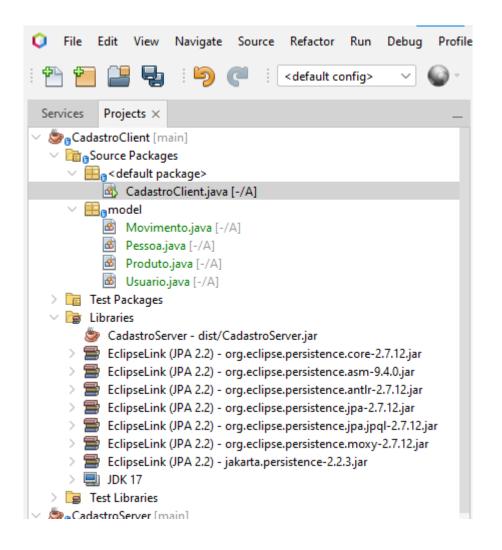
■META-INF
                                              11
     persistence.xml [-/A]
                                                                    private EntityManagerFactory emf = null;
                                              12
⊞<sub>@</sub>controller
     Movimento Jpa Control
                                              14
                                                                    public PessoaJpaController(EntityManagerFactory emf) {

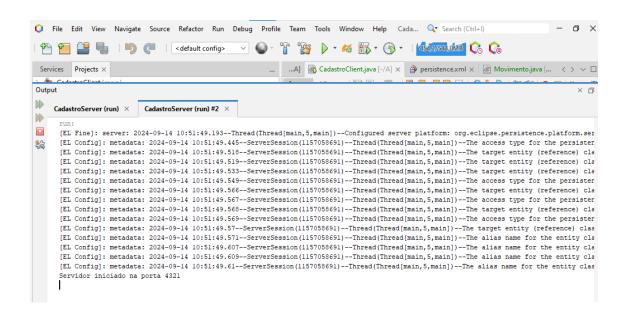
    PessoaJpaController.ja

                                                                        this.emf = emf;
                                              15
     ProdutoJpaController.j
                                               16
    ■ UsuarioJpaController.ja
econtroller.exceptions
                                               18 📮
                                                                   return emf.createEntityManager();
}
                                                                    public EntityManager getEntityManager() {
model
                                               19
     Movimento.java [-/A]
                                              20
    Pessoa.java [-/A]
                                               21
    Produto.java [-/A]
                                               22
                                                                     // Método para retornar todas as pessoas
     Usuario.java [-/A]
                                              23 🖃
                                                                    public List<Pessoa> findPessoaEntities() {
 Test Packages
                                              24
                                                                         EntityManager em = getEntityManager();
 Libraries
                                               25
                                                                             try {
 Test Libraries
                                              26
                                                                                    CriteriaQuery<Pessoa> cq = em.getCriteriaBuilder().createQuery(Pessoa.class);
                                                                                    cq.select(cq.from(Pessoa.class));
                                               27
                                               28
                                                                                     return em.createQuery(cq).getResultList();
                                               29
                                                                             } finally {
                                               30
                                                                                    em.close();
                                               31
                                                0.0 1.0 1. ...
```









```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... 🗣 Search (Ctrl+1)
                                                                                                                                                                                                                                                                - o ×
: 😷 🚰 📮 📲 🤚 (□ : | <default config> ∨ 🚳 - 🏗 👺 👂 + 66 🚯 + 🕦 + : | 3302/544,0MB| 📞 📞
                                                       CadastroServer.java [-/A] × 

GAGAStroClient.java [-/A] × 
Persistence.xml × 

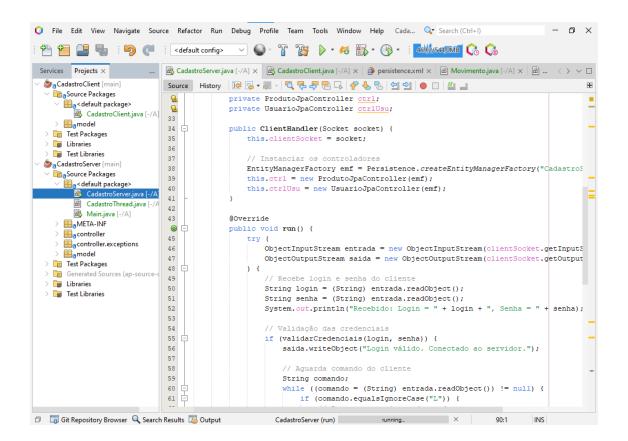
Movimento.java [-/A] × 

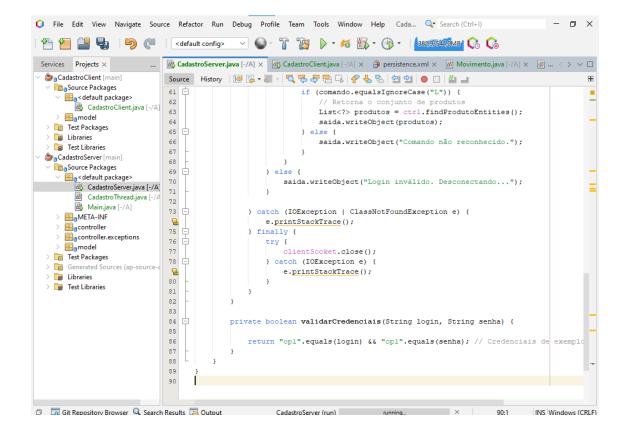
Movimento.java [
  Services Projects ×
    SacadastroClient [main]
                                                                   Source History | 🔀 🖟 🔻 🔻 🔻 🖓 🖓 🖓 🕞 | 😭 💇 | ● 🗆 | 💯 🚅
         Source Packages
                                                                    1 = import java.io.*;

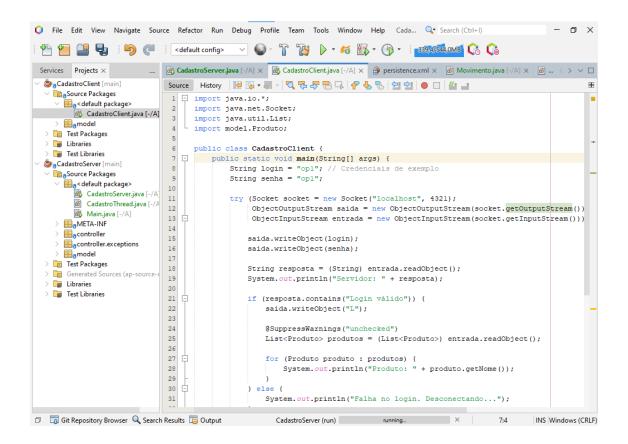
✓ Eg < default package >

GadastroClient.java [-/A]

                                                                               import java.net.*;
                                                                               import java.util.List;
              ⊞<sub>@</sub>model
                                                                               import javax.persistence.EntityManagerFactory;
        Test Packages
                                                                               import javax.persistence.Persistence;
         Libraries
                                                                               import controller.ProdutoJpaController;
         Test Libraries
                                                                               import controller.UsuarioJpaController;
    CadastroServer [main]
                                                                           import model.Produto;
        Source Packages
               🖺 g < default package>
                                                                               public class CadastroServer {
                  CadastroServer.java [-/A]
CadastroThread.java [-/A
                                                                   11
                                                                                      private static final int PORT = 4321;
                                                                   12
                    Main.java [-/A]
                                                                   13 F
14 F
                                                                                        public static void main(String[] args) {
          > EnMETA-INF
                                                                                                 try (ServerSocket serverSocket = new ServerSocket(PORT)) {
              # controller
                                                                   15
                                                                                                          System.out.println("Servidor iniciado e aguardando conexões na porta " + POF
              Egcontroller.exceptions
                                                                   16
              model
     > iii Test Packages
                                                                   18
                                                                                                                   Socket clientSocket = serverSocket.accept();
     > Generated Sources (ap-sour
                                                                                                                   System.out.println("Cliente conectado: " + clientSocket.getInetAddress()
                                                                   19
        Libraries
     > Test Libraries
                                                                   21
                                                                                                                   new ClientHandler(clientSocket).start();
                                                                   23
                                                                   24
                                                                                                  } catch (IOException e) {
                                                                                                          e.printStackTrace();
                                                                   26
                                                                   29 -
                                                                                         private static class ClientHandler extends Thread {
                                                                                                 private Socket clientSocket;
                                                                                                  private ProdutoJpaController ctrl;
🗇 🥫 Git Repository Browser 🔍 Search Results 🕫 Output
                                                                                                                        CadastroServer (run) running... ×
                                                                                                                                                                                                                                       90:1
                                                                                                                                                                                                                                                          INS
```







```
Output ×

CadastroServer (run) × CadastroClient (run) ×

run:
Servidor: Login válido. Conectado ao servidor.
Produto: Banana
Produto: Laranja
Produto: Manga
BUILD SUCCESSFUL (total time: 4 seconds)
```

# Resultados:

#### **Procedimento 01:**

```
Output ×

CadastroServer (run) × CadastroClient (run) ×

run:
Servidor: Login válido. Conectado ao servidor.

Produto: Banana
Produto: Laranja
Produto: Manga
BUILD SUCCESSFUL (total time: 4 seconds)
```

# Como Funcionam as Classes Socket e ServerSocket?

As classes Socket e ServerSocket são fundamentais para a comunicação cliente-servidor em Java.

#### ServerSocket:

- O ServerSocket é utilizado no lado do servidor para escutar conexões de entrada em uma porta específica.
- Ele aguarda a solicitação de conexão dos clientes e, quando um cliente tenta se conectar, o ServerSocket aceita a conexão e cria um novo Socket para interagir com o cliente.
- O ServerSocket permite que o servidor mantenha a escuta em uma porta, esperando por múltiplas conexões de clientes ao longo do tempo.

#### Socket:

- O Socket é utilizado tanto no lado do cliente quanto no lado do servidor para estabelecer uma comunicação de rede após a conexão ser aceita.
- Ele fornece os meios para enviar e receber dados entre o cliente e o servidor através de streams de entrada e saída.
- Em um cliente, o Socket conecta-se a um ServerSocket em um servidor, estabelecendo um canal de comunicação que pode ser usado para trocar dados.

# Qual a Importância das Portas para a Conexão com Servidores?

As portas são essenciais para a comunicação de rede porque elas identificam um ponto específico em um servidor para onde os dados devem ser enviados. Cada serviço de rede em um servidor escuta em uma porta específica:

# Identificação de Serviços:

 A porta atua como um identificador para o serviço específico que está sendo solicitado. Por exemplo, um servidor de web pode estar escutando na porta 80 (HTTP) ou 443 (HTTPS).

# Multiplexação:

 Em um único servidor, diferentes serviços podem ser executados em portas diferentes. Isso permite que múltiplos serviços de rede sejam oferecidos ao mesmo tempo.

# Comunicação Direcionada:

 Quando um cliente se conecta a um servidor, ele especifica a porta para onde os dados devem ser enviados. Isso garante que a comunicação é direcionada ao serviço correto no servidor.

Para que Servem as Classes de Entrada e Saída ObjectInputStream e ObjectOutputStream, e por que os Objetos Transmitidos Devem Ser Serializáveis?

# • ObjectInputStream e ObjectOutputStream:

- ObjectOutputStream: Utilizado para escrever objetos para um stream de saída. Ele transforma objetos em um formato que pode ser transmitido para outro lado da comunicação (para um cliente ou servidor).
- ObjectInputStream: Utilizado para ler objetos de um stream de entrada. Ele transforma o formato recebido de volta em objetos Java utilizáveis.

# Objetos Serializáveis:

- Para que os objetos possam ser transmitidos entre cliente e servidor, eles devem ser serializáveis. Isso significa que eles devem implementar a interface Serializable.
- A serialização permite que um objeto seja convertido em um formato que pode ser facilmente armazenado ou transmitido, e depois reconstruído em sua forma original.

Por que, Mesmo Utilizando as Classes de Entidades JPA no Cliente, Foi Possível Garantir o Isolamento do Acesso ao Banco de Dados?

# Isolamento do Banco de Dados:

- Embora o cliente utilize as classes de entidades JPA para manipular dados, ele n\u00e3o interage diretamente com o banco de dados. Todo acesso ao banco de dados \u00e9 mediado pelo servidor.
- O servidor é o único componente que possui a responsabilidade e a capacidade de executar operações de banco de dados. O cliente apenas solicita informações ou comandos que são processados pelo servidor.

# Segurança e Integridade:

Esse design garante que a lógica de acesso ao banco de dados e a integridade dos dados sejam mantidas no lado do servidor. O cliente, portanto, não pode executar diretamente comandos SQL ou modificar dados sem passar pelos controladores e validações implementadas no servidor.

# • Separação de Responsabilidades:

 A separação entre cliente e servidor ajuda a manter uma arquitetura limpa e modular, onde o cliente lida apenas com a interface do usuário e as interações, enquanto o servidor gerencia a persistência de dados e a lógica de negócios.

https://github.com/SaloGarcia/trabalhon5.git