



Estácio

CURSO: Desenvolvimento

FullStack

DISCIPLINA: BackEnd sem banco não tem!

TURMA: 2025.1

SEMESTRE LETIVO: Primeiro Semestre (2025)

ALUNO: Carlos Henrique

Título da prática:

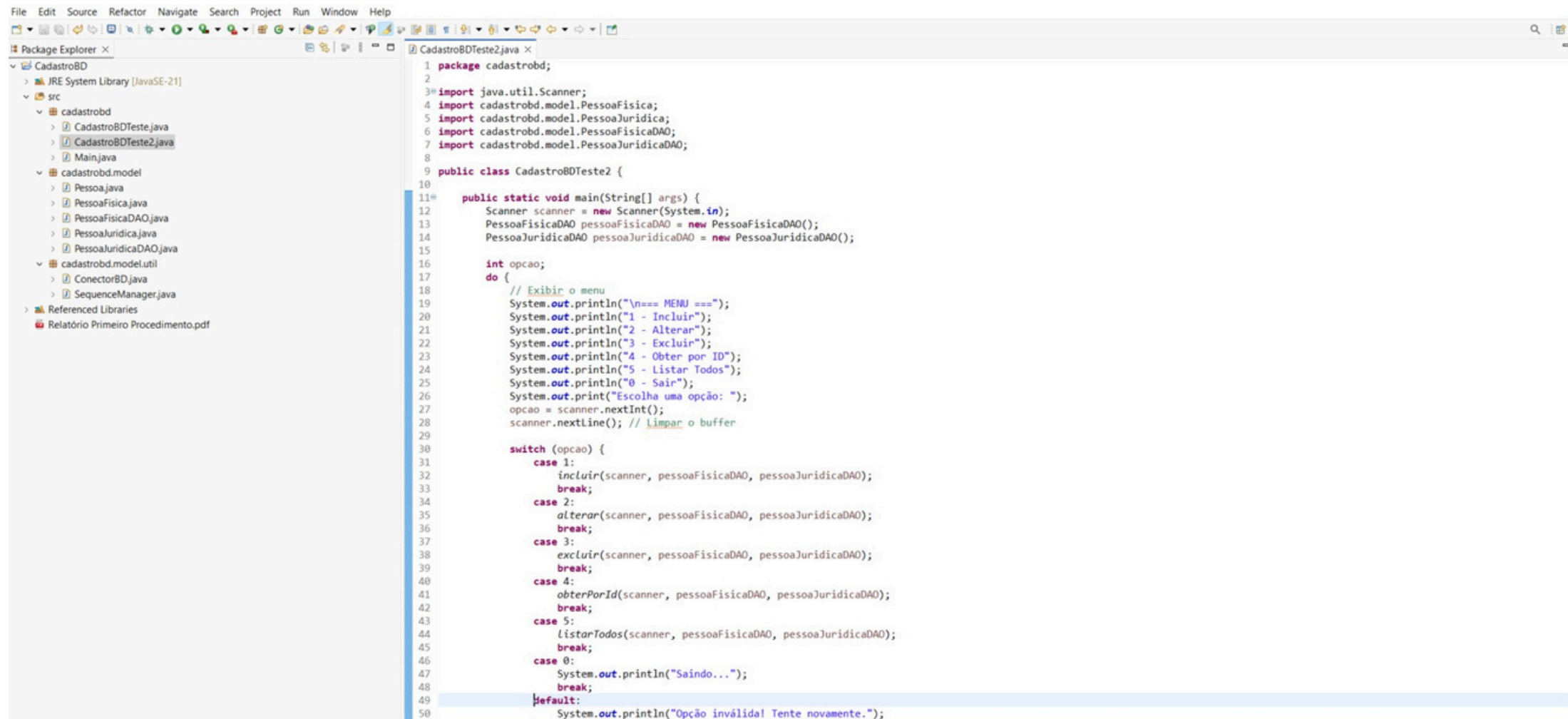
Criação de aplicativo Java, com acesso ao banco de dados SQL Server através do middleware JDBC.

Objetivos da prática:

- 1.Implementar persistência com base no middleware JDBC.
- 2.Utilizar o padrão DAO (Data Access Object) no manuseio de dados.
- 3.Implementar o mapeamento objeto-relacional em sistemas Java.
- 4.Criar sistemas cadastrais com persistência em banco relacional.

2º Procedimento | Alimentando a Base

Códigos usados neste roteiro



The screenshot shows an IDE with the Package Explorer on the left and the source code of `CadastroBDTeste2.java` on the right. The Package Explorer shows the project structure, including the `cadastrobd` package and its sub-packages `model` and `util`. The source code is as follows:

```
1 package cadastrobd;
2
3 import java.util.Scanner;
4 import cadastrobd.model.PessoaFisica;
5 import cadastrobd.model.PessoaJuridica;
6 import cadastrobd.model.PessoaFisicaDAO;
7 import cadastrobd.model.PessoaJuridicaDAO;
8
9 public class CadastroBDTeste2 {
10
11     public static void main(String[] args) {
12         Scanner scanner = new Scanner(System.in);
13         PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO();
14         PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO();
15
16         int opcao;
17         do {
18             // Exibir o menu
19             System.out.println("\n=== MENU ===");
20             System.out.println("1 - Incluir");
21             System.out.println("2 - Alterar");
22             System.out.println("3 - Excluir");
23             System.out.println("4 - Obter por ID");
24             System.out.println("5 - Listar Todos");
25             System.out.println("0 - Sair");
26             System.out.print("Escolha uma opção: ");
27             opcao = scanner.nextInt();
28             scanner.nextLine(); // Limpar o buffer
29
30             switch (opcao) {
31                 case 1:
32                     incluir(scanner, pessoaFisicaDAO, pessoaJuridicaDAO);
33                     break;
34                 case 2:
35                     alterar(scanner, pessoaFisicaDAO, pessoaJuridicaDAO);
36                     break;
37                 case 3:
38                     excluir(scanner, pessoaFisicaDAO, pessoaJuridicaDAO);
39                     break;
40                 case 4:
41                     obterPorId(scanner, pessoaFisicaDAO, pessoaJuridicaDAO);
42                     break;
43                 case 5:
44                     listarTodos(scanner, pessoaFisicaDAO, pessoaJuridicaDAO);
45                     break;
46                 case 0:
47                     System.out.println("Saindo...");
48                     break;
49                 default:
50                     System.out.println("Opção inválida! Tente novamente.");
```

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

CadastroBD

JRE System Library [JavaSE-21]

src

cadastrobd

CadastroBDTeste.java

CadastroBDTeste2.java

Main.java

cadastrobd.model

Pessoa.java

PessoaFisica.java

PessoaFisicaDAO.java

PessoaJuridica.java

PessoaJuridicaDAO.java

cadastrobd.model.util

ConectorBD.java

SequenceManager.java

Referenced Libraries

Relatório Primeiro Procedimento.pdf

CadastroBDTeste2.java

```
52     } while (opcao != 0);
53
54     scanner.close();
55 }
56
57 // Método para incluir uma nova pessoa
58 private static void incluir(Scanner scanner, PessoaFisicaDAO pessoaFisicaDAO, PessoaJuridicaDAO pessoaJuridicaDAO) {
59     System.out.println("\n--- INCLUIR ---");
60     System.out.println("Tipo de pessoa (1 - Fisica, 2 - Juridica): ");
61     int tipo = scanner.nextInt();
62     scanner.nextLine(); // Limpar o buffer
63
64     if (tipo == 1) {
65         System.out.println("Nome: ");
66         String nome = scanner.nextLine();
67         System.out.println("Endereço: ");
68         String endereco = scanner.nextLine();
69         System.out.println("Telefone: ");
70         String telefone = scanner.nextLine();
71         System.out.println("Email: ");
72         String email = scanner.nextLine();
73         System.out.println("CPF: ");
74         String cpf = scanner.nextLine();
75
76         PessoaFisica pessoaFisica = new PessoaFisica(0, nome, endereco, "", "", telefone, email, cpf);
77         if (!pessoaFisicaDAO.cpfExiste(cpf)) {
78             pessoaFisicaDAO.incluir(pessoaFisica);
79             System.out.println("Pessoa fisica incluída com sucesso!");
80         } else {
81             System.out.println("Erro: CPF já existe no banco de dados!");
82         }
83     } else if (tipo == 2) {
84         System.out.println("Nome: ");
85         String nome = scanner.nextLine();
86         System.out.println("Endereço: ");
87         String endereco = scanner.nextLine();
88         System.out.println("Telefone: ");
89         String telefone = scanner.nextLine();
90         System.out.println("Email: ");
91         String email = scanner.nextLine();
92         System.out.println("CNPJ: ");
93         String cnpj = scanner.nextLine();
94
95         PessoaJuridica pessoaJuridica = new PessoaJuridica(0, nome, endereco, "", "", telefone, email, cnpj);
96         if (!pessoaJuridicaDAO.cnpjExiste(cnpj)) {
97             pessoaJuridicaDAO.incluir(pessoaJuridica);
98             System.out.println("Pessoa juridica incluída com sucesso!");
99         } else {
100             System.out.println("Erro: CNPJ já existe no banco de dados!");
101         }
102     }
```

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

CadastroBD
JRE System Library [JavaSE-21]
src
cadastrobd
CadastroBDTeste.java
CadastroBDTeste2.java
Main.java
cadastrobd.model
Pessoa.java
PessoaFisica.java
PessoaFisicaDAO.java
PessoaJuridica.java
PessoaJuridicaDAO.java
cadastrobd.model.util
ConectorBD.java
SequenceManager.java
Referenced Libraries
Relatório Primeiro Procedimento.pdf

CadastroBDTeste2.java X

```
101     }  
102     } else {  
103         System.out.println("Tipo inválido!");  
104     }  
105 }  
106  
107 // Método para alterar uma pessoa existente  
108 private static void alterar(Scanner scanner, PessoaFisicaDAO pessoaFisicaDAO, PessoaJuridicaDAO pessoaJuridicaDAO) {  
109     System.out.println("\n--- ALTERAR ---");  
110     System.out.println("Tipo de pessoa (1 - Fisica, 2 - Juridica): ");  
111     int tipo = scanner.nextInt();  
112     scanner.nextLine(); // Limpar o buffer  
113  
114     System.out.println("ID da pessoa: ");  
115     int id = scanner.nextInt();  
116     scanner.nextLine(); // Limpar o buffer  
117  
118     if (tipo == 1) {  
119         PessoaFisica pessoaFisica = pessoaFisicaDAO.getPessoa(id);  
120         if (pessoaFisica != null) {  
121             System.out.println("Dados atuais:");  
122             pessoaFisica.exibir();  
123  
124             System.out.println("Novo nome (ou pressione Enter para manter): ");  
125             String nome = scanner.nextLine();  
126             if (!nome.isEmpty()) pessoaFisica.setNome(nome);  
127  
128             System.out.println("Novo endereço (ou pressione Enter para manter): ");  
129             String endereco = scanner.nextLine();  
130             if (!endereco.isEmpty()) pessoaFisica.setLogradouro(endereco);  
131  
132             System.out.println("Novo telefone (ou pressione Enter para manter): ");  
133             String telefone = scanner.nextLine();  
134             if (!telefone.isEmpty()) pessoaFisica.setTelefone(telefone);  
135  
136             System.out.println("Novo email (ou pressione Enter para manter): ");  
137             String email = scanner.nextLine();  
138             if (!email.isEmpty()) pessoaFisica.setEmail(email);  
139  
140             System.out.println("Novo CPF (ou pressione Enter para manter): ");  
141             String cpf = scanner.nextLine();  
142             if (!cpf.isEmpty() && !pessoaFisicaDAO.cpfExiste(cpf)) {  
143                 pessoaFisica.setCpf(cpf);  
144             } else if (!cpf.isEmpty()) {  
145                 System.out.println("Erro: CPF já existe no banco de dados!");  
146                 return;  
147             }  
148  
149             pessoaFisicaDAO.alterar(pessoaFisica);  
150             System.out.println("Pessoa fisica alterada com sucesso!");  
151         }  
152     }  
153 }
```



```
149     pessoaFisicaDAO.alterar(pessoaFisica);
150     System.out.println("Pessoa fisica alterada com sucesso!");
151 } else {
152     System.out.println("Pessoa fisica não encontrada!");
153 }
154 } else if (tipo == 2) {
155     PessoaJuridica pessoaJuridica = pessoaJuridicaDAO.getPessoa(id);
156     if (pessoaJuridica != null) {
157         System.out.println("Dados atuais:");
158         pessoaJuridica.exibir();
159
160         System.out.println("Novo nome (ou pressione Enter para manter): ");
161         String nome = scanner.nextLine();
162         if (!nome.isEmpty()) pessoaJuridica.setNome(nome);
163
164         System.out.println("Novo endereço (ou pressione Enter para manter): ");
165         String endereco = scanner.nextLine();
166         if (!endereco.isEmpty()) pessoaJuridica.setLogradouro(endereco);
167
168         System.out.println("Novo telefone (ou pressione Enter para manter): ");
169         String telefone = scanner.nextLine();
170         if (!telefone.isEmpty()) pessoaJuridica.setTelefone(telefone);
171
172         System.out.println("Novo email (ou pressione Enter para manter): ");
173         String email = scanner.nextLine();
174         if (!email.isEmpty()) pessoaJuridica.setEmail(email);
175
176         System.out.println("Novo CNPJ (ou pressione Enter para manter): ");
177         String cnpj = scanner.nextLine();
178         if (!cnpj.isEmpty() && !pessoaJuridicaDAO.cnpjExiste(cnpj)) {
179             pessoaJuridica.setCnpj(cnpj);
180         } else if (!cnpj.isEmpty()) {
181             System.out.println("Erro: CNPJ já existe no banco de dados!");
182             return;
183         }
184
185         pessoaJuridicaDAO.alterar(pessoaJuridica);
186         System.out.println("Pessoa juridica alterada com sucesso!");
187     } else {
188         System.out.println("Pessoa juridica não encontrada!");
189     }
190 } else {
191     System.out.println("Tipo inválido!");
192 }
193 }
194
195 // Método para excluir uma pessoa
196 private static void excluir(Scanner scanner, PessoaFisicaDAO pessoaFisicaDAO, PessoaJuridicaDAO pessoaJuridicaDAO) {
197     System.out.println("\n--- EXCLUIR ---");
198     System.out.println("Tipo de pessoa (1 - Fisica, 2 - Juridica): ");
```



Package Explorer

- CadastroBD
 - JRE System Library [JavaSE-21]
 - src
 - cadastrobd
 - CadastroBDTeste.java
 - CadastroBDTeste2.java
 - Main.java
 - cadastrobd.model
 - Pessoa.java
 - PessoaFisica.java
 - PessoaFisicaDAO.java
 - PessoaJuridica.java
 - PessoaJuridicaDAO.java
 - cadastrobd.model.util
 - ConectorBD.java
 - SequenceManager.java
 - Referenced Libraries
 - Relatório Primeiro Procedimento.pdf

CadastroBDTeste2.java

```
199     int tipo = scanner.nextInt();
200     scanner.nextLine(); // Limpar o buffer
201
202     System.out.println("ID da pessoa: ");
203     int id = scanner.nextInt();
204     scanner.nextLine(); // Limpar o buffer
205
206     if (tipo == 1) {
207         PessoaFisica pessoaFisica = pessoaFisicaDAO.getPessoa(id);
208         if (pessoaFisica != null) {
209             pessoaFisicaDAO.excluir(id);
210             System.out.println("Pessoa fisica excluída com sucesso!");
211         } else {
212             System.out.println("Pessoa fisica não encontrada!");
213         }
214     } else if (tipo == 2) {
215         PessoaJuridica pessoaJuridica = pessoaJuridicaDAO.getPessoa(id);
216         if (pessoaJuridica != null) {
217             pessoaJuridicaDAO.excluir(id);
218             System.out.println("Pessoa juridica excluída com sucesso!");
219         } else {
220             System.out.println("Pessoa juridica não encontrada!");
221         }
222     } else {
223         System.out.println("Tipo inválido!");
224     }
225 }
226
227 // Método para obter uma pessoa pelo ID
228 private static void obterPorId(Scanner scanner, PessoaFisicaDAO pessoaFisicaDAO, PessoaJuridicaDAO pessoaJuridicaDAO) {
229     System.out.println("\n--- OBTENHA POR ID ---");
230     System.out.println("Tipo de pessoa (1 - Fisica, 2 - Juridica): ");
231     int tipo = scanner.nextInt();
232     scanner.nextLine(); // Limpar o buffer
233
234     System.out.println("ID da pessoa: ");
235     int id = scanner.nextInt();
236     scanner.nextLine(); // Limpar o buffer
237
238     if (tipo == 1) {
239         PessoaFisica pessoaFisica = pessoaFisicaDAO.getPessoa(id);
240         if (pessoaFisica != null) {
241             pessoaFisica.exibir();
242         } else {
243             System.out.println("Pessoa fisica não encontrada!");
244         }
245     } else if (tipo == 2) {
246         PessoaJuridica pessoaJuridica = pessoaJuridicaDAO.getPessoa(id);
247         if (pessoaJuridica != null) {
248             pessoaJuridica.exibir();
```


File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer x

- CadastroBD
 - JRE System Library [JavaSE-21]
 - src
 - cadastrobd
 - CadastroBDTeste.java
 - CadastroBDTeste2.java
 - Main.java
 - cadastrobd.model
 - Pessoa.java
 - PessoaFisica.java
 - PessoaFisicaDAO.java
 - PessoaJuridica.java
 - PessoaJuridicaDAO.java
 - cadastrobd.model.util
 - ConectorBD.java
 - SequenceManager.java
 - Referenced Libraries
 - Relatório Primeiro Procedimento.pdf

CadastroBDTeste2.java x

```
242     } else {
243         System.out.println("Pessoa física não encontrada!");
244     }
245     } else if (tipo == 2) {
246         PessoaJuridica pessoaJuridica = pessoaJuridicaDAO.getPessoa(id);
247         if (pessoaJuridica != null) {
248             pessoaJuridica.exibir();
249         } else {
250             System.out.println("Pessoa jurídica não encontrada!");
251         }
252     } else {
253         System.out.println("Tipo inválido!");
254     }
255 }
256
257 // Método para listar todas as pessoas
258 private static void listarTodos(Scanner scanner, PessoaFisicaDAO pessoaFisicaDAO, PessoaJuridicaDAO pessoaJuridicaDAO) {
259     System.out.println("\n--- LISTAR TODOS ---");
260     System.out.println("Tipo de pessoa (1 - Física, 2 - Jurídica): ");
261     int tipo = scanner.nextInt();
262     scanner.nextLine(); // Limpar o buffer
263
264     if (tipo == 1) {
265         System.out.println("Pessoas Físicas:");
266         for (PessoaFisica pf : pessoaFisicaDAO.getPessoas()) {
267             pf.exibir();
268             System.out.println("-----");
269         }
270     } else if (tipo == 2) {
271         System.out.println("Pessoas Jurídicas:");
272         for (PessoaJuridica pj : pessoaJuridicaDAO.getPessoas()) {
273             pj.exibir();
274             System.out.println("-----");
275         }
276     } else {
277         System.out.println("Tipo inválido!");
278     }
279 }
280 }
281
```

Problems Javadoc Declaration Console x Gradle Tasks Gradle Executions Install Java 24 Support

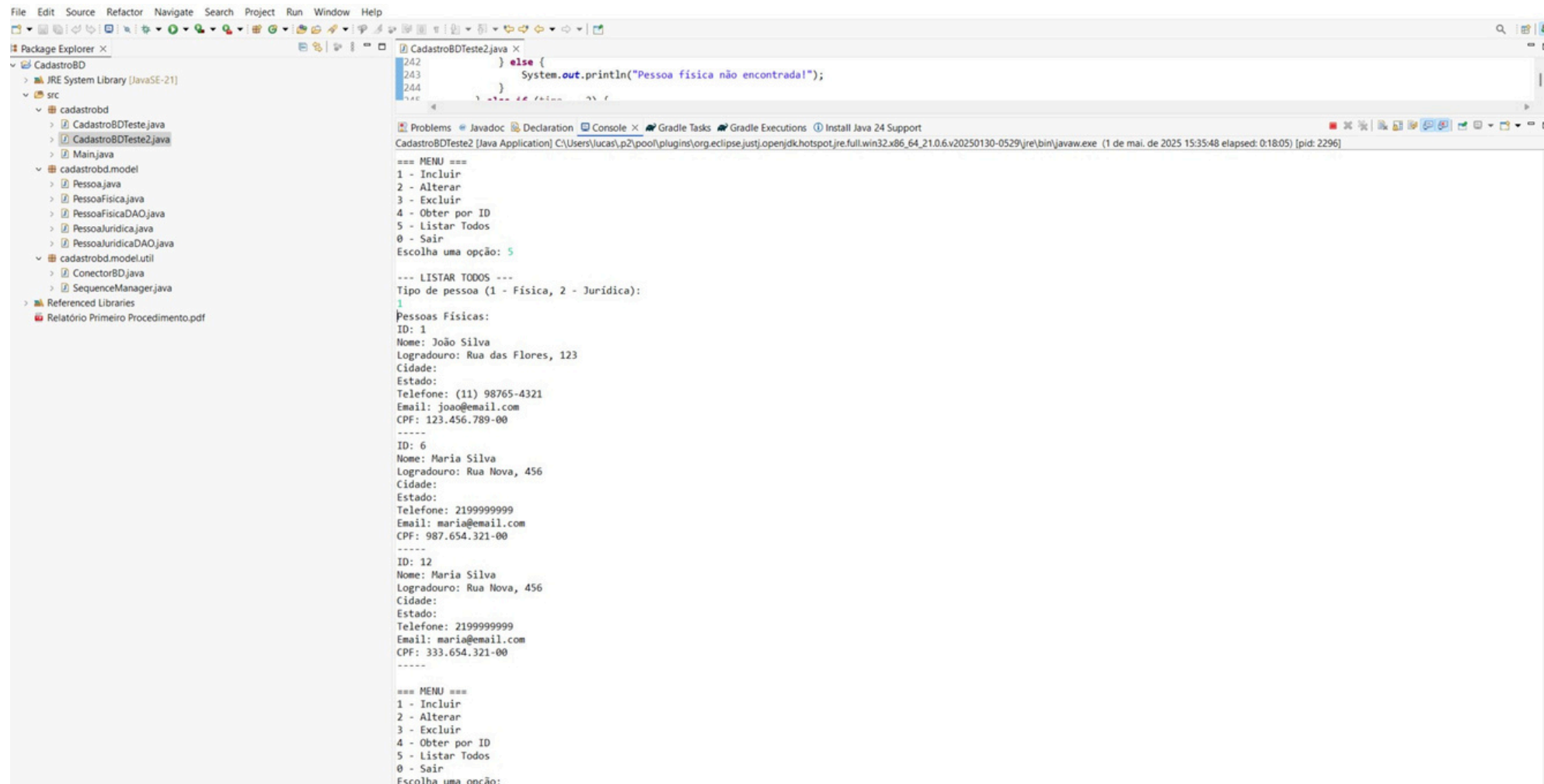
CadastroBDTeste2 [Java Application] C:\Users\lucas\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.6.v20250130-0529\jre\bin\javaw.exe (1 de mai. de 2025 15:35:48 elapsed: 0:09:14) [pid: 2296]

```
=== MENU ===
1 - Incluir
2 - Alterar
3 - Excluir
4 - Obter por ID
5 - Listar Todos
0 - Sair
Escolha uma opção: 5

--- LISTAR TODOS ---
Tipo de pessoa (1 - Física, 2 - Jurídica):
```

2º Procedimento | Alimentando a Base

Códigos usados neste roteiro



The screenshot displays the Eclipse IDE interface. On the left, the Package Explorer shows the project structure: 'CadastroBD' contains 'JRE System Library [JavaSE-21]', 'src' (with 'CadastroBDTeste.java' and 'Main.java'), 'cadastrobd' (with 'CadastroBDTeste2.java'), 'cadastrobd.model' (with 'Pessoa.java', 'PessoaFisica.java', 'PessoaFisicaDAO.java', 'PessoaJuridica.java', and 'PessoaJuridicaDAO.java'), and 'cadastrobd.model.util' (with 'ConectorBD.java' and 'SequenceManager.java').

The main editor shows the code for 'CadastroBDTeste2.java', with lines 242, 243, and 244 visible. Line 243 contains the statement: `System.out.println("Pessoa fisica não encontrada!");`.

The bottom console window shows the output of the application. It starts with a menu: `=== MENU ===`, followed by options 1 to 5 and 0. Option 1 is selected, leading to a list of physical persons. The output shows details for two persons: ID 1 (João Silva) and ID 6 (Maria Silva). The output is as follows:

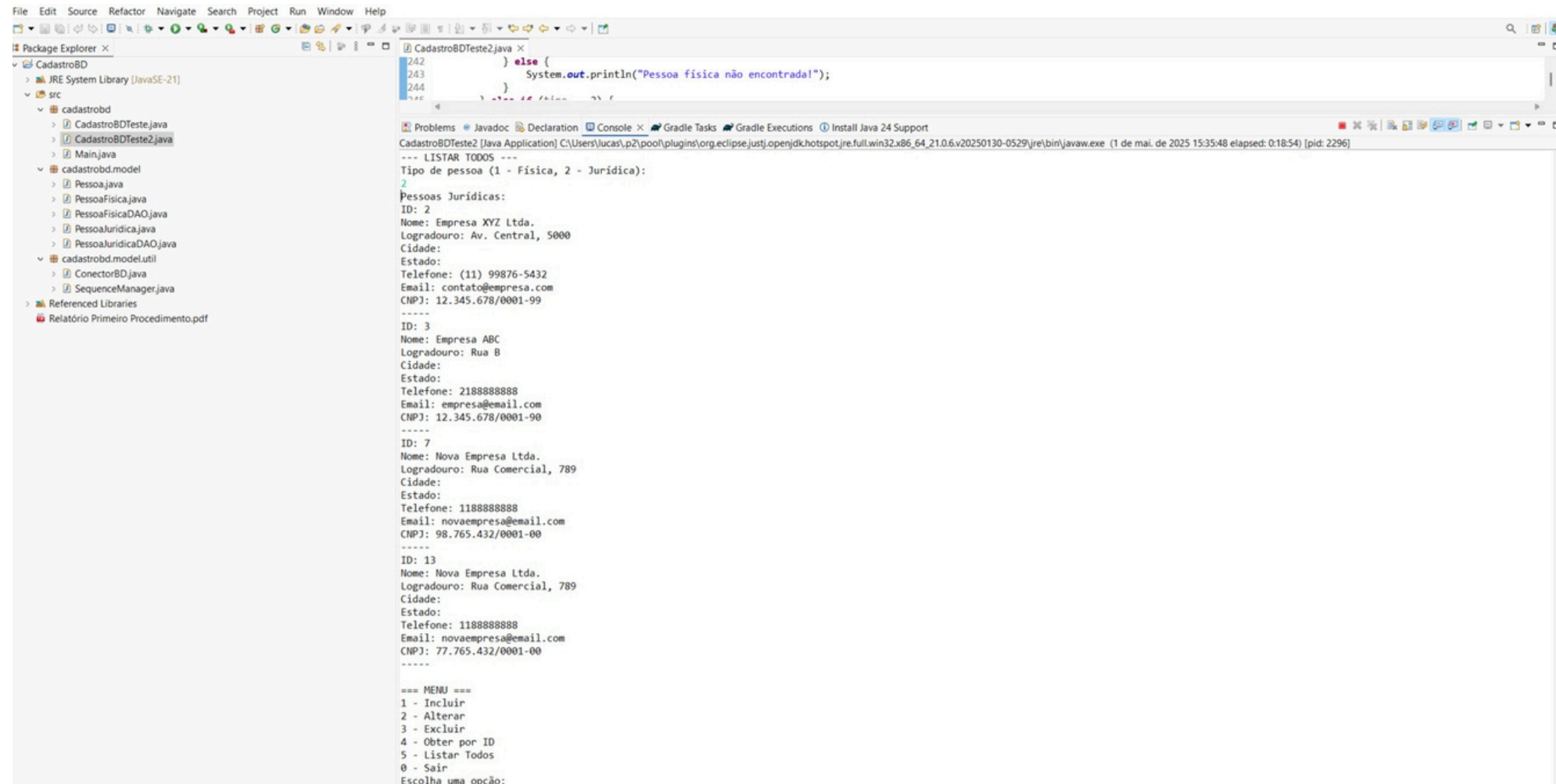
```
=== MENU ===
1 - Incluir
2 - Alterar
3 - Excluir
4 - Obter por ID
5 - Listar Todos
0 - Sair
Escolha uma opção: 5

--- LISTAR TODOS ---
Tipo de pessoa (1 - Física, 2 - Jurídica):
1
Pessoas Físicas:
ID: 1
Nome: João Silva
Logradouro: Rua das Flores, 123
Cidade:
Estado:
Telefone: (11) 98765-4321
Email: joao@email.com
CPF: 123.456.789-00
-----
ID: 6
Nome: Maria Silva
Logradouro: Rua Nova, 456
Cidade:
Estado:
Telefone: 2199999999
Email: maria@email.com
CPF: 987.654.321-00
-----
ID: 12
Nome: Maria Silva
Logradouro: Rua Nova, 456
Cidade:
Estado:
Telefone: 2199999999
Email: maria@email.com
CPF: 333.654.321-00
-----

=== MENU ===
1 - Incluir
2 - Alterar
3 - Excluir
4 - Obter por ID
5 - Listar Todos
0 - Sair
Escolha uma opção:
```

2º Procedimento | Alimentando a Base

Códigos usados neste roteiro



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays the project structure for 'CadastroBD', including 'src', 'cadastrobd', 'cadastrobd.model', and 'cadastrobd.model.util'. The main editor shows the 'CadastroBDTeste2.java' file with a snippet of Java code. The console window at the bottom displays the output of the application, showing a menu and data for three legal entities (Pessoas Jurídicas).

```
242     } else {  
243         System.out.println("Pessoa física não encontrada!");  
244     }  
245 }
```

Problems Javadoc Declaration Console X Gradle Tasks Gradle Executions Install Java 24 Support
CadastroBDTeste2 [Java Application] C:\Users\lucas\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.6.v20250130-0529\jre\bin\javaw.exe (1 de mai. de 2025 15:35:48 elapsed: 0:18:54) [pid: 2296]
--- LISTAR TODOS ---
Tipo de pessoa (1 - Física, 2 - Jurídica):
2
Pessoas Jurídicas:
ID: 2
Nome: Empresa XYZ Ltda.
Logradouro: Av. Central, 5000
Cidade:
Estado:
Telefone: (11) 99876-5432
Email: contato@empresa.com
CNPJ: 12.345.678/0001-99

ID: 3
Nome: Empresa ABC
Logradouro: Rua B
Cidade:
Estado:
Telefone: 2188888888
Email: empresa@email.com
CNPJ: 12.345.678/0001-90

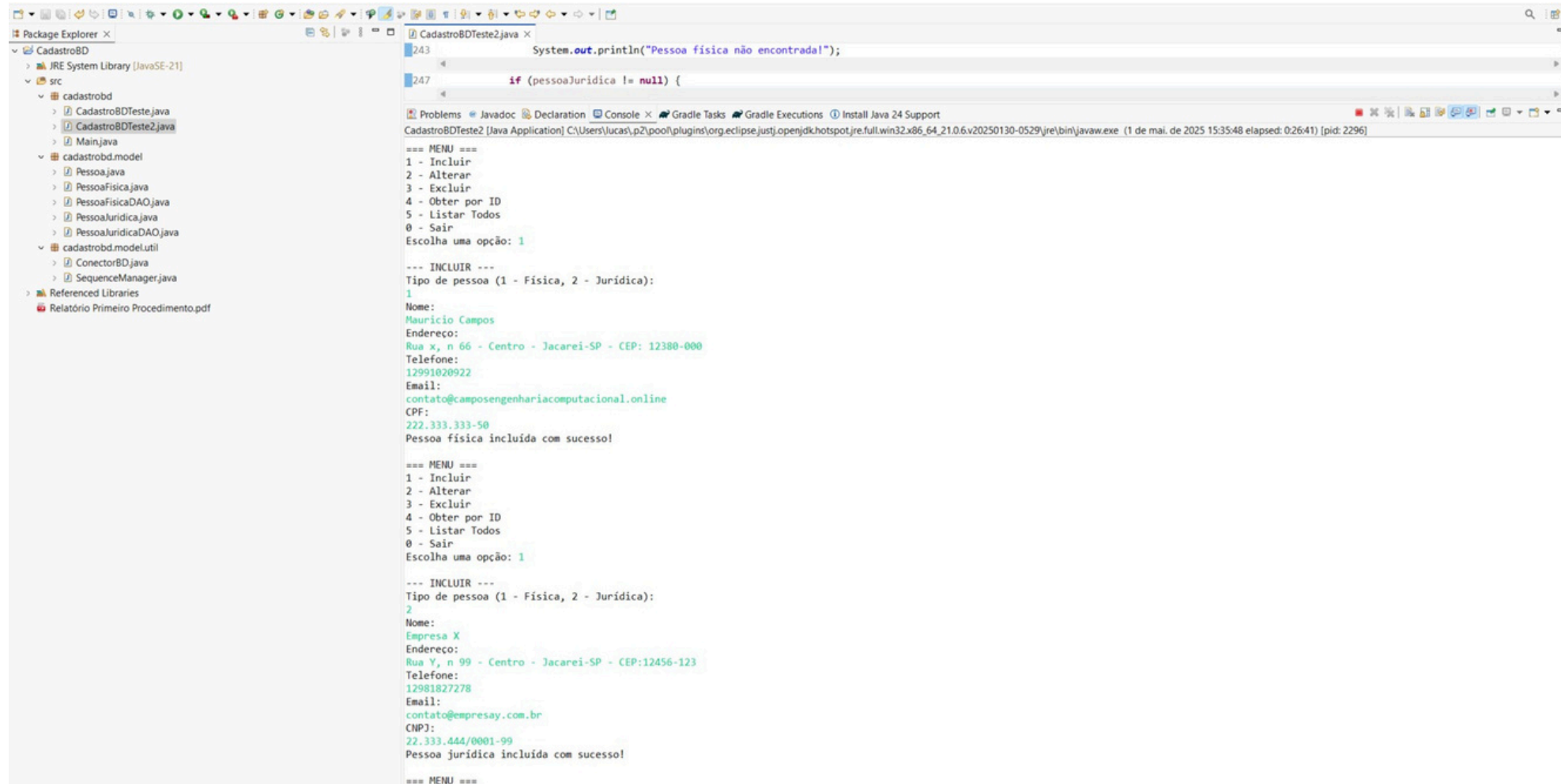
ID: 7
Nome: Nova Empresa Ltda.
Logradouro: Rua Comercial, 789
Cidade:
Estado:
Telefone: 1188888888
Email: novaempresa@email.com
CNPJ: 98.765.432/0001-00

ID: 13
Nome: Nova Empresa Ltda.
Logradouro: Rua Comercial, 789
Cidade:
Estado:
Telefone: 1188888888
Email: novaempresa@email.com
CNPJ: 77.765.432/0001-00

=== MENU ===
1 - Incluir
2 - Alterar
3 - Excluir
4 - Obter por ID
5 - Listar Todos
0 - Sair
Escolha uma opção:

2º Procedimento | Alimentando a Base

Códigos usados neste roteiro



The screenshot displays the Eclipse IDE interface. On the left, the Package Explorer shows the project structure for 'CadastroBD', including source files like 'CadastroBDTeste2.java' and 'Main.java'. The main editor window shows the code for 'CadastroBDTeste2.java', with lines 243 and 247 highlighted. The console window at the bottom shows the output of the application, which is a menu-driven program for managing a database. The output shows the program running, displaying a menu with options 1 through 5, and the user selecting option 1. The program then prompts for the type of person (1 for Física, 2 for Jurídica) and the user enters '1'. It then prompts for the name, address, phone number, email, and CPF of the physical person, and finally displays the message 'Pessoa física incluída com sucesso!'.

```
243 System.out.println("Pessoa física não encontrada!");
247 if (pessoaJuridica != null) {

Problems Javadoc Declaration Console × Gradle Tasks Gradle Executions Install Java 24 Support
CadastroBDTeste2 [Java Application] C:\Users\lucas.p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.6.v20250130-0529\jre\bin\javaw.exe (1 de mai. de 2025 15:35:48 elapsed: 0:26:41) [pid: 2296]

=== MENU ===
1 - Incluir
2 - Alterar
3 - Excluir
4 - Obter por ID
5 - Listar Todos
0 - Sair
Escolha uma opção: 1

--- INCLUIR ---
Tipo de pessoa (1 - Física, 2 - Jurídica):
1
Nome:
Mauricio Campos
Endereço:
Rua X, n 66 - Centro - Jacarei-SP - CEP: 12380-000
Telefone:
12991020922
Email:
contato@camposengenhariacomputacional.online
CPF:
222.333.333-50
Pessoa física incluída com sucesso!

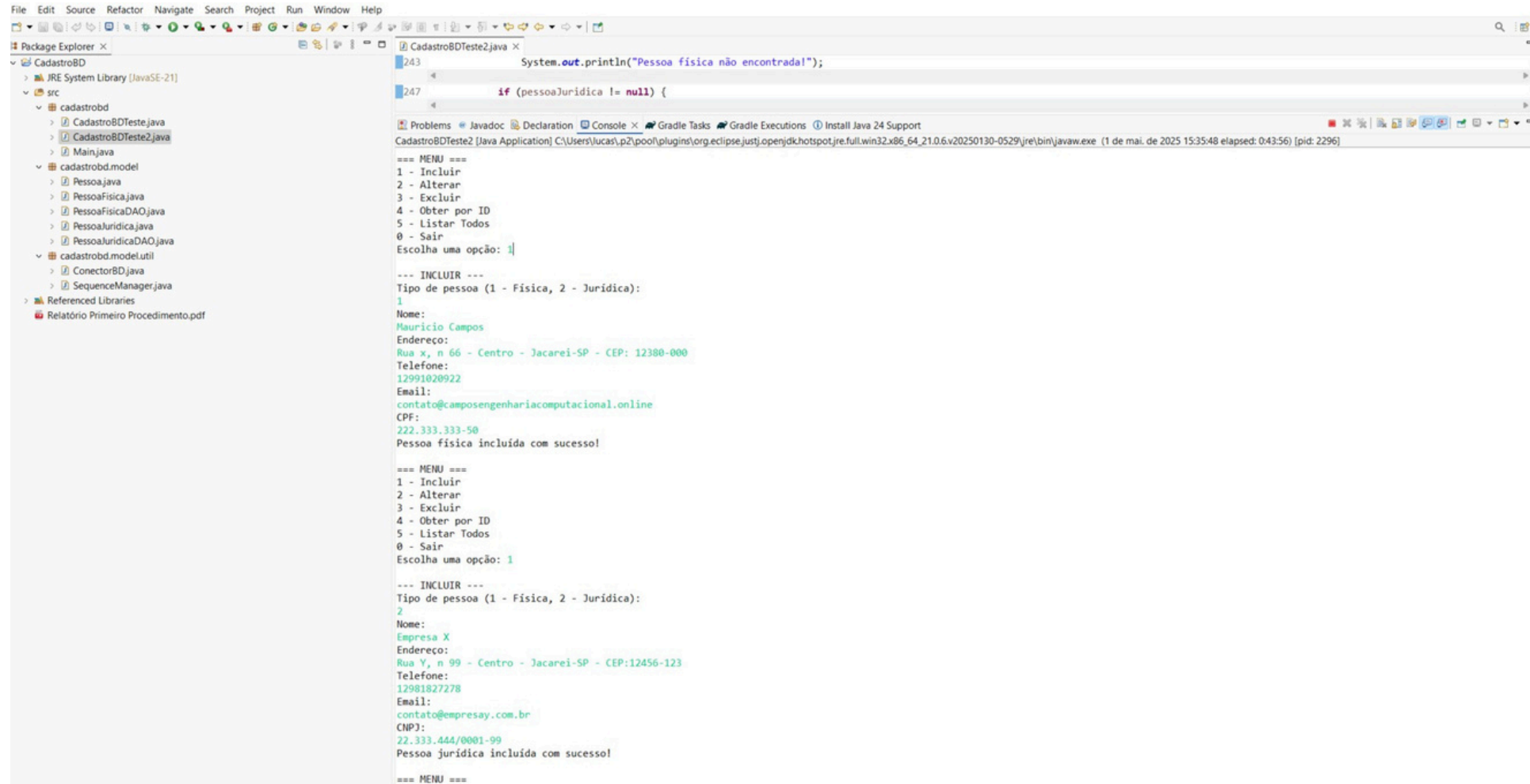
=== MENU ===
1 - Incluir
2 - Alterar
3 - Excluir
4 - Obter por ID
5 - Listar Todos
0 - Sair
Escolha uma opção: 1

--- INCLUIR ---
Tipo de pessoa (1 - Física, 2 - Jurídica):
2
Nome:
Empresa X
Endereço:
Rua Y, n 99 - Centro - Jacarei-SP - CEP:12456-123
Telefone:
12901827278
Email:
contato@empresay.com.br
CNPJ:
22.333.444/0001-99
Pessoa jurídica incluída com sucesso!

=== MENU ===
```

2º Procedimento | Alimentando a Base

Códigos usados neste roteiro



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays the project structure for 'CadastroBD', including 'src', 'cadastrobd', 'cadastrobd.model', and 'cadastrobd.model.util'. The main editor shows the 'CadastroBDTeste2.java' file with the following code:

```
243 System.out.println("Pessoa física não encontrada!");
247 if (pessoaJuridica != null) {
```

Below the editor, the Console window shows the output of the application:

```
=== MENU ===
1 - Incluir
2 - Alterar
3 - Excluir
4 - Obter por ID
5 - Listar Todos
0 - Sair
Escolha uma opção: 1

--- INCLUIR ---
Tipo de pessoa (1 - Física, 2 - Jurídica):
1
Nome:
Mauricio Campos
Endereço:
Rua x, n 66 - Centro - Jacarei-SP - CEP: 12380-000
Telefone:
12991020922
Email:
contato@camposengenhariacomputacional.online
CPF:
222.333.333-50
Pessoa física incluída com sucesso!

=== MENU ===
1 - Incluir
2 - Alterar
3 - Excluir
4 - Obter por ID
5 - Listar Todos
0 - Sair
Escolha uma opção: 1

--- INCLUIR ---
Tipo de pessoa (1 - Física, 2 - Jurídica):
2
Nome:
Empresa X
Endereço:
Rua Y, n 99 - Centro - Jacarei-SP - CEP:12456-123
Telefone:
12981827278
Email:
contato@empresay.com.br
CNPJ:
22.333.444/0001-99
Pessoa jurídica incluída com sucesso!

=== MENU ===
```


2º Procedimento | Alimentando a Base

Códigos usados neste roteiro

← → ↻ 127.0.0.1:49765/browser/

pgAdmin File ▾ Object ▾ Tools ▾ Help ▾

Welcome loja/postgres@PostgreSQL 17* x

loja/postgres@PostgreSQL 17

Query Query History

```
1 SELECT p.id_pessoa, p.nome, p.endereco, p.telefone, p.email, pf.cpf
2 FROM pessoa p
3 JOIN pessoa_fisica pf ON p.id_pessoa = pf.id_pessoa;
4
```

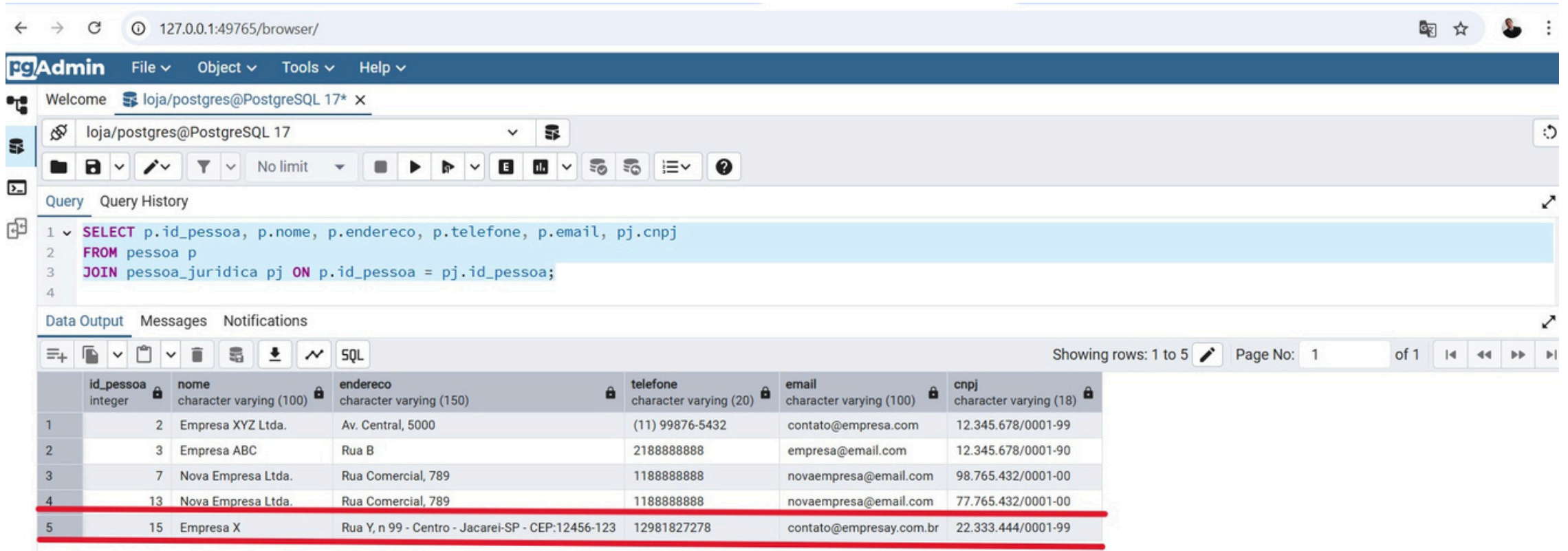
Data Output Messages Notifications

Showing rows: 1 to 4 Page No: 1 of 1

	id_pessoa integer	nome character varying (100)	endereco character varying (150)	telefone character varying (20)	email character varying (100)	cpf character varying (14)
1	1	João Silva	Rua das Flores, 123	(11) 98765-4321	joao@email.com	123.456.789-00
2	6	Maria Silva	Rua Nova, 456	2199999999	maria@email.com	987.654.321-00
3	12	Maria Silva	Rua Nova, 456	2199999999	maria@email.com	333.654.321-00
4	14	Mauricio Campos	Rua x, n 66 - Centro - Jacarei-SP - CEP: 12380-000	12991020922	contato@camposengenhariacomputacional.online	222.333.333-50

2º Procedimento | Alimentando a Base

Códigos usados neste roteiro



The screenshot shows the pgAdmin web interface. The browser address bar displays '127.0.0.1:49765/browser/'. The interface includes a menu bar (File, Object, Tools, Help) and a toolbar with various icons. The main area is divided into sections: 'Query' and 'Query History' at the top, followed by a SQL query editor. The query is as follows:

```
1 SELECT p.id_pessoa, p.nome, p.endereco, p.telefone, p.email, pj.cnpj
2 FROM pessoa p
3 JOIN pessoa_juridica pj ON p.id_pessoa = pj.id_pessoa;
4
```

Below the query editor is the 'Data Output' section, which displays the results of the query in a table. The table has 7 columns: id_pessoa, nome, endereco, telefone, email, and cnpj. The results are shown in a grid with 5 rows. The first row is highlighted in red, and the last row is also highlighted in red.

	id_pessoa integer	nome character varying (100)	endereco character varying (150)	telefone character varying (20)	email character varying (100)	cnpj character varying (18)
1	2	Empresa XYZ Ltda.	Av. Central, 5000	(11) 99876-5432	contato@empresa.com	12.345.678/0001-99
2	3	Empresa ABC	Rua B	2188888888	empresa@email.com	12.345.678/0001-90
3	7	Nova Empresa Ltda.	Rua Comercial, 789	1188888888	novaempresa@email.com	98.765.432/0001-00
4	13	Nova Empresa Ltda.	Rua Comercial, 789	1188888888	novaempresa@email.com	77.765.432/0001-00
5	15	Empresa X	Rua Y, n 99 - Centro - Jacarel-SP - CEP:12456-123	12981827278	contato@empresay.com.br	22.333.444/0001-99

Análise e conclusão:

Quais as diferenças entre a persistência em arquivo e a persistência em banco de dados?

A persistência em arquivo e a persistência em banco de dados são duas formas de salvar dados de forma durável, mas possuem diferenças importantes em termos de organização, desempenho, escalabilidade e segurança.

Como o uso de operador lambda simplificou a impressão dos valores contidos nas entidades, nas versões mais recentes do Java?

O uso de operadores lambda no Java (introduzidos a partir do Java 8) trouxe uma forma muito mais concisa, expressiva e moderna de trabalhar com coleções, inclusive para imprimir valores de entidades.

Análise e conclusão:

Por que métodos acionados diretamente pelo método main, sem o uso de um objeto, precisam ser marcados como static?

Métodos chamados diretamente a partir do método main precisam ser static porque o main também é um método estático e métodos estáticos só podem acessar diretamente outros métodos ou atributos que também sejam estáticos.