



Estácio

CAMPUS POLO AUSTIN - NOVA IGUAÇU – RJ

DESENVOLVIMENTO FULL STACK

Nível 3: Back-end Sem Banco Não Tem

RPG0016 9001

2024/2

SALOMÃO ISAAC CARVALHO GARCIA

Implementação de Cadastro e Manipulação de Dados em Modo Texto

Nova Iguaçu

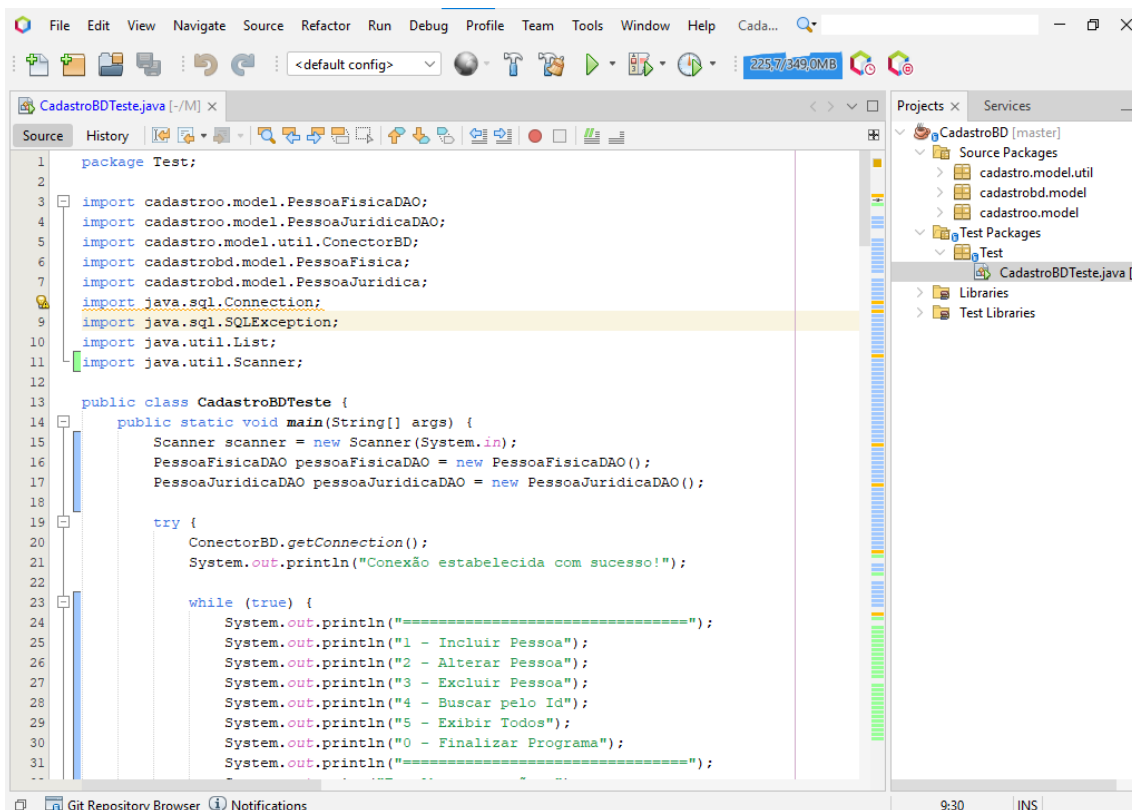
15/08/2024

Implementação de Cadastro e Manipulação de Dados em Modo Texto

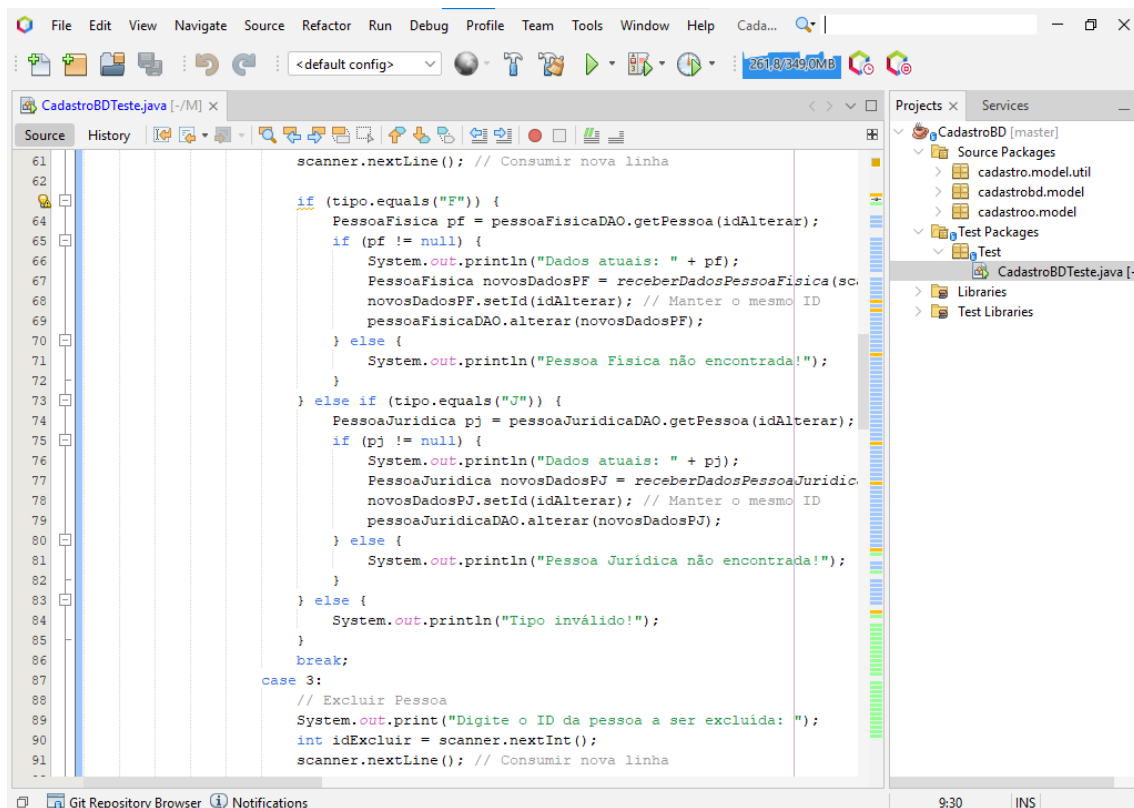
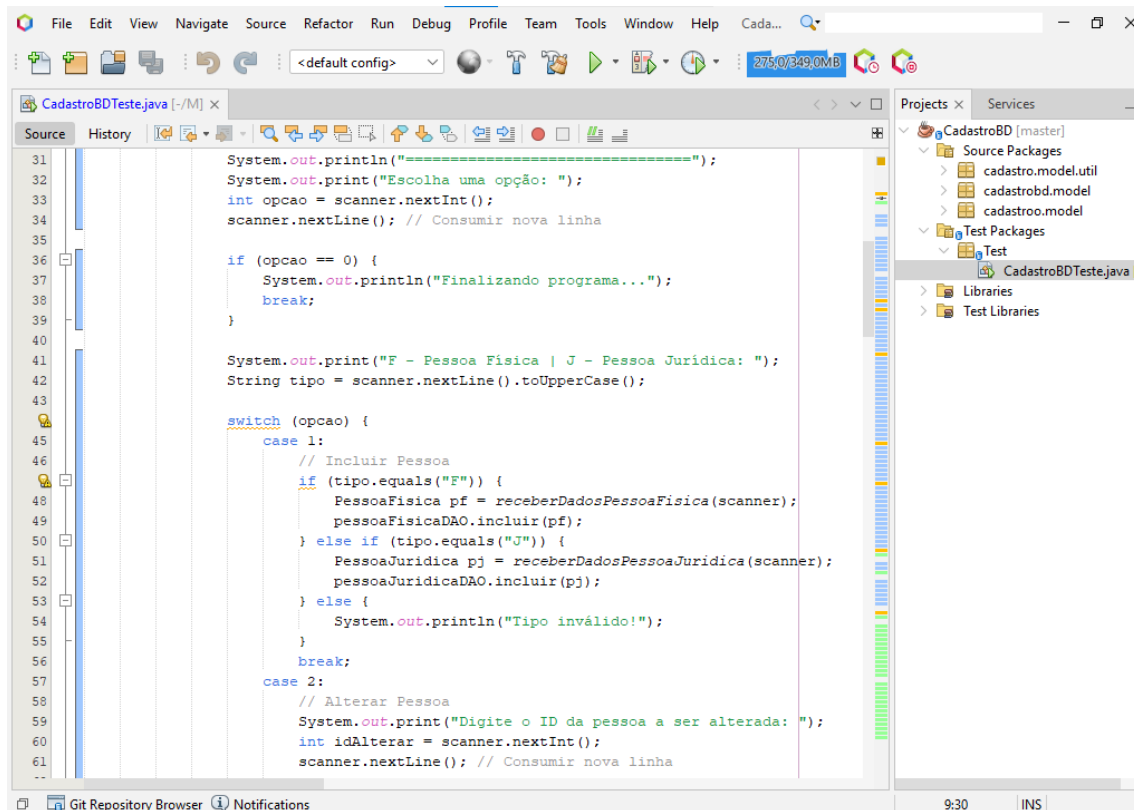
Objetivo da Prática: O objetivo desta prática é implementar um sistema de cadastro de pessoas físicas e jurídicas em modo texto, utilizando operações básicas de CRUD (Create, Read, Update, Delete) no banco de dados. O sistema deverá permitir ao usuário incluir, alterar, excluir, e exibir registros tanto para pessoas físicas quanto para jurídicas através de uma interface de linha de comando. A prática inclui a criação de métodos para interação com o banco de dados usando as classes DAO (Data Access Object), tratamento de exceções e verificação dos resultados das operações tanto no SQL Server quanto na interface de gerenciamento do banco de dados.

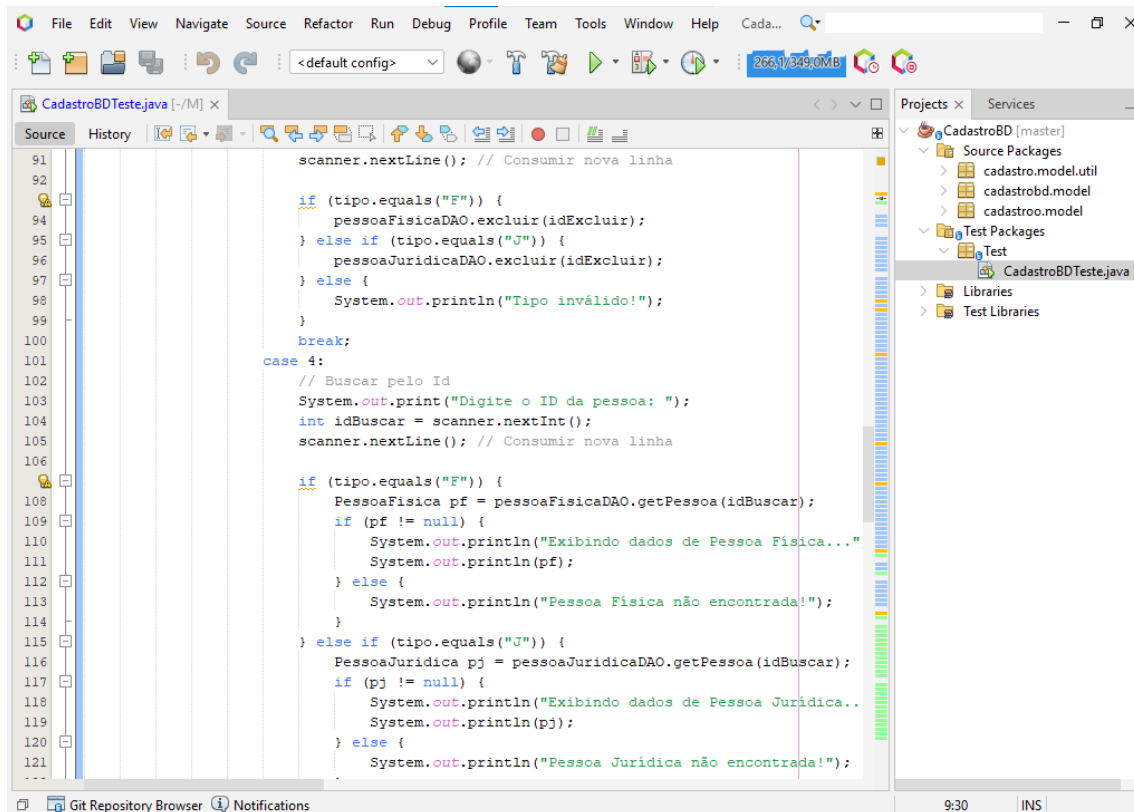
<https://github.com/SaloGarcia/nivel03mundo03.git>

CÓDIGOS

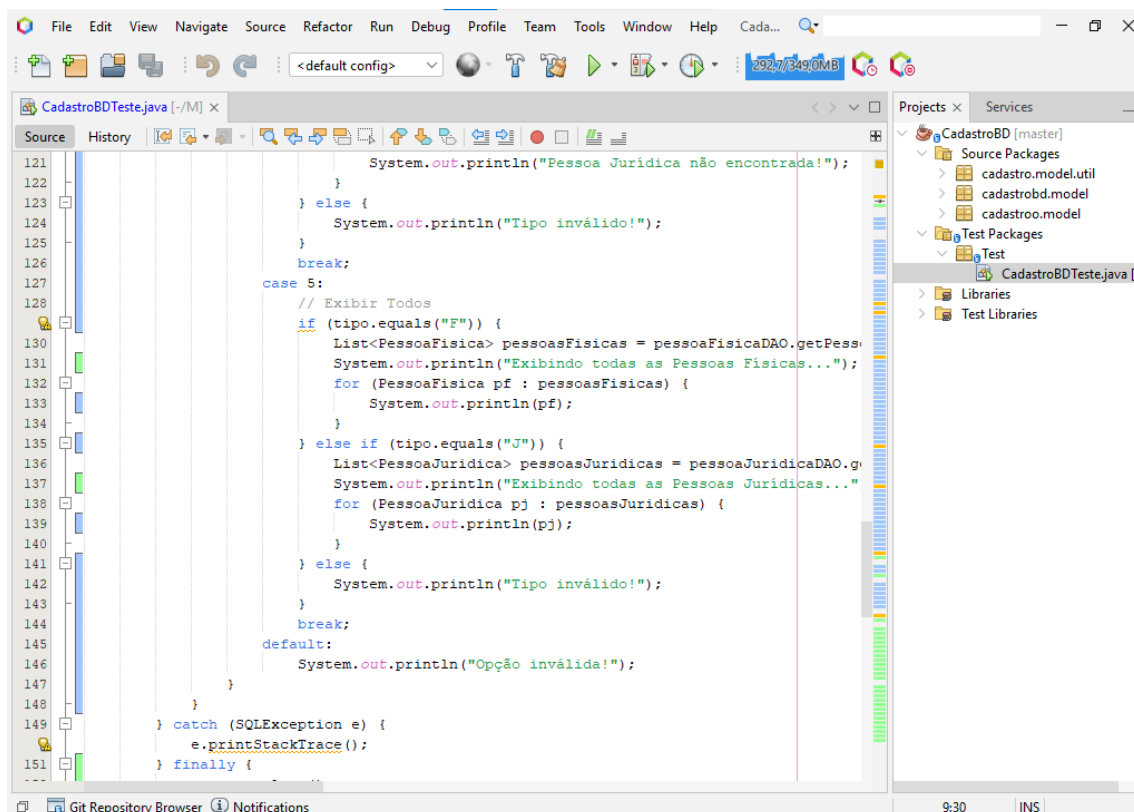


```
1 package Test;
2
3 import cadastro.model.PessoaFisicaDAO;
4 import cadastro.model.PessoaJuridicaDAO;
5 import cadastro.model.util.ConectorBD;
6 import cadastrobd.model.PessoaFisica;
7 import cadastrobd.model.PessoaJuridica;
8 import java.sql.Connection;
9 import java.sql.SQLException;
10 import java.util.List;
11 import java.util.Scanner;
12
13 public class CadastroBDTeste {
14     public static void main(String[] args) {
15         Scanner scanner = new Scanner(System.in);
16         PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO();
17         PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO();
18
19         try {
20             ConectorBD.getConnection();
21             System.out.println("Conexão estabelecida com sucesso!");
22
23             while (true) {
24                 System.out.println("=====");
25                 System.out.println("1 - Incluir Pessoa");
26                 System.out.println("2 - Alterar Pessoa");
27                 System.out.println("3 - Excluir Pessoa");
28                 System.out.println("4 - Buscar pelo Id");
29                 System.out.println("5 - Exibir Todos");
30                 System.out.println("0 - Finalizar Programa");
31                 System.out.println("=====");
```





```
91 scanner.nextLine(); // Consumir nova linha
92
93
94 if (tipo.equals("F")) {
95     pessoaFisicaDAO.excluir(idExcluir);
96 } else if (tipo.equals("J")) {
97     pessoaJuridicaDAO.excluir(idExcluir);
98 } else {
99     System.out.println("Tipo inválido!");
100 }
101 break;
102 case 4:
103     // Buscar pelo Id
104     System.out.print("Digite o ID da pessoa: ");
105     int idBuscar = scanner.nextInt();
106     scanner.nextLine(); // Consumir nova linha
107
108     if (tipo.equals("F")) {
109         PessoaFisica pf = pessoaFisicaDAO.getPessoa(idBuscar);
110         if (pf != null) {
111             System.out.println("Exibindo dados de Pessoa Fisica...");
112             System.out.println(pf);
113         } else {
114             System.out.println("Pessoa Fisica não encontrada!");
115         }
116     } else if (tipo.equals("J")) {
117         PessoaJuridica pj = pessoaJuridicaDAO.getPessoa(idBuscar);
118         if (pj != null) {
119             System.out.println("Exibindo dados de Pessoa Juridica..");
120             System.out.println(pj);
121         } else {
122             System.out.println("Pessoa Juridica não encontrada!");
123         }
124     }
125     break;
126 case 5:
127     // Exibir Todos
128     if (tipo.equals("F")) {
129         List<PessoaFisica> pessoasFisicas = pessoaFisicaDAO.getPess;
130         System.out.println("Exibindo todas as Pessoas Fisicas...");
131         for (PessoaFisica pf : pessoasFisicas) {
132             System.out.println(pf);
133         }
134     } else if (tipo.equals("J")) {
135         List<PessoaJuridica> pessoasJuridicas = pessoaJuridicaDAO.g;
136         System.out.println("Exibindo todas as Pessoas Juridicas...");
137         for (PessoaJuridica pj : pessoasJuridicas) {
138             System.out.println(pj);
139         }
140     } else {
141         System.out.println("Tipo inválido!");
142     }
143     break;
144 default:
145     System.out.println("Opção inválida!");
146 }
147
148 }
149 catch (SQLException e) {
150     e.printStackTrace();
151 } finally {
152 }
```



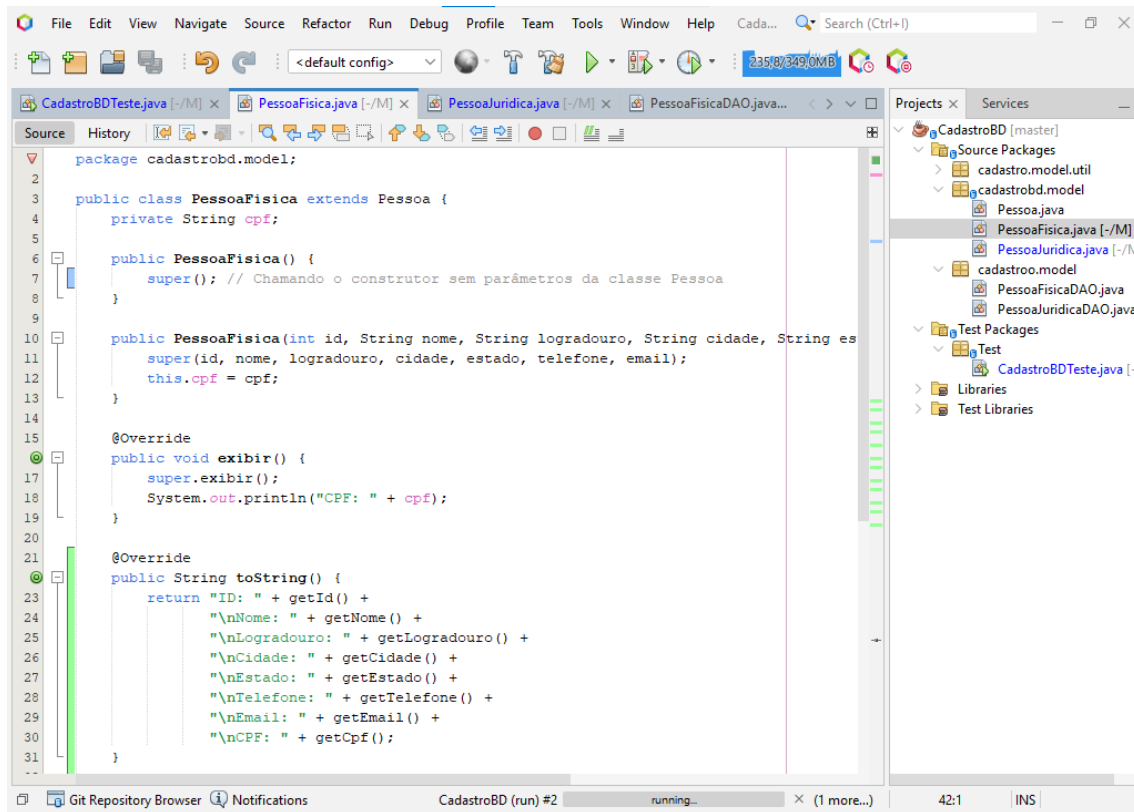
```
121 System.out.println("Pessoa Juridica não encontrada!");
122 }
123 } else {
124     System.out.println("Tipo inválido!");
125 }
126 break;
127 case 5:
128     // Exibir Todos
129     if (tipo.equals("F")) {
130         List<PessoaFisica> pessoasFisicas = pessoaFisicaDAO.getPess;
131         System.out.println("Exibindo todas as Pessoas Fisicas...");
132         for (PessoaFisica pf : pessoasFisicas) {
133             System.out.println(pf);
134         }
135     } else if (tipo.equals("J")) {
136         List<PessoaJuridica> pessoasJuridicas = pessoaJuridicaDAO.g;
137         System.out.println("Exibindo todas as Pessoas Juridicas...");
138         for (PessoaJuridica pj : pessoasJuridicas) {
139             System.out.println(pj);
140         }
141     } else {
142         System.out.println("Tipo inválido!");
143     }
144     break;
145 default:
146     System.out.println("Opção inválida!");
147 }
148 }
149 catch (SQLException e) {
150     e.printStackTrace();
151 } finally {
152 }
```

```
151         } finally {
152             scanner.close();
153         }
154     }
155
156     private static PessoaFisica receberDadosPessoaFisica(Scanner scanner) {
157         System.out.print("Nome: ");
158         String nome = scanner.nextLine();
159         System.out.print("Logradouro: ");
160         String logradouro = scanner.nextLine();
161         System.out.print("Cidade: ");
162         String cidade = scanner.nextLine();
163         System.out.print("Estado: ");
164         String estado = scanner.nextLine();
165         System.out.print("Telefone: ");
166         String telefone = scanner.nextLine();
167         System.out.print("Email: ");
168         String email = scanner.nextLine();
169         System.out.print("CPF: ");
170         String cpf = scanner.nextLine();
171
172         return new PessoaFisica(0, nome, logradouro, cidade, estado, telefone, email, cpf);
173     }
174
175     private static PessoaJuridica receberDadosPessoaJuridica(Scanner scanner) {
176         System.out.print("Nome: ");
177         String nome = scanner.nextLine();
178         System.out.print("Logradouro: ");
179         String logradouro = scanner.nextLine();
180         System.out.print("Cidade: ");
181         String cidade = scanner.nextLine();
182         System.out.print("Estado: ");
183         String estado = scanner.nextLine();
184         System.out.print("Telefone: ");
185         String telefone = scanner.nextLine();
186         System.out.print("Email: ");
187         String email = scanner.nextLine();
188         System.out.print("CNPJ: ");
189         String cnpj = scanner.nextLine();
190
191         return new PessoaJuridica(0, nome, logradouro, cidade, estado, telefone, email, cnpj);
192     }
193
194 }
```

Git Repository Browser Notifications 9:30 INS

```
172         return new PessoaJuridica(0, nome, logradouro, cidade, estado, telefone, email, cnpj);
173     }
174
175     private static PessoaJuridica receberDadosPessoaJuridica(Scanner scanner) {
176         System.out.print("Nome: ");
177         String nome = scanner.nextLine();
178         System.out.print("Logradouro: ");
179         String logradouro = scanner.nextLine();
180         System.out.print("Cidade: ");
181         String cidade = scanner.nextLine();
182         System.out.print("Estado: ");
183         String estado = scanner.nextLine();
184         System.out.print("Telefone: ");
185         String telefone = scanner.nextLine();
186         System.out.print("Email: ");
187         String email = scanner.nextLine();
188         System.out.print("CNPJ: ");
189         String cnpj = scanner.nextLine();
190
191         return new PessoaJuridica(0, nome, logradouro, cidade, estado, telefone, email, cnpj);
192     }
193
194 }
```

Git Repository Browser Notifications 9:30 INS



```
package cadastrorbd.model;

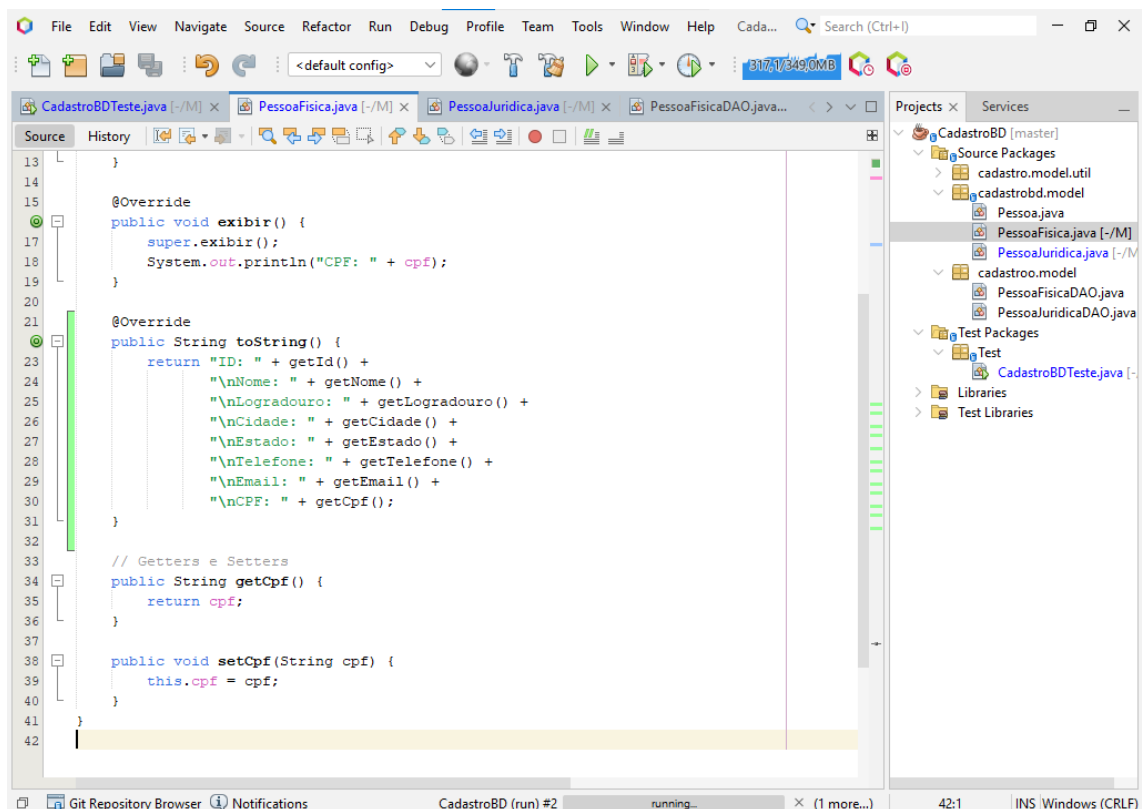
public class PessoaFisica extends Pessoa {
    private String cpf;

    public PessoaFisica() {
        super(); // Chamando o construtor sem parâmetros da classe Pessoa
    }

    public PessoaFisica(int id, String nome, String logradouro, String cidade, String estado, String telefone, String email) {
        super(id, nome, logradouro, cidade, estado, telefone, email);
        this.cpf = cpf;
    }

    @Override
    public void exhibit() {
        super.exibir();
        System.out.println("CPF: " + cpf);
    }

    @Override
    public String toString() {
        return "ID: " + getId() +
            "\nNome: " + getNome() +
            "\nLogradouro: " + getLogradouro() +
            "\nCidade: " + getCidade() +
            "\nEstado: " + getEstado() +
            "\nTelefone: " + getTelefone() +
            "\nEmail: " + getEmail() +
            "\nCPF: " + getCpf();
    }
}
```

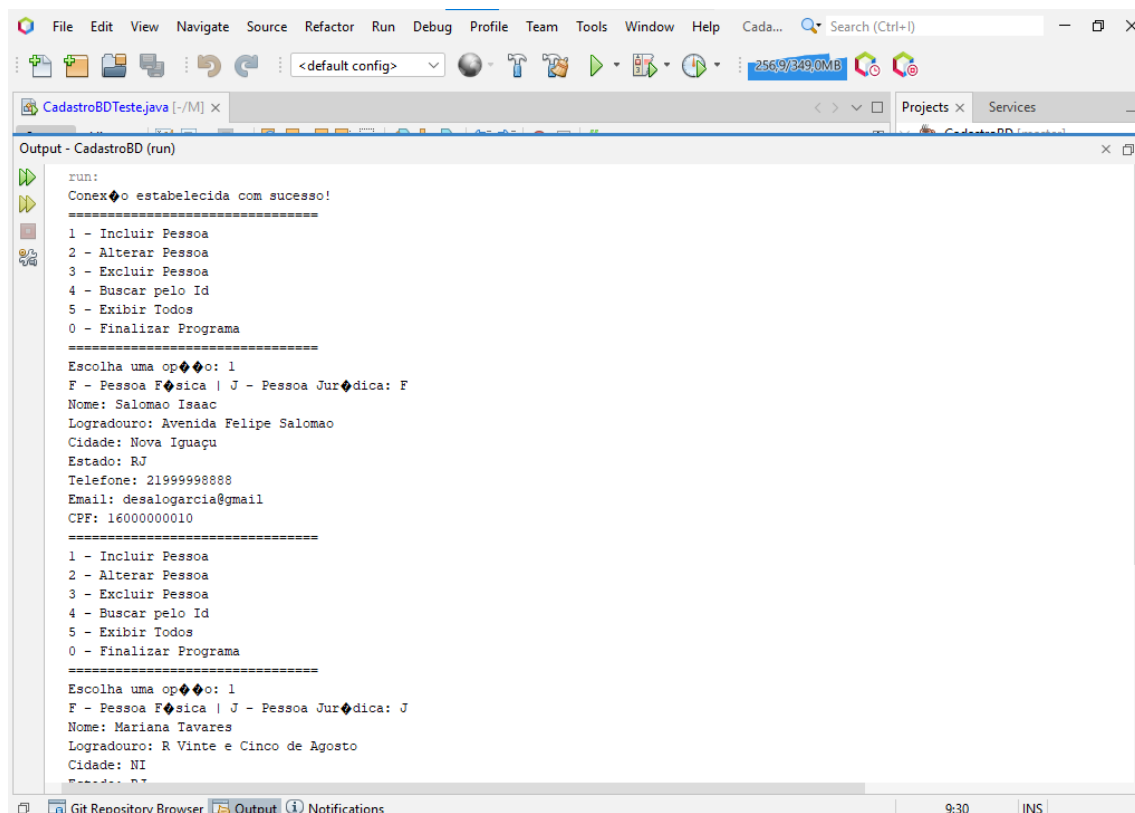
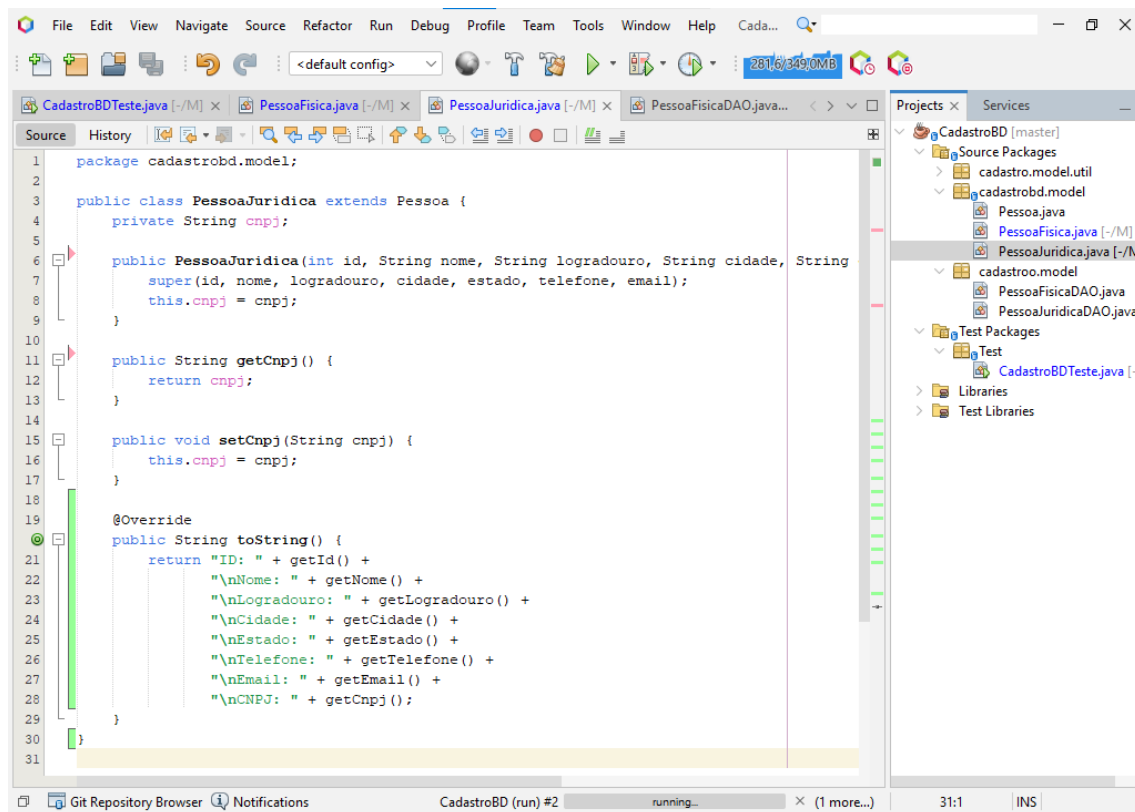


```
    @Override
    public void exhibit() {
        super.exibir();
        System.out.println("CPF: " + cpf);
    }

    @Override
    public String toString() {
        return "ID: " + getId() +
            "\nNome: " + getNome() +
            "\nLogradouro: " + getLogradouro() +
            "\nCidade: " + getCidade() +
            "\nEstado: " + getEstado() +
            "\nTelefone: " + getTelefone() +
            "\nEmail: " + getEmail() +
            "\nCPF: " + getCpf();
    }

    // Getters e Setters
    public String getCpf() {
        return cpf;
    }

    public void setCpf(String cpf) {
        this.cpf = cpf;
    }
}
```



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... Search (Ctrl+I)
<default config> 192.9/349.0MB
CadastroBDTeste.java [~/M] x PessoaFisica.java [~/M] x PessoaJuridica.java [~/M] x PessoaFisicaDAO.java... Projects x Services
Output - CadastroBD (run)
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====
Escolha uma opç o: 2
F - Pessoa F sica | J - Pessoa Jur dica: f
Digite o ID da pessoa a ser alterada: 17
Dados atuais: ID: 17
Nome: Salomao Isaac
Logradouro: Avenida Felipe Salomao
Cidade: Nova Igua u
Estado: RJ
Telefone: 21999998888
Email: desalogarcia@gmail
CPF: 16000000010
Nome: Salomao Isaac
Logradouro: Avenida Felipe Salomao
Cidade: Nova Igua u
Estado: RJ
Telefone: 21999998888
Email: desalogarciajesus@gmail
CPF: 16000000010
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====
Git Repository Browser Output Notifications CadastroBD (run) running... 31:1 INS
```

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... Search (Ctrl+I)
<default config> 198.6/349.0MB
CadastroBDTeste.java [~/M] x PessoaFisica.java [~/M] x PessoaJuridica.java [~/M] x PessoaFisicaDAO.java... Projects x Services
Output - CadastroBD (run)
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====
Escolha uma opç o: 3
F - Pessoa F sica | J - Pessoa Jur dica: f
Digite o ID da pessoa a ser exclu da: 17
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====
Escolha uma opç o: 4
F - Pessoa F sica | J - Pessoa Jur dica: f
Digite o ID da pessoa: 17
Pessoa F sica n o encontrada!
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====
Escolha uma opç o: 5
F - Pessoa F sica | J - Pessoa Jur dica: f
=====
Git Repository Browser Output Notifications CadastroBD (run) running... 31:1 INS
```



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... Search (Ctrl+I)
<default config> 2673/349.0MB
CadastroBDTeste.java [-/M] x PessoaFisica.java [-/M] x PessoaJuridica.java [-/M] x PessoaFisicaDAO.java... Projects x Services
Output - CadastroBD (run)
=====
Escolha uma opção: 5
F - Pessoa Física | J - Pessoa Jurídica: f
Exibindo todas as Pessoas Físicas...
ID: 1
Nome: Joao
Logradouro: Rua 12, casa 3, Quitanda
Cidade: Riacho do Sul
Estado: PA
Telefone: 1111111111
Email: joao@riacho.com
CPF: 1111111111
ID: 3
Nome: Ana Silva
Logradouro: Av. das Flores, 123
Cidade: Fortaleza
Estado: CE
Telefone: 9999999999
Email: ana.silva@exemplo.com
CPF: 98765432100
ID: 4
Nome: Carlos Souza
Logradouro: Rua da Paz, 456
Cidade: Salvador
Estado: BA
Telefone: 9888888888
Email: carlos.souza@exemplo.com
CPF: 12345678901
ID: 15
Nome: Ana Silva
Logradouro: Av. das Flores, 123
Cidade: Fortaleza
Estado: CE
Git Repository Browser Output Notifications CadastroBD (run) running... 31:1 INS
```

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Cada... Search (Ctrl+I)
<default config> 2255/349.0MB
CadastroBDTeste.java [-/M] x PessoaFisica.java [-/M] x PessoaJuridica.java [-/M] x PessoaFisicaDAO.java... Projects x Services
Output - CadastroBD (run)
Logradouro: Av. das Flores, 123
Cidade: Fortaleza
Estado: CE
Telefone: 9999999999
Email: ana.silva@exemplo.com
CPF: 98765432100
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====
Escolha uma opção: 5
F - Pessoa Física | J - Pessoa Jurídica: j
Exibindo todas as Pessoas Jurídicas...
ID: 2
Nome: JJC
Logradouro: Rua 11, Centro
Cidade: Riacho do Norte
Estado: PA
Telefone: 12121212121
Email: jjc@riacho.com
CNPJ: 2222222222222
ID: 5
Nome: Empresa Exemplo
Logradouro: Av. Principal, 500
Cidade: São Paulo
Estado: SP
Telefone: 11223344556
Email: contato@empresa.com
CNPJ: 12345678000195
Git Repository Browser Output Notifications CadastroBD (run) running... 31:1 INS
```

SQLQuery7.sql - DESKTOP-UGRK24\SQLEXPRESS02.LojaDB (loja (54)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

LojaDB

Pesquisador de Objetos

Conectar

DESKTOP-UGRK24\SQLEXPRESS02 (SQL S

Bancos de Dados

Bancos de Dados do Sistema

master

model

msdb

tempdb

Instantâneos do Banco de Dados

LojaDB

Diagramas de Banco de Dados

Tabelas

Tabelas do Sistema

FileTables

Tabelas Externas

Tabelas de Gráfico

dbo.Movimento

dbo.Pessoa

Colunas

idPessoa (PK, int, nã

nome (varchar(255),

logradouro (varchar

cidade (varchar(255)

estado (char(2), nul

telefone (varchar(11

email (varchar(255),

tipo (char(1), nulo)

cpf (varchar(11), nul

cnpj (varchar(14), nu

Chaves

SQLQuery7.sql - DESKTOP-UGRK24\SQLEXPRESS02.LojaDB (loja (54))

SQLQuery4.sql - não conectado

SQLQuery3.sql - não conectado

SELECT * FROM Pessoa;

100 %

Resultados Mensagens

	idPessoa	nome	logradouro	cidade	estado	telefone	email
1	1	Joao	Rua 12, casa 3, Quitanda	Riacho do Sul	PA	11111111111	joao@riacho.com
2	2	JJC	Rua 11, Centro	Riacho do Norte	PA	12121212121	jic@riacho.com
3	3	Ana Silva	Av. das Flores, 123	Fortaleza	CE	99999999999	ana.silva@exemplo.com
4	4	Carlos Souza	Rua da Paz, 456	Salvador	BA	98888888888	carlos.souza@exemplo.com
5	5	Empresa Exemplo	Av. Principal, 500	São Paulo	SP	11223344556	contato@empresa.com
6	15	Ana Silva	Av. das Flores, 123	Fortaleza	CE	99999999999	ana.silva@exemplo.com
7	16	Empresa Exemplo	Av. Principal, 500	São Paulo	SP	11223344556	contato@empresa.com

Consulta executada com êxito.

DESKTOP-UGRK24\SQLEXPRESS02.LojaDB (loja (54)) LojaDB 00:00:00 7 linhas

=====

- 1 - Incluir Pessoa
- 2 - Alterar Pessoa
- 3 - Excluir Pessoa
- 4 - Buscar pelo Id
- 5 - Exibir Todos
- 0 - Finalizar Programa

=====

Escolha uma opção:

0

Finalizando programa...

BUILD SUCCESSFUL (total time: 6 minutes 0 seconds)

|

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

Diferenças entre Persistência em Arquivo e Persistência em Banco de Dados

Persistência em Arquivo:

Armazenamento: Dados são armazenados em arquivos no sistema de arquivos.

Consultas: Não há suporte para consultas complexas. É necessário ler e processar o arquivo manualmente.

Integridade: Menos mecanismos para garantir integridade e segurança dos dados.

Escalabilidade: Menos escalável, especialmente quando o volume de dados cresce.

Persistência em Banco de Dados:

Armazenamento: Dados são armazenados em tabelas dentro de um banco de dados gerenciado por um SGBD.

Consultas: Suporte para consultas SQL complexas, permitindo operações avançadas e análise de dados.

Integridade: Oferece mecanismos robustos para garantir integridade e segurança dos dados, como transações e restrições.

Escalabilidade: Mais escalável e eficiente para grandes volumes de dados e operações complexas.

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

Uso de Operador Lambda para Impressão de Valores

- Simplificação: O operador lambda em Java simplifica a escrita de funções anônimas e permite uma sintaxe mais concisa para operações sobre coleções. Por exemplo, a utilização de `forEach` com uma expressão lambda permite imprimir valores de uma lista de maneira mais compacta e legível:

```
list.forEach(item -> System.out.println(item));
```

Isso elimina a necessidade de criar classes anônimas ou métodos separados para operações simples de iteração e impressão.

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

Métodos static e o Método main

- Método main e static:
 - O método main é o ponto de entrada para a execução de um programa Java e deve ser static porque ele é chamado pela JVM (Java Virtual Machine) antes de qualquer instância da classe ser criada.
 - Métodos static pertencem à classe e não a uma instância específica. Isso significa que métodos static podem ser chamados diretamente sem a necessidade de criar um objeto da classe, o que é essencial para o método main funcionar como o ponto inicial de execução do programa.

<https://github.com/SaloGarcia/nivel03mundo03.git>