



**Estácio**

**CAMPUS POLO AUSTIN - NOVA IGUAÇU – RJ**

**DESENVOLVIMENTO FULL STACK**

**Nível 1: Iniciando o Caminho Pelo Java**

**RPG0014 9001**

**2024/2**

**SALOMÃO ISAAC CARVALHO GARCIA**

**Implementação de Cadastro em Modo Texto em Java com Persistência de Dados.**

Nova Iguaçu

15/07/2024

## Implementação de Cadastro em Modo Texto em Java com Persistência de Dados.

<https://github.com/SaloGarcia/Nivel01.git>

**Objetivo da Prática:** Adaptar o sistema de cadastro de pessoas para operar em modo texto, utilizando interação com o usuário via console e persistência de dados em arquivos binários. As funcionalidades incluem inclusão, alteração, exclusão, busca, listagem, salvamento e recuperação de dados.

### Códigos Desenvolvidos:

- Atualização do método Main para suportar interações via console.
- Implementação das operações de inclusão, alteração, exclusão, busca, listagem, salvamento e recuperação de dados.
- Adaptação das classes de entidades (PessoaFisica, PessoaJuridica) e repositórios para suportar operações em modo texto.

### Main.java

```
Source History
1 package cadastroPOO;
2
3 import model.*;
4 import java.io.IOException;
5 import java.util.List;
6 import java.util.Scanner;
7
8 public class Main {
9     public static void main(String[] args) {
10         Scanner scanner = new Scanner(System.in);
11         PessoaFisicaRepo repoFisica = new PessoaFisicaRepo();
12         PessoaJuridicaRepo repoJuridica = new PessoaJuridicaRepo();
13
14         int opcao = -1;
15         while (opcao != 0) {
16             System.out.println("\n=====");
17             System.out.println("1 - Incluir Pessoa");
18             System.out.println("2 - Alterar Pessoa");
19             System.out.println("3 - Excluir Pessoa");
20             System.out.println("4 - Buscar pelo Id");
21             System.out.println("5 - Exibir Todos");
22             System.out.println("6 - Persistir Dados");
23             System.out.println("7 - Recuperar Dados");
24             System.out.println("0 - Finalizar Programa");
25             System.out.println("=====");
26             System.out.print("Escolha uma opção: ");
27             opcao = scanner.nextInt();
28             scanner.nextLine(); // consume newline
29
30             switch (opcao) {
31                 case 1:
32                     System.out.print("F - Pessoa Fisica | J - Pessoa Juridica: ");
```

```
Source History
31 case 1:
32     System.out.print("F - Pessoa Física | J - Pessoa Jurídica: ");
33     String tipo = scanner.nextLine().toUpperCase();
34     if (tipo.equals("F")) {
35         System.out.print("Digite o id da pessoa: ");
36         int id = scanner.nextInt();
37         scanner.nextLine(); // consume newline
38         System.out.println("Insira os dados...");
39         System.out.print("Nome: ");
40         String nome = scanner.nextLine();
41         System.out.print("CPF: ");
42         String cpf = scanner.nextLine();
43         System.out.print("Idade: ");
44         int idade = scanner.nextInt();
45         repoFisica.inserir(new PessoaFisica(id, nome, cpf, idade));
46     } else if (tipo.equals("J")) {
47         System.out.print("Digite o id da pessoa: ");
48         int id = scanner.nextInt();
49         scanner.nextLine(); // consume newline
50         System.out.println("Insira os dados...");
51         System.out.print("Nome: ");
52         String nome = scanner.nextLine();
53         System.out.print("CNPJ: ");
54         String cnpj = scanner.nextLine();
55         repoJuridica.inserir(new PessoaJuridica(id, nome, cnpj));
56     }
57     break;
58 case 2:
59     // Alterar Pessoa
60     System.out.print("F - Pessoa Física | J - Pessoa Jurídica: ");
61     tipo = scanner.nextLine().toUpperCase();
62     if (tipo.equals("F")) {
```

```
Source History
61 tipo = scanner.nextLine().toUpperCase();
62 if (tipo.equals("F")) {
63     System.out.print("Digite o id da pessoa: ");
64     int id = scanner.nextInt();
65     scanner.nextLine(); // consume newline
66     PessoaFisica pf = repoFisica.obter(id);
67     if (pf != null) {
68         System.out.println("Dados atuais: ");
69         pf.exibir();
70         System.out.println("Insira os novos dados...");
71         System.out.print("Nome: ");
72         String nome = scanner.nextLine();
73         System.out.print("CPF: ");
74         String cpf = scanner.nextLine();
75         System.out.print("Idade: ");
76         int idade = scanner.nextInt();
77         repoFisica.alterar(new PessoaFisica(id, nome, cpf, idade));
78     } else {
79         System.out.println("Pessoa não encontrada.");
80     }
81 } else if (tipo.equals("J")) {
82     System.out.print("Digite o id da pessoa: ");
83     int id = scanner.nextInt();
84     scanner.nextLine(); // consume newline
85     PessoaJuridica pj = repoJuridica.obter(id);
86     if (pj != null) {
87         System.out.println("Dados atuais: ");
88         pj.exibir();
89         System.out.println("Insira os novos dados...");
90         System.out.print("Nome: ");
91         String nome = scanner.nextLine();
92         System.out.print("CNPJ: ");
```

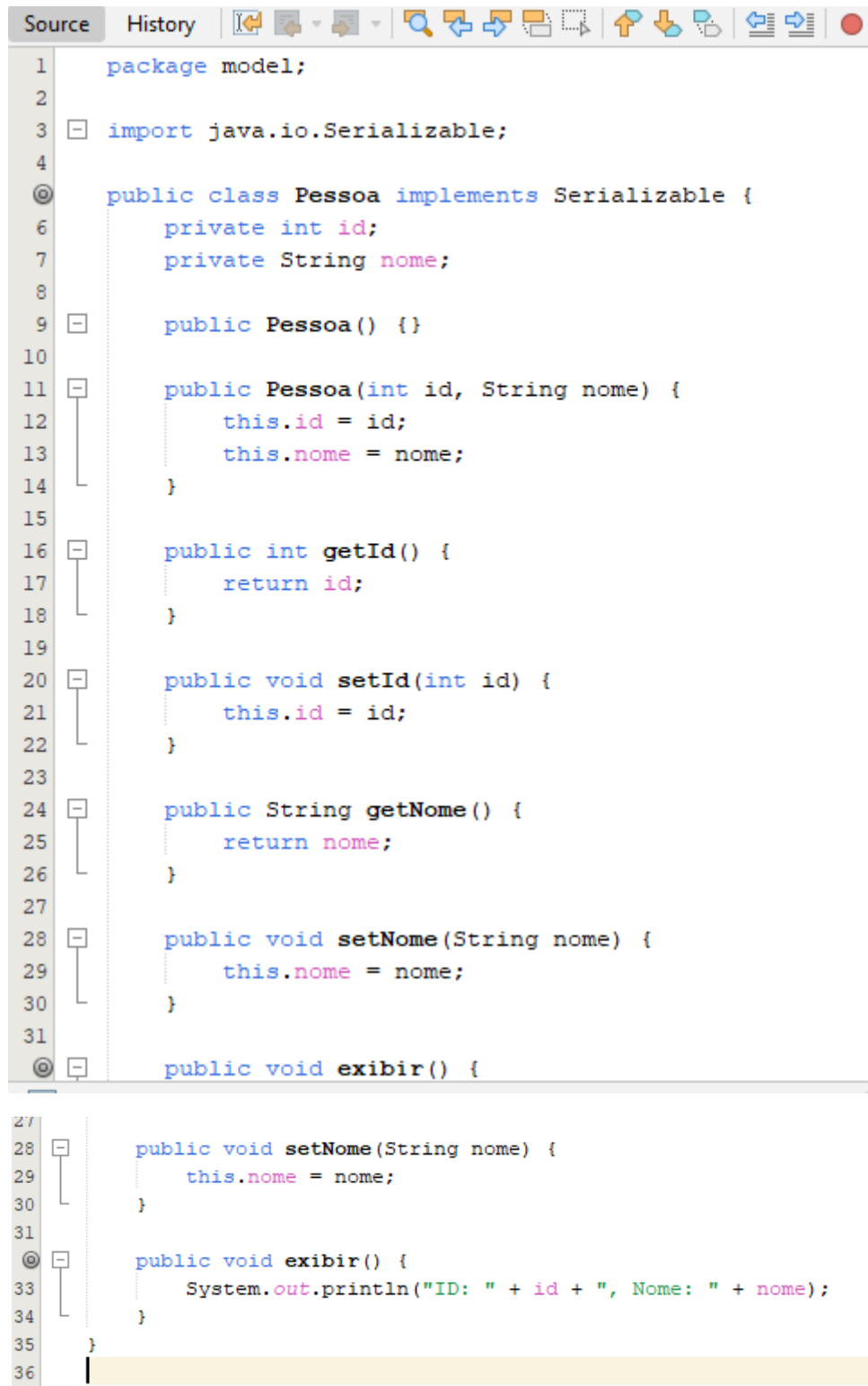
```
Source History
91         String nome = scanner.nextLine();
92         System.out.print("CNPJ: ");
93         String cnpj = scanner.nextLine();
94         repoJuridica.alterar(new PessoaJuridica(id, nome, cnpj));
95     } else {
96         System.out.println("Pessoa não encontrada.");
97     }
98 }
99 break;
100 case 3:
101     // Excluir Pessoa
102     System.out.print("F - Pessoa Fisica | J - Pessoa Juridica: ");
103     tipo = scanner.nextLine().toUpperCase();
104     if (tipo.equals("F")) {
105         System.out.print("Digite o id da pessoa: ");
106         int id = scanner.nextInt();
107         scanner.nextLine(); // consume newline
108         repoFisica.excluir(id);
109     } else if (tipo.equals("J")) {
110         System.out.print("Digite o id da pessoa: ");
111         int id = scanner.nextInt();
112         scanner.nextLine(); // consume newline
113         repoJuridica.excluir(id);
114     }
115     break;
116 case 4:
117     // Buscar pelo Id
118     System.out.print("F - Pessoa Fisica | J - Pessoa Juridica: ");
119     tipo = scanner.nextLine().toUpperCase();
120     if (tipo.equals("F")) {
121         System.out.print("Digite o id da pessoa: ");
122         int id = scanner.nextInt();
```

```
Source History
123         int id = scanner.nextInt();
124         scanner.nextLine(); // consume newline
125         PessoaFisica pf = repoFisica.obter(id);
126         if (pf != null) {
127             pf.exibir();
128         } else {
129             System.out.println("Pessoa não encontrada.");
130         }
131     } else if (tipo.equals("J")) {
132         System.out.print("Digite o id da pessoa: ");
133         int id = scanner.nextInt();
134         scanner.nextLine(); // consume newline
135         PessoaJuridica pj = repoJuridica.obter(id);
136         if (pj != null) {
137             pj.exibir();
138         } else {
139             System.out.println("Pessoa não encontrada.");
140         }
141     }
142     break;
143 case 5:
144     // Exibir Todos
145     System.out.print("F - Pessoa Fisica | J - Pessoa Juridica: ");
146     tipo = scanner.nextLine().toUpperCase();
147     if (tipo.equals("F")) {
148         List<PessoaFisica> todasFisicas = repoFisica.obterTodos();
149         todasFisicas.forEach(PessoaFisica::exibir);
150     } else if (tipo.equals("J")) {
151         List<PessoaJuridica> todasJuridicas = repoJuridica.obterTodos();
152         todasJuridicas.forEach(PessoaJuridica::exibir);
153     }
154     break;
```

```
Source History
154 case 6:
155     // Persistir Dados
156     System.out.print("Digite o prefixo dos arquivos: ");
157     String prefixoSalvar = scanner.nextLine();
158     try {
159         repoFisica.persistir(prefixoSalvar + ".fisica.bin");
160         repoJuridica.persistir(prefixoSalvar + ".juridica.bin");
161         System.out.println("Dados salvos com sucesso.");
162     } catch (IOException e) {
163         System.out.println("Erro ao salvar os dados: " + e.getMessage());
164     }
165     break;
166 case 7:
167     // Recuperar Dados
168     System.out.print("Digite o prefixo dos arquivos: ");
169     String prefixoRecuperar = scanner.nextLine();
170     try {
171         repoFisica.recuperar(prefixoRecuperar + ".fisica.bin");
172         repoJuridica.recuperar(prefixoRecuperar + ".juridica.bin");
173         System.out.println("Dados recuperados com sucesso.");
174     } catch (IOException | ClassNotFoundException e) {
175         System.out.println("Erro ao recuperar os dados: " + e.getMessage());
176     }
177     break;
178 case 0:
179     // Finalizar Programa
180     System.out.println("Finalizando o programa...");
181     break;
182 default:
183     System.out.println("Opção inválida!");
184 }
185 }
```


```
Source History
160         repoJuridica.persistir(prefixoSalvar + ".juridica.bin");
161         System.out.println("Dados salvos com sucesso.");
162     } catch (IOException e) {
163         System.out.println("Erro ao salvar os dados: " + e.getMessage());
164     }
165     break;
166 case 7:
167     // Recuperar Dados
168     System.out.print("Digite o prefixo dos arquivos: ");
169     String prefixoRecuperar = scanner.nextLine();
170     try {
171         repoFisica.recuperar(prefixoRecuperar + ".fisica.bin");
172         repoJuridica.recuperar(prefixoRecuperar + ".juridica.bin");
173         System.out.println("Dados recuperados com sucesso.");
174     } catch (IOException | ClassNotFoundException e) {
175         System.out.println("Erro ao recuperar os dados: " + e.getMessage());
176     }
177     break;
178 case 0:
179     // Finalizar Programa
180     System.out.println("Finalizando o programa...");
181     break;
182 default:
183     System.out.println("Opção inválida!");
184 }
185 }
186 }
187 scanner.close();
188 }
189 }
190 }
```

## Pessoa.java



```
1 package model;
2
3 import java.io.Serializable;
4
5 public class Pessoa implements Serializable {
6     private int id;
7     private String nome;
8
9     public Pessoa() {}
10
11     public Pessoa(int id, String nome) {
12         this.id = id;
13         this.nome = nome;
14     }
15
16     public int getId() {
17         return id;
18     }
19
20     public void setId(int id) {
21         this.id = id;
22     }
23
24     public String getNome() {
25         return nome;
26     }
27
28     public void setNome(String nome) {
29         this.nome = nome;
30     }
31
32     public void exibir() {
33
34     }
35 }
36
```

## PessoaFisica.java



```
Source History 
1 package model;
2
3 public class PessoaFisica extends Pessoa {
4     private String cpf;
5     private int idade;
6
7     public PessoaFisica() {
8         super();
9     }
10
11     public PessoaFisica(int id, String nome, String cpf, int idade) {
12         super(id, nome);
13         this.cpf = cpf;
14         this.idade = idade;
15     }
16
17     public String getCpf() {
18         return cpf;
19     }
20
21     public void setCpf(String cpf) {
22         this.cpf = cpf;
23     }
24
25     public int getIdade() {
26         return idade;
27     }
28
29     public void setIdade(int idade) {
30         this.idade = idade;
31     }
32
33     @Override
34     public void exibir() {
35         System.out.println("ID: " + getId() + ", Nome: " + getNome() + ", CPF: " + cpf + ", Idade: " + idade);
36     }
37 }
```

## PessoaJuridica.java

```
Source History
1 package model;
2
3 public class PessoaJuridica extends Pessoa {
4     private String cnpj;
5
6     public PessoaJuridica() {
7         super();
8     }
9
10    public PessoaJuridica(int id, String nome, String cnpj) {
11        super(id, nome);
12        this.cnpj = cnpj;
13    }
14
15    public String getCnpj() {
16        return cnpj;
17    }
18
19    public void setCnpj(String cnpj) {
20        this.cnpj = cnpj;
21    }
22
23    @Override
24    public void exibir() {
25        System.out.println("ID: " + getId() + ", Nome: " + getNome() + ", CNPJ: " + cnpj);
26    }
27 }
28
```

## PessoaFisicaRepo.java

```
Source History
1 package model;
2
3 import java.io.*;
4 import java.util.ArrayList;
5 import java.util.List;
6
7 public class PessoaFisicaRepo implements Serializable {
8     private List<PessoaFisica> pessoas;
9
10    public PessoaFisicaRepo() {
11        this.pessoas = new ArrayList<>();
12    }
13
14    public void inserir(PessoaFisica pessoa) {
15        pessoas.add(pessoa);
16    }
17
18    public void alterar(PessoaFisica pessoaNova) {
19        for (PessoaFisica pf : pessoas) {
20            if (pf.getId() == pessoaNova.getId()) {
21                pf.setNome(pessoaNova.getNome());
22                pf.setCpf(pessoaNova.getCpf());
23                pf.setIdade(pessoaNova.getIdade());
24                return;
25            }
26        }
27    }
28
29    public void excluir(int id) {
30        pessoas.removeIf(p -> p.getId() == id);
31    }
32
33    Git Repository Browser CadastroPOO (run) #7 runn
```



```
Source History
28
29 public void excluir(int id) {
30     pessoas.removeIf(p -> p.getId() == id);
31 }
32
33 public PessoaFisica obter(int id) {
34     for (PessoaFisica pf : pessoas) {
35         if (pf.getId() == id) {
36             return pf;
37         }
38     }
39     return null;
40 }
41
42 public List<PessoaFisica> obterTodos() {
43     return pessoas;
44 }
45
46 public void persistir(String arquivo) throws IOException {
47     ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(arquivo));
48     oos.writeObject(pessoas);
49     oos.close();
50 }
51
52 public void recuperar(String arquivo) throws IOException, ClassNotFoundException {
53     ObjectInputStream ois = new ObjectInputStream(new FileInputStream(arquivo));
54     pessoas = (List<PessoaFisica>) ois.readObject();
55     ois.close();
56 }
57 }
58
```

## PessoaJuridicaRepo.java

```
Source History
1 package model;
2
3 import java.io.*;
4 import java.util.ArrayList;
5 import java.util.List;
6
7 public class PessoaJuridicaRepo implements Serializable {
8     private List<PessoaJuridica> pessoas;
9
10    public PessoaJuridicaRepo() {
11        this.pessoas = new ArrayList<>();
12    }
13
14    public void inserir(PessoaJuridica pessoa) {
15        pessoas.add(pessoa);
16    }
17
18    public void alterar(PessoaJuridica pessoaNova) {
19        for (PessoaJuridica pj : pessoas) {
20            if (pj.getId() == pessoaNova.getId()) {
21                pj.setNome(pessoaNova.getNome());
22                pj.setCnpj(pessoaNova.getCnpj());
23                return;
24            }
25        }
26    }
27
28    public void excluir(int id) {
29        pessoas.removeIf(p -> p.getId() == id);
30    }
31
32    public PessoaJuridica obter(int id) {
```

```

31
32 public PessoaJuridica obter(int id) {
33     for (PessoaJuridica pj : pessoas) {
34         if (pj.getId() == id) {
35             return pj;
36         }
37     }
38     return null;
39 }
40
41 public List<PessoaJuridica> obterTodos() {
42     return pessoas;
43 }
44
45 public void persistir(String arquivo) throws IOException {
46     ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(arquivo));
47     oos.writeObject(pessoas);
48     oos.close();
49 }
50
51 public void recuperar(String arquivo) throws IOException, ClassNotFoundException {
52     ObjectInputStream ois = new ObjectInputStream(new FileInputStream(arquivo));
53     pessoas = (List<PessoaJuridica>) ois.readObject();
54     ois.close();
55 }
56 }
57

```

As funcionalidades solicitadas e os arquivos gerados foram verificados:

```

Output - CadastroPOO (run) #6

run:

=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

Escolha uma opção: 1
F - Pessoa Física | J - Pessoa Jurídica: f
Digite o id da pessoa: 1
Insira os dados...
Nome: João da Silva
CPF: 123.456.789-00
Idade: 30

```

#### Output - CadastroPOO (run) #6

```
Idade: 30

=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

Escolha uma opção: 1
F - Pessoa Física | J - Pessoa Jurídica: j
Digite o id da pessoa: 2
Insira os dados...
Nome: Empresa XYZ
CNPJ: 12.345.678/0001-90
```

#### Output - CadastroPOO (run) #6

```
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

Escolha uma opção: 2
F - Pessoa Física | J - Pessoa Jurídica: f
Digite o id da pessoa: 1
Dados atuais:
ID: 1, Nome: João da Silva, CPF: 123.456.789-00, Idade: 30
Insira os novos dados...
Nome: Maria da Silva
CPF: 987.654.321-00
Idade: 35
```

#### Output - CadastroPOO (run) #6

```
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====
Escolha uma opção: 3
F - Pessoa Física | J - Pessoa Jurídica: f
Digite o id da pessoa: 1
```

#### Output - CadastroPOO (run) #6

```
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====
Escolha uma opção: 4
F - Pessoa Física | J - Pessoa Jurídica: j
Digite o id da pessoa: 2
ID: 2, Nome: Empresa XYZ, CNPJ: 12.345.678/0001-90
```

#### Output - CadastroPOO (run) #6

```

ID: 2, Nome: Empresa XYZ, CNPJ: 12.345.678/0001-90

=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

Escolha uma opção: 5
F - Pessoa Física | J - Pessoa Jurídica: j
ID: 2, Nome: Empresa XYZ, CNPJ: 12.345.678/0001-90
```

#### Output - CadastroPOO (run) #6

```

=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

Escolha uma opção: 6
Digite o prefixo dos arquivos: dados_pessoas
Dados salvos com sucesso.
```

#### Output - CadastroPOO (run) #6

```
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

Escolha uma opção: 7
Digite o prefixo dos arquivos: dados_pessoas
Dados recuperados com sucesso.
```

#### Output - CadastroPOO (run) #6

```
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

Escolha uma opção: 7
Digite o prefixo dos arquivos: dados_pessoas
Dados recuperados com sucesso.

=====

1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

Escolha uma opção: 0
Finalizando o programa...
BUILD SUCCESSFUL (total time: 19 minutes 11 seconds)
```

## **Resultados da Execução:**

- As funcionalidades foram testadas e operam conforme esperado, permitindo interação completa do usuário via console.
- Os dados são persistidos corretamente nos arquivos dados\_pessoas.fisica.bin e dados\_pessoas.juridica.bin.

### **1. O que são elementos estáticos e qual o motivo para o método main adotar esse modificador?**

- **Elementos estáticos:** São membros de uma classe que pertencem à classe em si, não a instâncias individuais dessa classe. Eles são compartilhados por todas as instâncias da classe.
- **Motivo para o método main adotar static:** O método main precisa ser estático para que a JVM possa invocá-lo diretamente ao iniciar a aplicação Java, sem precisar criar uma instância da classe que o contém.

### **2. Para que serve a classe Scanner?**

- A classe Scanner em Java é utilizada para ler entradas de dados do usuário a partir do console ou de arquivos. Ela fornece métodos para ler diferentes tipos de dados primitivos (inteiros, floats, doubles) e strings, facilitando a interação do programa com o usuário.

### **3. Como o uso de classes de repositório impactou na organização do código?**

- O uso de classes de repositório promove uma separação clara de responsabilidades no código, isolando a lógica de persistência de dados das entidades do restante da aplicação.
- Isso melhora a modularidade do código, facilita a manutenção e permite a escalabilidade do sistema ao facilitar a adição de novas funcionalidades relacionadas à persistência sem modificar outras partes do código principal.

<https://github.com/SaloGarcia/Nivel01.git>