



**Campus de Santa Cruz da Serra  
Desenvolvimento Full Stack  
Iniciando o Caminho Pelo Java**



**2023.1**

**3º Semestre**

**Carlos Daniel Pereira dos Santos**

**Iniciando o caminho pelo Java - *Criação do Cadastro em Modo Texto***

---

**Objetivo:**

Implementação de um cadastro de clientes em modo texto, com persistência em arquivos, baseado na tecnologia Java.

**Link GitHub:** <https://github.com/dan-perr/Iniciando-o-caminho-pelo-Java.git>

Todos os códigos solicitados:

### ProjetoPOO (Main):

```
1 package projetopoo;
2
3 import java.util.List;
4 import java.io.IOException;
5 import java.util.Scanner;
6 import model.PessoaFisicaRepo;
7 import model.PessoaFisica;
8 import model.PessoaJuridicaRepo;
9 import model.PessoaJuridica;
10
11 public class ProjetoPOO {
12
13     public static void main(String[] args) {
14         try (Scanner scanner = new Scanner(System.in)) {
15             PessoaFisicaRepo repol = new PessoaFisicaRepo();
16             PessoaJuridicaRepo repo2 = new PessoaJuridicaRepo();
17             boolean continuar = true;
18
19             while (continuar) {
20                 System.out.println("=====");
21                 System.out.println("1 - Incluir Pessoa");
22                 System.out.println("2 - Alterar Pessoa");
23                 System.out.println("3 - Excluir Pessoa");
24                 System.out.println("4 - Buscar pelo ID");
25                 System.out.println("5 - Exibir Todos");
26                 System.out.println("6 - Persistir Dados");
27                 System.out.println("7 - Recuperar Dados");
28                 System.out.println("0 - Finalizar Programa");
29                 System.out.println("=====");
30                 int opcao = scanner.nextInt();
31
32                 switch (opcao) {
33                     case 1:
34                         System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
35                         char tipoPessoaChar = scanner.next().toUpperCase().charAt(0);
36                         scanner.nextLine();
37
38                         if (tipoPessoaChar == 'F') {
39                             System.out.println("Informe o ID: ");
40                             int id = scanner.nextInt();
41                             scanner.nextLine();
42
43                             System.out.println("Insira os dados...");
44
45                             System.out.println("Informe o nome: ");
46                             String nome = scanner.nextLine();
47
48                             System.out.println("Informe o CPF: ");
49                             String cpf = scanner.nextLine();
50                             PessoaFisica pessoaFisica = new PessoaFisica(id, nome, cpf);
51                             repol.incluir(pessoaFisica);
52                         } else if (tipoPessoaChar == 'J') {
53                             System.out.println("Informe o ID: ");
54                             int id = scanner.nextInt();
55                             scanner.nextLine();
56
57                             System.out.println("Insira os dados...");
58
59                             System.out.println("Informe o nome: ");
60                             String nome = scanner.nextLine();
61                             System.out.println("Informe o CNPJ: ");
62                             String cnpj = scanner.nextLine();
63                             PessoaJuridica pessoaJuridica = new PessoaJuridica(id, nome, cnpj);
64                             repo2.incluir(pessoaJuridica);
65                         }
66                     case 2:
67                         System.out.println("Informe o ID: ");
68                         int id = scanner.nextInt();
69                         scanner.nextLine();
70
71                         System.out.println("Informe o nome: ");
72                         String nome = scanner.nextLine();
73
74                         System.out.println("Informe o CPF: ");
75                         String cpf = scanner.nextLine();
76                         PessoaFisica pessoaFisica = new PessoaFisica(id, nome, cpf);
77                         repol.alterar(pessoaFisica);
78                     case 3:
79                         System.out.println("Informe o ID: ");
80                         int id = scanner.nextInt();
81                         scanner.nextLine();
82                         PessoaFisica pessoaFisica = new PessoaFisica(id, "", "");
83                         repol.excluir(pessoaFisica);
84                     case 4:
85                         System.out.println("Informe o ID: ");
86                         int id = scanner.nextInt();
87                         scanner.nextLine();
88                         PessoaFisica pessoaFisica = new PessoaFisica(id, "", "");
89                         PessoaFisica pessoaFisicaBuscada = repol.buscar(id);
90                         if (pessoaFisicaBuscada != null) {
91                             System.out.println("Pessoa encontrada:");
92                             System.out.println(pessoaFisicaBuscada);
93                         } else {
94                             System.out.println("Pessoa não encontrada.");
95                         }
96                     case 5:
97                         PessoaFisica[] pessoasFisicas = repol.todos();
98                         PessoaJuridica[] pessoasJuridicas = repo2.todos();
99                         System.out.println("Pessoas Fisicas:");
100                        for (PessoaFisica pessoaFisica : pessoasFisicas) {
101                            System.out.println(pessoaFisica);
102                        }
103                        System.out.println("Pessoas Juridicas:");
104                        for (PessoaJuridica pessoaJuridica : pessoasJuridicas) {
105                            System.out.println(pessoaJuridica);
106                        }
107                    case 6:
108                        repol.persistir();
109                        repo2.persistir();
110                    case 7:
111                        PessoaFisica[] pessoasFisicas = repol.recuperar();
112                        PessoaJuridica[] pessoasJuridicas = repo2.recuperar();
113                        System.out.println("Pessoas Fisicas recuperadas:");
114                        for (PessoaFisica pessoaFisica : pessoasFisicas) {
115                            System.out.println(pessoaFisica);
116                        }
117                        System.out.println("Pessoas Juridicas recuperadas:");
118                        for (PessoaJuridica pessoaJuridica : pessoasJuridicas) {
119                            System.out.println(pessoaJuridica);
120                        }
121                    case 0:
122                        continuar = false;
123                }
124            }
125        }
126    }
127 }
```

```

49 String cpf = scanner.next();
50
51 System.out.println("Informe a idade: ");
52 int idade = scanner.nextInt();
53
54 repol.inserir(new PessoaFisica(id, nome, cpf, idade));
55 } else if (tipoPessoaChar == 'J') {
56 System.out.println("Informe o ID, nome e CNPJ da Pessoa Juridica:");
57 int id = scanner.nextInt();
58 scanner.nextLine();
59
60 System.out.println("Informe o nome:");
61 String nome = scanner.nextLine();
62
63 System.out.println("Informe o CNPJ:");
64 String cnpj = scanner.next();
65
66 repol.inserir(new PessoaJuridica(id, nome, cnpj));
67 } else {
68 System.out.println("Opcao inválida. Tente novamente.");
69 }
70 break;
71 case 2:
72 System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
73 char tipoAlteracaoChar = scanner.next().toUpperCase().charAt(0);
74 scanner.nextLine();
75
76 System.out.println("Informe o ID da pessoa:");
77 int idAlteracao = scanner.nextInt();
78 scanner.nextLine();
79
80 System.out.println("Insira os dados...");
81
82 if (tipoAlteracaoChar == 'F') {
83 PessoaFisica pessoaFisicaExistente = repol.obter(indice: idAlteracao);
84 if (pessoaFisicaExistente != null) {
85 System.out.println("Informe o novo nome:");
86 String novoNome = scanner.nextLine();
87
88 System.out.println("Informe o novo CPF:");
89 String novoCpf = scanner.nextLine();
90
91 System.out.println("Informe a nova idade:");
92 int novaIdade = Integer.parseInt(scanner.nextLine());
93
94 PessoaFisica pessoaFisicaAlterada = new PessoaFisica(id: idAlteracao, nome: novoNome, cpf: novoCpf, idade: novaIdade);
95
96 repol.alterar(indice: idAlteracao, pessoaFisica: pessoaFisicaAlterada);
97 System.out.println("Pessoa Fisica alterada com sucesso");

```

```

98         } else {
99             System.out.println(x: "Pessoa Fisica nao encontrada.");
100         }
101     } else if (tipoAlteracaoChar == 'J') {
102         PessoaJuridica pessoaJuridica = repo2.obter(id: idAlteracao);
103         if (pessoaJuridica != null) {
104             System.out.println(x: "Dados atuais:");
105             pessoaJuridica.exibir();
106             System.out.println(x: "Informe o novo nome: ");
107             System.out.println(x: "Informe o novo CNPJ: ");
108             scanner.nextLine();
109             String nomeAlteracao = scanner.nextLine();
110             String cnpjAlteracao = scanner.next();
111             pessoaJuridica.setNome(nome: nomeAlteracao);
112             pessoaJuridica.setCnpj(cnpj: cnpjAlteracao);
113             repo2.alterar(pessoaJuridica);
114             System.out.println(x: "Pessoa Juridica alterada com sucesso.");
115         } else {
116             System.out.println(x: "Pessoa Juridica nao encontrada.");
117         }
118     } else {
119         System.out.println(x: "Opcao inválida. Tente novamente.");
120     }
121     break;
122 case 3:
123     System.out.println(x: "F - Pessoa Fisica | J - Pessoa Juridica");
124     char tipoExclusaoChar = scanner.next().toUpperCase().charAt(index: 0);
125     scanner.nextLine();
126
127     System.out.println(x: "Informe o ID da pessoa:");
128     int idExclusao = scanner.nextInt();
129
130     System.out.println(x: "Insira os dados...");
131
132     if (tipoExclusaoChar == 'F') {
133         PessoaFisica pessoaFisica = repol.obter(indice: idExclusao);
134         if (pessoaFisica != null) {
135             repol.excluir(indice: idExclusao);
136             System.out.println(x: "Pessoa Fisica excluida com sucesso.");
137         } else {
138             System.out.println(x: "Pessoa Fisica nao encontrada.");
139         }
140     } else if (tipoExclusaoChar == 'J') {
141         PessoaJuridica pessoaJuridica = repo2.obter(id: idExclusao);
142         if (pessoaJuridica != null) {
143             repo2.excluir(id: idExclusao);
144             System.out.println(x: "Pessoa Juridica excluida com sucesso.");
145         } else {
146             System.out.println(x: "Pessoa Juridica nao encontrada.");
147         }
148     } else {
149         System.out.println(x: "Opcao inválida. Tente novamente.");
150     }
151     break;
152 case 4:
153     System.out.println(x: "F - Pessoa Fisica | J - Pessoa Juridica");
154     char tipoObtencaoChar = scanner.next().toUpperCase().charAt(index: 0);
155     scanner.nextLine();
156
157     System.out.println(x: "Informe o ID da pessoa:");
158     int idObtencao = scanner.nextInt();
159
160     System.out.println(x: "Insira os dados...");
161
162     switch (tipoObtencaoChar) {
163         case 1 -> {
164             PessoaFisica pessoaFisica = repol.obter(indice: idObtencao);
165             if (pessoaFisica != null) {
166                 System.out.println(x: "Dados da Pessoa Fisica:");
167                 pessoaFisica.exibir();
168             } else {
169                 System.out.println(x: "Pessoa Fisica nao encontrada.");
170             }
171         }
172         case 2 -> {
173             PessoaJuridica pessoaJuridica = repo2.obter(id: idObtencao);
174             if (pessoaJuridica != null) {
175                 System.out.println(x: "Dados da Pessoa Juridica:");
176                 pessoaJuridica.exibir();
177             } else {
178                 System.out.println(x: "Pessoa Juridica nao encontrada.");
179             }
180         }
181         default -> System.out.println(x: "Opcao inválida. Tente novamente.");
182     }
183     break;
184
185 case 5:
186     System.out.println(x: "F - Pessoa Fisica | J - Pessoa Juridica");
187     char tipoListagemChar = scanner.next().toUpperCase().charAt(index: 0);
188
189     switch (tipoListagemChar) {
190         case 1 -> {
191             List<PessoaFisica> pessoasFisicas = repol.obterTodos();
192             System.out.println(x: "Pessoas Fisicas:");
193

```

```

194         for (PessoaFisica pessoa : pessoasFisicas) {
195             pessoa.exibir();
196         }
197     }
198     case 2 -> {
199         List<PessoaJuridica> pessoasJuridicas = repo2.obterTodos();
200         System.out.println(s: "Pessoas Juridicas:");
201         for (PessoaJuridica pessoa : pessoasJuridicas) {
202             pessoa.exibir();
203         }
204     }
205     default -> System.out.println(s: "Opcao inválida. Tente novamente.");
206 }
207 break;
208
209 case 6:
210     System.out.println(s: "Informe o prefixo dos arquivos:");
211     String prefixo = scanner.next();
212
213     repo1.persistir(prefixo + ".fisica.bin");
214     repo1.persistir(nomeArquivo: "C:/ProjetoNetBeans/ProjetoPOO/src/model/pessoas_fisicas.bin");
215     System.out.println("Dados de Pessoa Fisica salvos em " + prefixo + ".fisica.bin.");
216
217     try {
218         repo2.persistir(prefixo + ".juridica.bin");
219         repo2.persistir(nomeArquivo: "C:/ProjetoNetBeans/ProjetoPOO/src/model/pessoas_juridicas.bin");
220         System.out.println("Dados de Pessoa Juridica salvos em " + prefixo + ".juridica.bin.");
221     } catch (IOException e) {
222         System.out.println("Erro ao salvar os dados das pessoas juridicas: " + e.getMessage());
223     }
224
225 case 7:
226     System.out.println(s: "Informe o prefixo dos arquivos:");
227     String prefixoRecuperacao = scanner.next();
228
229     repo1.recuperar(prefixoRecuperacao + ".fisica.bin");
230     System.out.println("Dados de Pessoa Fisica recuperados de " + prefixoRecuperacao + ".fisica.bin.");
231
232     try {
233         repo2.recuperar(prefixoRecuperacao + ".juridica.bin");
234         System.out.println("Dados de Pessoa Juridica recuperados de " + prefixoRecuperacao + ".juridica.bin.");
235     } catch (IOException e) {
236         System.out.println("Erro ao recuperar os dados das pessoas juridicas: " + e.getMessage());
237     } catch (ClassNotFoundException e) {
238         System.out.println("Erro ao recuperar os dados das pessoas juridicas: " + e.getMessage());
239     }
240     break;
241
242 case 0:
243     try {
244         repo2.recuperar(prefixoRecuperacao + ".juridica.bin");
245         System.out.println("Dados de Pessoa Juridica recuperados de " + prefixoRecuperacao + ".juridica.bin.");
246     } catch (IOException e) {
247         System.out.println("Erro ao recuperar os dados das pessoas juridicas: " + e.getMessage());
248     } catch (ClassNotFoundException e) {
249         System.out.println("Erro ao recuperar os dados das pessoas juridicas: " + e.getMessage());
250     }
251     break;
252
253 case 8:
254     System.out.println(s: "Saindo do sistema.");
255     continuar = false;
256     break;
257 default:
258     System.out.println(s: "Opcao inválida. Tente novamente.");
259 }
260 }
261 }

```

## Por Extenso (ProjetoPOO.java):

*package projetopoo;*

*import java.util.List;*

*import java.io.IOException;*

*import java.util.Scanner;*

*import model.PessoaFisicaRepo;*

*import model.PessoaFisica;*

*import model.PessoaJuridicaRepo;*

*import model.PessoaJuridica;*

*public class ProjetoPOO {*

*public static void main(String[] args) {*

```

try (Scanner scanner = new Scanner(System.in)) {
    PessoaFisicaRepo repo1 = new PessoaFisicaRepo();
    PessoaJuridicaRepo repo2 = new PessoaJuridicaRepo();
    boolean continuar = true;

    while (continuar) {
        System.out.println("=====");
        System.out.println("1 - Incluir Pessoa");
        System.out.println("2 - Alterar Pessoa");
        System.out.println("3 - Excluir Pessoa");
        System.out.println("4 - Buscar pelo ID");
        System.out.println("5 - Exibir Todos");
        System.out.println("6 - Persistir Dados");
        System.out.println("7 - Recuperar Dados");
        System.out.println("0 - Finalizar Programa");
        System.out.println("=====");
        int opcao = scanner.nextInt();

        switch (opcao) {
            case 1:
                System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
                char tipoPessoaChar = scanner.next().toUpperCase().charAt(0);
                scanner.nextLine();

                if (tipoPessoaChar == 'F') {
                    System.out.println("Informe o ID: ");
                    int id = scanner.nextInt();
                    scanner.nextLine();

                    System.out.println("Insira os dados...");

                    System.out.println("Informe o nome: ");
                    String nome = scanner.nextLine();

                    System.out.println("Informe o CPF: ");
                    String cpf = scanner.next();

                    System.out.println("Informe a idade: ");
                    int idade = scanner.nextInt();

                    repo1.inserir(new PessoaFisica(id, nome, cpf, idade));
                } else if (tipoPessoaChar == 'J') {
                    System.out.println("Informe o ID, nome e CNPJ da Pessoa Juridica:");
                    int id = scanner.nextInt();
                    scanner.nextLine();

                    System.out.println("Informe o nome:");

```

```

String nome = scanner.nextLine();

System.out.println("Informe o CNPJ:");
String cnpj = scanner.next();

repo2.inserir(new PessoaJuridica(id, nome, cnpj));
} else {
    System.out.println("Opcao inválida. Tente novamente.");
}
break;
case 2:
    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
    char tipoAlteracaoChar = scanner.next().toUpperCase().charAt(0);
    scanner.nextLine();

    System.out.println("Informe o ID da pessoa:");
    int idAlteracao = scanner.nextInt();
    scanner.nextLine();

    System.out.println("Insira os dados...");

    if (tipoAlteracaoChar == 'F') {
        PessoaFisica pessoaFisicaExistente = repo1.obter(idAlteracao);
        if (pessoaFisicaExistente != null) {
            System.out.println("Informe o novo nome:");
            String novoNome = scanner.nextLine();

            System.out.println("Informe o novo CPF:");
            String novoCpf = scanner.nextLine();

            System.out.println("Informe a nova idade:");
            int novaldade = Integer.parseInt(scanner.nextLine());

            PessoaFisica pessoaFisicaAlterada = new
PessoaFisica(idAlteracao, novoNome, novoCpf, novaldade);

            repo1.alterar(idAlteracao, pessoaFisicaAlterada);
            System.out.println("Pessoa Fisica alterada com sucesso");
        } else {
            System.out.println("Pessoa Fisica nao encontrada.");
        }
    } else if (tipoAlteracaoChar == 'J') {
        PessoaJuridica pessoaJuridica = repo2.obter(idAlteracao);
        if (pessoaJuridica != null) {
            System.out.println("Dados atuais:");
            pessoaJuridica.exibir();
            System.out.println("Informe o novo nome: ");

```

```

        System.out.println("Informe o novo CNPJ: ");
        scanner.nextLine();
        String nomeAlteracao = scanner.nextLine();
        String cnpjAlteracao = scanner.next();
        pessoaJuridica.setNome(nomeAlteracao);
        pessoaJuridica.setCnpj(cnpjAlteracao);
        repo2.alterar(pessoaJuridica);
        System.out.println("Pessoa Juridica alterada com sucesso.");
    } else {
        System.out.println("Pessoa Juridica nao encontrada.");
    }
} else {
    System.out.println("Opcao inválida. Tente novamente.");
}
break;
case 3:
    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
    char tipoExclusaoChar = scanner.next().toUpperCase().charAt(0);
    scanner.nextLine();

    System.out.println("Informe o ID da pessoa:");
    int idExclusao = scanner.nextInt();

    System.out.println("Insira os dados...");

    if (tipoExclusaoChar == 'F') {
        PessoaFisica pessoaFisica = repo1.obter(idExclusao);
        if (pessoaFisica != null) {
            repo1.excluir(idExclusao);
            System.out.println("Pessoa Fisica excluida com sucesso.");
        } else {
            System.out.println("Pessoa Fisica nao encontrada.");
        }
    } else if (tipoExclusaoChar == 'J') {
        PessoaJuridica pessoaJuridica = repo2.obter(idExclusao);
        if (pessoaJuridica != null) {
            repo2.excluir(idExclusao);
            System.out.println("Pessoa Juridica excluida com sucesso.");
        } else {
            System.out.println("Pessoa Juridica nao encontrada.");
        }
    } else {
        System.out.println("Opcao inválida. Tente novamente.");
    }
    break;
case 4:
    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");

```



```
char tipoObtencaoChar = scanner.next().toUpperCase().charAt(0);
scanner.nextLine();
```

```
System.out.println("Informe o ID da pessoa:");
int idObtencao = scanner.nextInt();
```

```
System.out.println("Insira os dados...");
```

```
switch (tipoObtencaoChar) {
    case 1 -> {
        PessoaFisica pessoaFisica = repo1.obter(idObtencao);
        if (pessoaFisica != null) {
            System.out.println("Dados da Pessoa Fisica:");
            pessoaFisica.exibir();
        } else {
            System.out.println("Pessoa Fisica nao encontrada.");
        }
    }
    case 2 -> {
        PessoaJuridica pessoaJuridica = repo2.obter(idObtencao);
        if (pessoaJuridica != null) {
            System.out.println("Dados da Pessoa Juridica:");
            pessoaJuridica.exibir();
        } else {
            System.out.println("Pessoa Juridica nao encontrada.");
        }
    }
    default -> System.out.println("Opcao inválida. Tente novamente.");
}
break;
```

case 5:

```
System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
char tipoListagemChar = scanner.next().toUpperCase().charAt(0);
```

```
switch (tipoListagemChar) {
    case 1 -> {
        List<PessoaFisica> pessoasFisicas = repo1.obterTodos();
        System.out.println("Pessoas Fisicas:");
        for (PessoaFisica pessoa : pessoasFisicas) {
            pessoa.exibir();
        }
    }
    case 2 -> {
        List<PessoaJuridica> pessoasJuridicas = repo2.obterTodos();
        System.out.println("Pessoas Juridicas:");
    }
}
```

```

        for (PessoaJuridica pessoa : pessoasJuridicas) {
            pessoa.exibir();
        }
    }
    default -> System.out.println("Opcao inválida. Tente novamente.");
}
break;

case 6:
    System.out.println("Informe o prefixo dos arquivos:");
    String prefixo = scanner.next();

    repo1.persistir(prefixo + ".fisica.bin");

repo1.persistir("C:/ProjetoNetBeans/ProjetoPOO/src/model/pessoas_fisicas.bin")
;
    System.out.println("Dados de Pessoa Fisica salvos em " + prefixo +
".fisica.bin.");

    try{
        repo2.persistir(prefixo + ".juridica.bin");

repo2.persistir("C:/ProjetoNetBeans/ProjetoPOO/src/model/pessoas_juridicas.bin
");
        System.out.println("Dados de Pessoa Juridica salvos em " + prefixo +
".juridica.bin.");
    } catch (IOException e) {
        System.out.println("Erro ao salvar os dados das pessoas juridicas: " +
e.getMessage());
    }
case 7:
    System.out.println("Informe o prefixo dos arquivos:");
    String prefixoRecuperacao = scanner.next();

    repo1.recuperar(prefixoRecuperacao + ".fisica.bin");
    System.out.println("Dados de Pessoa Fisica recuperados de " +
prefixoRecuperacao + ".fisica.bin.");

    try{
        repo2.recuperar(prefixoRecuperacao + ".juridica.bin");
        System.out.println("Dados de Pessoa Juridica recuperados de " +
prefixoRecuperacao + ".juridica.bin.");
    } catch (IOException e) {
        System.out.println("Erro ao recuperar os dados das pessoas
juridicas: " + e.getMessage());
    } catch (ClassNotFoundException e) {

```

```

        System.out.println("Erro ao recuperar os dados das pessoas
juridicas: " + e.getMessage());
    }
    break;

    case 0:
        System.out.println("Saindo do sistema.");
        continuar = false;
        break;
    default:
        System.out.println("Opcao invalida. Tente novamente.");
    }
}
}
}
}
}

```

---

### Análise e Conclusões:

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

**Resposta:**

*Elementos estáticos pertencem à classe, não a uma instância. O método main é estático para permitir que a JVM o acesse sem criar uma instância da classe.*

2. Para que serve a classe Scanner?

**Resposta:** A classe Scanner é usada para obter a entrada do usuário.

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

**Resposta:**

*Classes de repositório separam a lógica de negócios da lógica de persistência de dados, tornando o código mais organizado e manutenível.*