



UNIVERSIDADE ESTÁCIO DE SÁ

Campus Estácio - Castelo - Belo Horizonte - MG

Curso: Desenvolvimento Full Stack

Disciplina: Nível 1 - Iniciando o Caminho Pelo Java (RPG0014)

Turma: 9002

Semestre Letivo: 2025.1

Nome do discente: Ivan de Ávila Carvalho Fleury Mortimer

Repositório do projeto: <https://github.com/ivanmortimer/CadastroPOO>

---

## **Missão Prática | Nível 1 | Mundo 3:**

### **2º Procedimento | Criação do Cadastro em Modo Texto**

#### **Objetivos da Prática**

1. Implementar um sistema de menu em modo texto para interação com o usuário.
2. Permitir a inclusão, alteração, exclusão, busca e listagem de registros de pessoas físicas e jurídicas.
3. Realizar a persistência e a recuperação de dados em arquivos binários.
4. Utilizar tratamento de exceções para validar entradas do usuário e manter a integridade dos dados.
5. Explorar os recursos da IDE NetBeans para facilitar o desenvolvimento e execução do sistema.

## Códigos Fonte

O código principal do sistema foi desenvolvido no arquivo '**CadastroP002.java**', dentro do pacote '**cadastropoo**', utilizando a IDE NetBeans. Esse arquivo contém o novo método '**main**', que implementa todas as funcionalidades do sistema em modo texto solicitadas pela prática.

As opções de menu apresentadas ao usuário incluem:

- Incluir Pessoa
- Alterar Pessoa (com verificação de id existente e rollback em caso de erro)
- Excluir Pessoa (com verificação de id existente)
- Buscar Pessoa por ID
- Exibir Todos os registros
- Persistir Dados em arquivos binários
- Recuperar Dados a partir dos arquivos
- Encerrar o programa

Todas as entradas do usuário são validadas. Entradas fora do esperado (como strings ou floats onde só são permitidos inteiros) são tratadas com mensagens informativas sem encerrar o programa abruptamente.

O sistema também verifica se arquivos com o mesmo prefixo já existem antes da persistência, solicitando confirmação ao usuário antes de sobrescrevê-los.

Além disso, se o usuário escolher a opção 2 (Alterar Pessoa) ou a opção 3 (Excluir Pessoa), após ele especificar o tipo de pessoa cujos dados ele deseja alterar ou excluir (se é uma Pessoa Física ou um Pessoa Jurídica), a informação que o sistema solicita do usuário imediatamente em seguida é a '**id**' da pessoa. Caso essa '**id**' não exista o sistema exibirá ao usuário uma das seguinte mensagens dependendo se a escolha foi alterar ou excluir os dados de uma pessoa:

```
Erro ao alterar pessoa: ID não encontrado na lista.
```

```
Erro ao excluir pessoa: Nenhum elemento com o ID fornecido foi encontrado.
```

E, em seguida, o sistema volta ao menu inicial.

Por fim, o sistema também oferece mais uma solução robusta em termos de consistência e segurança. Quando o usuário escolhe a opção 2 (Alterar Pessoa), após ele especificar o tipo de pessoa cujos dados ele deseja alterar (se é uma Pessoa Física ou um Pessoa Jurídica), e especificar também a 'id' da pessoa cujos dados ele deseja alterar, caso essa pessoa exista no sistema, são exibidos na saída os dados dessa pessoa presentes no sistema e, em seguida, são solicitados ao usuário os novos dados da Pessoa Física (Id, Nome, CPF e Idade, nessa sequência) ou da Pessoa Jurídica (Id, Nome e CNPJ, nessa sequência) que ele deseja que substituam os dados existentes no sistema. Porém, antes de se alterar sequencialmente esses dados no sistema, eles são antes armazenados em variáveis temporárias para que, se durante o processo de alteração dos atributos dessa pessoa, alguma exceção relativa ao formato ou à faixa permitida do dado inserido seja lançada (como uma exceção lançada na validação do CPF, da Idade, ou do CNPJ fornecido), então o sistema restaura todos os atributos aos valores originais, de forma que os dados não fiquem inconsistentes (como por exemplo, o nome de uma pessoa ficar associado ao CPF ou à idade ou ao CNPJ de outra pessoa, pelo simples fato de que os dados dessa pessoa foram alterados só parcialmente devido à ocorrência de uma exceção antes que a alteração de todos os seus dados tivesse sido completada).

Além do pacote Java 'cadastropoo', onde está presente o arquivo 'CadastroPOO2.java' que contém o código principal do sistema, em que está declarada a classe pública 'CadastroPOO2', que contém o método 'main', por onde começa a execução do programa, o programa aplicativo Java 'CadastroPOO', também inclui o pacote Java 'model' que contém os seguintes arquivos:

- Pessoa.java: Classe abstrata base contendo atributos comuns entre pessoas físicas e jurídicas.
- PessoaFisica.java: Subclasse de Pessoa que representa uma pessoa física, contendo CPF e data de nascimento.
- PessoaJuridica.java: Subclasse de Pessoa que representa uma pessoa jurídica, contendo CNPJ e razão social.
- PessoaFisicaRepo.java: Classe responsável pelo repositório de pessoas físicas, com funcionalidades de inclusão, listagem, exclusão e persistência em arquivos binários.
- PessoaJuridicaRepo.java: Classe responsável pelo repositório de pessoas jurídicas, com funcionalidades análogas ao repositório de pessoas físicas.
- ExcecaoValidacaoCPF.java, ExcecaoValidacaoCNPJ.java, ExcecaoValidacaoIdade.java: Exceções personalizadas utilizadas para validar entradas do usuário.

## Resultados da Execução dos Códigos

O programa foi executado com sucesso na IDE NetBeans, permitindo ao usuário:

- Incluir e alterar pessoas com validação de id, CPF, idade e CNPJ.
- Exibir e buscar dados de maneira clara e formatada.

- Persistir e recuperar os dados utilizando arquivos binários.
- Corrigir erros de entrada sem travar o programa.

Abaixo, um exemplo da saída exibida após uma execução completa do sistema:

```
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
Digite uma opção (0-7): 6
Digite o prefixo para os arquivos: repo5
Aviso: já existem arquivos salvos com o mesmo prefixo, deseja sobrescrever esses arquivos?
S - Sim | N - Não
s
Dados do objeto da classe 'PessoaFisica' armazenados no arquivo:
repo5.fisica.bin
Dados do objeto da classe 'PessoaJuridica' armazenados no arquivo:
repo5.juridica.bin
```

## Análise e Conclusão

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

Elementos estáticos pertencem à classe e não às instâncias. O método '**main**' é estático porque ele é o ponto de entrada do programa e deve ser acessível pela JVM sem a necessidade de criar um objeto da classe.

### b. Para que serve a classe Scanner?

A classe '**Scanner**' é utilizada para capturar entradas do usuário via teclado. No projeto, ela permite interagir com o usuário por meio de um menu em modo texto, recebendo comandos e dados.

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

As classes de repositório, como '**PessoaFisicaRepo**' e '**PessoaJuridicaRepo**', centralizam a lógica de armazenamento, busca, alteração, exclusão e persistência dos dados. Isso separa as responsabilidades da lógica de controle, tornando o código mais modular, reutilizável e fácil de manter.

**Ivan de Ávila Carvalho Fleury Mortimer**

Belo Horizonte, MG

Abril de 2025