

Desenvolvimento de Programação Back-end com Java

Edson Victor Miranda de Oliveira - 2024 0836 7775

Polo Parangaba - Fortaleza/CE

Programação Back-end com Java – 2025/3

TÍTULO DA PRÁTICA

Desenvolvimento de um Sistema de Cadastro de Pessoas em Java utilizando Programação Orientada a Objetos, Repositórios e Persistência de Dados

OBJETIVO DA PRÁTICA

A prática tem como objetivo implementar um sistema em Java com menus interativos que permita:

- Cadastrar novas pessoas;
- Alterar dados de pessoas cadastradas;
- Excluir registros;
- Realizar buscas por ID;
- Exibir todos os registros armazenados;
- Persistir dados em arquivo;
- Recuperar dados previamente salvos;

Além disso, reforçar conceitos como:

- Classes, objetos e encapsulamento;
- Estruturas repositório;
- Manipulação de arquivos;
- Uso da classe Scanner;
- Elementos e métodos estáticos, especialmente o método main.

RESULTADOS DA EXECUÇÃO

Durante a execução do programa, é exibido um menu interativo com as seguintes opções:

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

O sistema permite adicionar, editar, buscar e remover registros corretamente, além de salvar e carregar dados a partir de arquivos, garantindo persistência entre execuções.

Todo o funcionamento foi testado e validado através do console.

ANÁLISE E CONCLUSÃO

1) O que são elementos estáticos e por que o método main usa static?

Elementos estáticos são aqueles que pertencem à **classe**, e não a um objeto específico. Isso significa que podem ser acessados sem que seja criada uma instância da classe.

O método main precisa obrigatoriamente ser static porque:

- Ele é o **ponto de entrada** da aplicação Java;
- O Java precisa chamá-lo antes mesmo de qualquer objeto existir;
- Sem static, seria necessário instanciar a classe para chamar o main, o que é impossível nesse ponto inicial.

2) Para que serve a classe Scanner?

A classe Scanner é utilizada para **ler entrada de dados** fornecida pelo usuário através do teclado.

Ela permite capturar:

- Strings
- Números inteiros
- Números reais
- Caracteres
- Opções de menu

É fundamental para tornar aplicações Java **interativas**.

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

A utilização de um **Repositório** melhora a arquitetura do projeto porque:

- Centraliza operações de armazenamento, busca, edição e remoção de registros.
- Separa a lógica de negócios da interface de usuário.
- Evita repetição de código.
- Facilita manutenção e futuras expansões (como trocar de arquivo para banco de dados).

- Torna o sistema mais modular, limpo e organizado.

O padrão de repositório é essencial em projetos reais de back-end.

LINK DO REPOSITÓRIO GIT

<https://github.com/edinbrr/CadastroPOO>

CONCLUSÃO FINAL

A prática permitiu exercitar conceitos fundamentais da Programação Orientada a Objetos e demonstrar a importância da modularização na construção de sistemas reais.

O uso de repositórios, métodos estáticos, leitura com Scanner e manipulação de arquivos proporcionou uma compreensão clara de como aplicações Java estruturadas são criadas e mantidas no mundo profissional.