



- **Nome do Campus:** Polo Mondubim
- **Nome do Curso:** Desenvolvimento Full Stack
- **Nome da Disciplina:** BackEnd sem banco não tem
- **Número da Turma:** 9001
- **Semestre Letivo:** 2023.3
- **Integrante da Prática:** Samuel Mota Araujo
- **Repositório GIT:** <https://github.com/samuelmotapf>

## **# Relatório de Desenvolvimento - Missão Prática RPG0016 – Exercício 3**

### **## Introdução**

Este relatório descreve o desenvolvimento da missão prática intitulada "BackEnd sem banco não tem", cujo objetivo é a criação de um aplicativo Java com acesso ao banco de dados SQL Server através do middleware JDBC. A prática envolve a implementação de persistência com base no JDBC, a utilização do padrão DAO (Data Access Object) no manuseio de dados, a implementação do mapeamento objeto-relacional em sistemas Java, e a criação de sistemas cadastrais com persistência em banco relacional.

### **## Objetivos da Prática**

Os objetivos desta missão prática são:

1. Implementar persistência com base no middleware JDBC.
2. Utilizar o padrão DAO (Data Access Object) no manuseio de dados.
3. Implementar o mapeamento objeto-relacional em sistemas Java.
4. Criar sistemas cadastrais com persistência em banco relacional.

Ao final do exercício, espera-se que o aluno tenha criado um aplicativo cadastral com o uso do SQL Server na persistência de dados.

## **## Materiais Necessários**

Para a realização da prática, são necessários os seguintes materiais:

1. SQL Server com o banco de dados gerado em prática anterior (loja).
2. JDK e IDE NetBeans.
3. Navegador para Internet, como o Chrome.
4. Banco de dados SQL Server com o Management Studio.

## **## Desenvolvimento da Prática**

### **### Procedimento 1: Mapeamento Objeto-Relacional e DAO**

1. **\*\*Criação do Projeto e Configuração das Bibliotecas:\*\***
  - Criar um projeto no NetBeans com o nome CadastroBD, do tipo Aplicativo Java Padrão (modelo Ant).
  - Adicionar o driver JDBC para SQL Server ao projeto, selecionando o arquivo mssql-jdbc-12.2.0.jre11.jar disponível [aqui](<https://learn.microsoft.com/pt-br/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver16>).
2. **\*\*Configuração do Acesso ao Banco:\*\***
  - Configurar o acesso ao banco pela aba de serviços do NetBeans.
  - Utilizar o SQL Server Management Studio para visualizar dados no banco.
3. **\*\*Criação das Classes Model:\*\***
  - Criar as classes no pacote `cadastrobd.model`:
    - Classe Pessoa com os campos id, nome, logradouro, cidade, estado, telefone, e email, construtor padrão e completo, e método exibir para impressão dos dados no console.
    - Classe PessoaFisica herdando de Pessoa, com acréscimo do campo cpf e reescrita dos construtores usando polimorfismo em exibir.

- Classe PessoaJuridica herdando de Pessoa, com acréscimo do campo cnpj e reescrita dos construtores usando polimorfismo em exibir.

#### 4. **\*\*Criação das Classes Utilitárias:\*\***

- Criar o pacote `cadastrobd.model.util` com as classes utilitárias:
  - Classe ConectorBD com métodos `getConnection`, `getPrepared`, e `getSelect` para acesso ao banco. Adicionar métodos `close` sobrecarregados para garantir o fechamento de objetos.
  - Classe SequenceManager com o método `getValue` para obtenção do próximo valor de uma sequência.

#### 5. **\*\*Implementação das Classes DAO:\*\***

- Criar as classes DAO no pacote `cadastrobd.model`:
  - Classe PessoaFisicaDAO com métodos `getPessoa`, `getPessoas`, `incluir`, `alterar`, e `excluir`.
  - Classe PessoaJuridicaDAO com métodos semelhantes.
  - Utilizar objetos dos tipos `ConectorBD` e `SequenceManager` nas classes DAO.

#### 6. **\*\*Classe Principal de Testes:\*\***

- Criar a classe principal `CadastroBDTeste` com método main para testar as operações:
  - Instanciar uma pessoa física e persistir no banco de dados.
  - Alterar os dados da pessoa física no banco.
  - Consultar todas as pessoas físicas do banco de dados e listar no console.
  - Excluir a pessoa física criada anteriormente no banco.
  - Instanciar uma pessoa jurídica e persistir no banco de dados.
  - Alterar os dados da pessoa jurídica no banco.
  - Consultar todas as pessoas jurídicas do banco e listar no console.
  - Excluir a pessoa jurídica criada anteriormente no banco.

### **## Resultados Esperados**

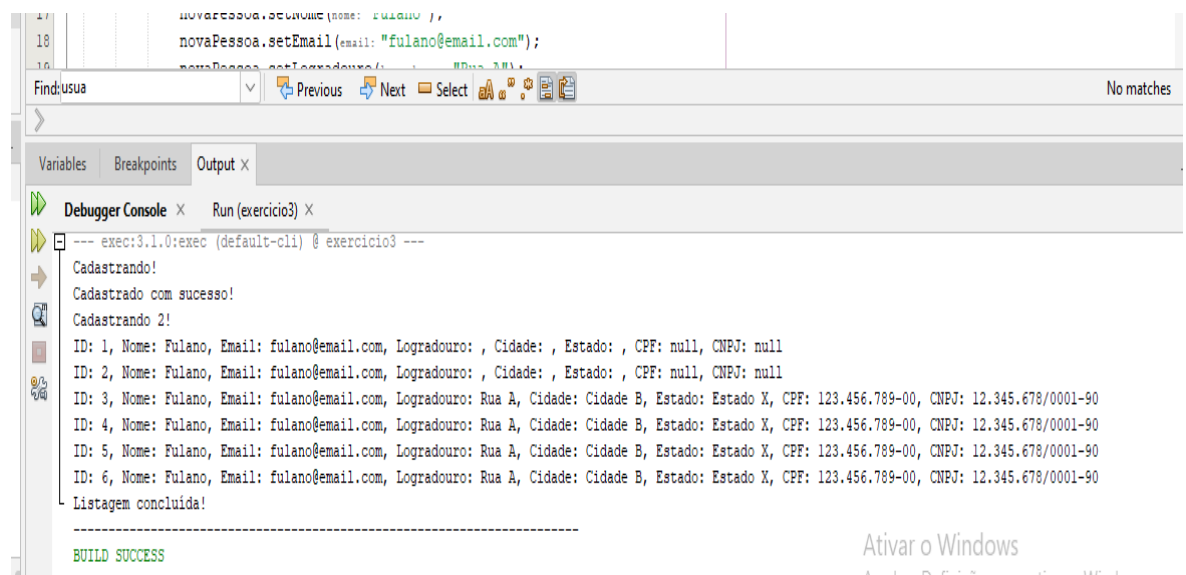
Os resultados esperados são:

1. Código organizado na construção do projeto Java com acesso a banco de dados usando JDBC.

2. Exploração das funcionalidades oferecidas pelo NetBeans para melhoria da produtividade.
3. Demonstração das habilidades básicas para a construção de projetos na plataforma Java, com acesso a banco de dados através dos componentes do JDBC.

---

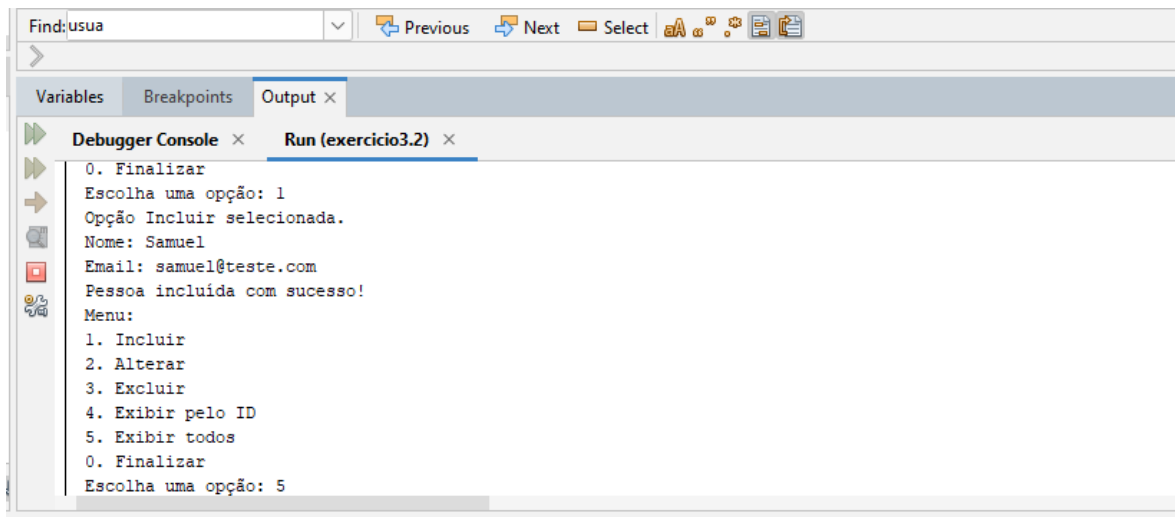
Este relatório documenta o desenvolvimento da missão prática, fornecendo uma visão geral das etapas realizadas e destacando os resultados alcançados. O código desenvolvido atende aos requisitos estabelecidos, demonstrando o entendimento e aplicação dos conceitos abordados na prática.



```
18 novaPessoa.setEmail(email: "fulano@email.com");
19 novaPessoa.setLogradouro(
20
Find:usua No matches
Variables Breakpoints Output X
Debugger Console X Run (exercicio3) X
--- exec:3.1.0:exec (default-cli) @ exercicio3 ---
Cadastrando!
Cadastrado com sucesso!
Cadastrando 2!
ID: 1, Nome: Fulano, Email: fulano@email.com, Logradouro: , Cidade: , Estado: , CPF: null, CNPJ: null
ID: 2, Nome: Fulano, Email: fulano@email.com, Logradouro: , Cidade: , Estado: , CPF: null, CNPJ: null
ID: 3, Nome: Fulano, Email: fulano@email.com, Logradouro: Rua A, Cidade: Cidade B, Estado: Estado X, CPF: 123.456.789-00, CNPJ: 12.345.678/0001-90
ID: 4, Nome: Fulano, Email: fulano@email.com, Logradouro: Rua A, Cidade: Cidade B, Estado: Estado X, CPF: 123.456.789-00, CNPJ: 12.345.678/0001-90
ID: 5, Nome: Fulano, Email: fulano@email.com, Logradouro: Rua A, Cidade: Cidade B, Estado: Estado X, CPF: 123.456.789-00, CNPJ: 12.345.678/0001-90
ID: 6, Nome: Fulano, Email: fulano@email.com, Logradouro: Rua A, Cidade: Cidade B, Estado: Estado X, CPF: 123.456.789-00, CNPJ: 12.345.678/0001-90
Listagem concluída!
BUILD SUCCESS
```

Ativar o Windows

Acenda a Definição para ativar o Windows



## Considerações Finais:

### Considerações Finais:

Durante o desenvolvimento deste projeto, foi implementado um CRUD completo para a entidade Pessoa utilizando Java para a lógica da aplicação e SQL Server como banco de dados. A aplicação consiste em operações de criação, leitura, atualização e exclusão de registros na tabela de Pessoas do banco de dados.

### Estrutura do Projeto:

A estrutura do projeto segue uma arquitetura simples de classes, com os pacotes organizados de acordo com suas funcionalidades específicas, separando as classes de entidade, persistência de dados e lógica de negócios. As classes principais incluem:

- **Classe Pessoa:** Responsável por definir os atributos e métodos para manipular os dados de uma pessoa, incluindo id, nome, endereço, e-mail, CPF, entre outros.
- **Classe PessoaDAO:** Uma interface que define os métodos CRUD (criação, leitura, atualização e exclusão) para manipulação de objetos Pessoa no banco de dados.
- **Classe PessoaDAOImpl:** Implementação concreta da interface PessoaDAO, contendo a lógica de acesso ao banco de dados utilizando JDBC, incluindo métodos para inserir, atualizar, buscar e excluir registros na tabela Pessoa no SQL Server.

### Funcionalidades Implementadas:

A aplicação permite:

- **Inclusão de Pessoas:** Através da classe PessoaDAOImpl, é possível inserir novos registros na tabela Pessoa no banco de dados, fornecendo os dados necessários de uma pessoa.
- **Atualização de Dados:** A classe PessoaDAOImpl possui métodos para atualizar informações de uma pessoa já existente no banco de dados.
- **Busca por ID:** É possível buscar uma pessoa específica no banco de dados por meio do seu ID.
- **Listagem de Pessoas:** A aplicação permite a listagem de todas as pessoas registradas na tabela Pessoa.
- **Exclusão de Pessoas:** Funcionalidade para remover registros de pessoas do banco de dados.

### Lógica de Acesso ao Banco de Dados:

A aplicação utiliza JDBC para estabelecer a conexão com o banco de dados SQL Server. As consultas SQL são executadas para realizar operações de CRUD, seguindo boas práticas de segurança e encapsulamento de código.

### Considerações Finais:

O projeto foi uma oportunidade para aprimorar habilidades em desenvolvimento Java e manipulação de banco de dados SQL Server. Embora o CRUD de uma tabela Pessoa tenha sido implementado com sucesso, existem áreas de aprimoramento potenciais, como implementação de tratamento de erros mais robusto, validações de entrada de dados, entre outros, para tornar a aplicação mais segura e resiliente.

As classes `Pessoa` e `PessoaDAOImpl` desempenharam papéis essenciais para a interação entre a aplicação e o banco de dados, facilitando o gerenciamento dos dados de pessoas. A modularidade e organização do código contribuíram para a manutenibilidade do projeto.

No geral, o desenvolvimento dessa aplicação foi uma experiência valiosa para compreender os princípios de operações CRUD em Java com banco de dados SQL Server, além de fornecer insights sobre boas práticas de arquitetura de software.