



## Missão Prática | Nível 3 | Mundo 3

Rian Joseph Ramos Felizardo - 202202923931

POLO BARREIRO - Belo Horizonte, MG Nível 3 - BackEnd sem banco não tem – 2023.1 – 3º Semestre Letivo

Repositório: [desenvolvimento-sistemas-mundo-3/nivel-03](https://github.com/rianjrp/desenvolvimento-sistemas-mundo-3/nivel-03) at [main](#) · [rianjrp/desenvolvimento-sistemas-mundo-3](https://github.com/rianjrp/desenvolvimento-sistemas-mundo-3)

### Objetivo da Prática

Implementar persistência com base no middleware JDBC.

Utilizar o padrão DAO (Data Access Object) no manuseio de dados.

Implementar o mapeamento objeto-relacional em sistemas Java.

Criar sistemas cadastrais com persistência em banco relacional.

No final do exercício, o aluno terá criado um aplicativo cadastral com uso do SQL Server na persistência de dados.

### 1º Procedimento | Mapeamento Objeto-Relacional e DAO

#### a) Qual a importância dos componentes de middleware, como o JDBC?

O JDBC desempenha um papel fundamental na simplificação do desenvolvimento de software distribuído, promovendo a portabilidade, abstração de banco de dados e facilitando a interação entre aplicativos e fontes de dados.

#### b) Qual a diferença no uso de *Statement* ou *PreparedStatement* para a manipulação de dados?

Os dois são usados para executar instruções SQL num banco de dados, o que os diferencia são seu uso e desempenho. Basicamente, o *Statement* é usado quando a consulta não tem parâmetros dinâmicos, é executada poucas vezes ou uma

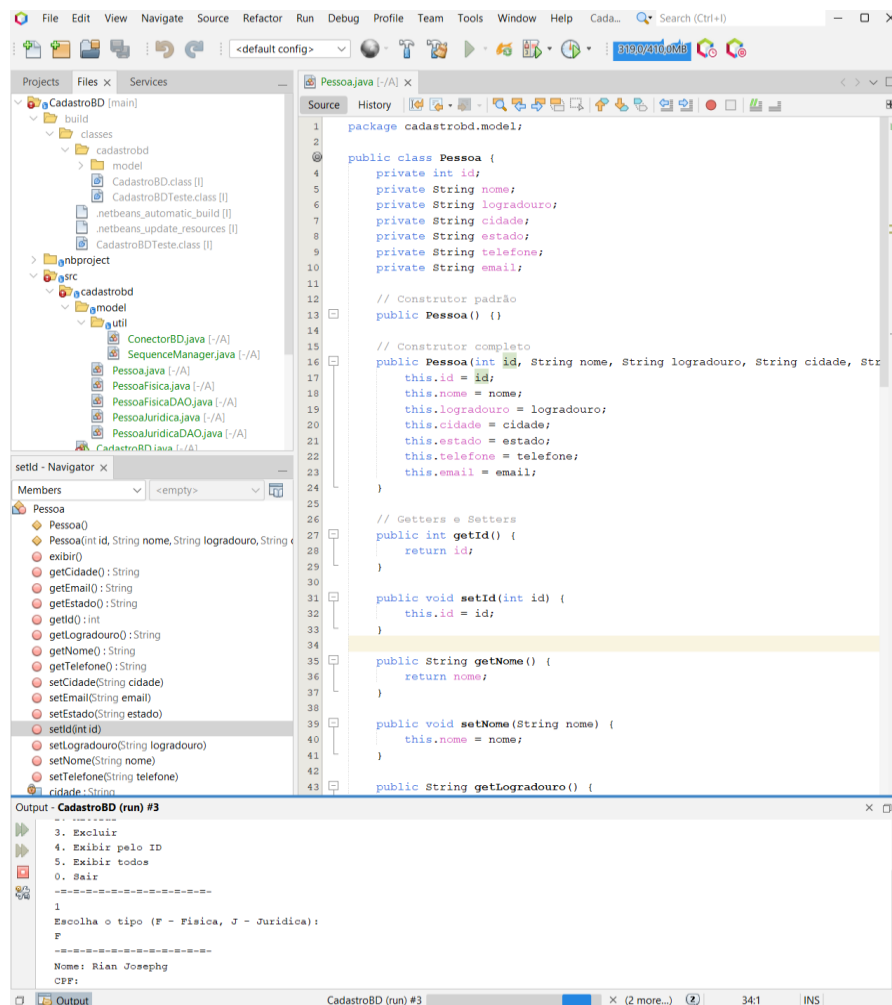
única vez e a complexidade da aplicação é baixa. Já o PreparedStatement é usado a consulta contém parâmetros que, podem variar, é executada várias vezes com diferentes valores de parâmetros, a segurança é uma preocupação ou a aplicação é complexa e precisa de uma abordagem mais robusta.

**c) Como o padrão DAO melhora a manutenibilidade do software?**

Melhora ao fornecer uma estrutura organizada e flexível para o acesso a dados, facilitando a evolução do sistema e a manutenção ao longo do tempo.

**d) Como a herança é refletida no banco de dados, quando lidamos com um modelo estritamente relacional?**

Geralmente é implementada de uma das seguintes maneiras: herança por tabela única (single table inheritance) ou herança por tabela por classe (class table inheritance). A escolha vai depender das necessidades específicas do sistema e dos padrões de consulta mais comuns.



## 2º Procedimento | Alimentando a Base

Inserir neste campo, **de forma organizada**, todos os códigos do roteiro do 2º Procedimento da Atividade Prática, os resultados da execução do código e a Análise e Conclusão:

**a) Quais as diferenças entre a persistência em arquivo e a persistência em banco de dados?**

Ambos são usados para armazenar e recuperar dados. Na persistência em arquivos os dados são salvos no sistema de arquivos do computador, o que quer dizer que pode ser salvo em diferentes formatos (XML, JSON), a recuperação geralmente envolve a leitura do arquivo inteiro, o controle de acesso é gerenciado pelo sistema operacional e a segurança depende das permissões do arquivo. Em contrapartida, na persistência em banco de dados, os dados são armazenados em tabelas relacionadas em um Sistema de Gerenciamento de Banco de Dados (SGBD), os SGBD oferecem linguagens de consulta como SQL que facilitam a recuperação, os SGBD tem recursos avançados de segurança e controle de acesso que possibilitam definir permissões específicas, etc.

**b) Como o uso de operador *lambda* simplificou a impressão dos valores contidos nas entidades, nas versões mais recentes do Java?**

Trouxe uma forma mais concisa de lidar com a implementação de interfaces funcionais, simplificaram a sintaxe, principalmente ao lidar com interfaces funcionais. Pode ser usado em operações de filtragem, mapeamento, redução entre outros, o que facilita o entendimento do código.

**c) Por que métodos acionados diretamente pelo método main, sem o uso de um objeto, precisam ser marcados como *static*?**

O método main é o ponto de entrada da aplicação, e é chamado diretamente pela máquina virtual Java (JVM) ao iniciar o programa. Para ser invocado sem a necessidade de criar uma instância da classe que contém o método main, ele deve ser marcado como static.

## **Conclusão**

Este projeto tem como objetivo desenvolver um aplicativo em Java que se conecta a um banco de dados SQL usando o middleware JDBC, seguindo o padrão DAO para gerenciar a persistência dos dados. O foco está em implementar o mapeamento objeto-relacional, organizar o código de maneira que facilite a manutenção, e aplicar conceitos de herança e polimorfismo em classes que representam pessoas físicas e jurídicas.

A prática é essencial para o desenvolvimento de sistemas com persistência em banco de dados, proporcionando a experiência de manipular dados com classes DAO e testar operações CRUD, como inclusão, alteração, consulta e exclusão de registros.