

Atividade: Relatório de Prática

Nome: Jamison Queiroz **Matrícula:** 202208101127

Curso: Desenvolvimento Full Stack

Turma: 2022.4

Disciplina: Back-end Sem Banco Não Tem **Professor:** Simone Ingrid Monteiro Gama



Sumário

1.	INTRODUÇÃO	3
	OBJETIVO	
	SOFTWARE UTILIZADO	
	PROCEDIMENTOS	
	4.1 Mapeamento Objeto-Relacional e DAO	
	4.1.1 Análise e Conclusão	
	4.2 Alimentando a Base	
	4.2.1 Análise e Conclusão	
5.	CONCLUSÃO	



1. INTRODUÇÃO

Criação de aplicativo Java, com acesso ao banco de dados SQL Server através do middleware JDBC.

2. OBJETIVO

- **2.1.** Implementar persistência com base no middleware JDBC.
- 2.2. Utilizar o padrão DAO (Data Access Object) no manuseio de dados.
- 2.3. Implementar o mapeamento objeto-relacional em sistemas Java.
- **2.4.** Criar sistemas cadastrais com persistência em banco relacional.
- **2.5.** No final do exercício, o aluno terá criado um aplicativo cadastral com uso do SQL Server na persistência de dados.

3. SOFTWARE UTILIZADO

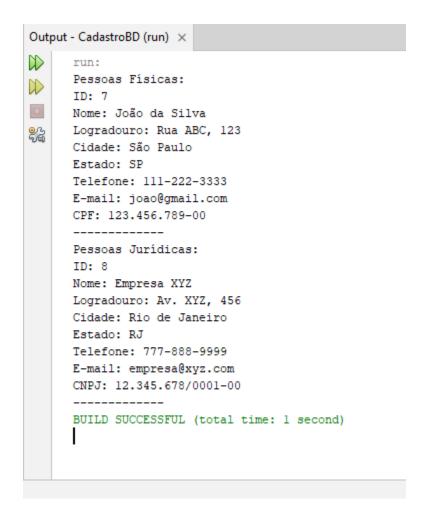
- 3.1. SQL Server Management Studio 19
- **3.2.** Microsoft SQL Server Express 2022
- 3.3. NetBeans 18
- **3.4**. JDK 8

4. PROCEDIMENTOS

4.1. Mapeamento Objeto-Relacional e DAO

Nesta etapa, foi criado um projeto denominado "CadastroBD" e configurado uma conexão com o banco de dados Microsoft SQL Server Express 2022 usando o JDBC. Foram desenvolvidas as classes DAO, responsáveis por interagir com o banco de dados, assim como as classes para gerenciar a conexão com o banco e sequências necessárias para realizar transações. Também foi criado um arquivo de teste para verificar a funcionalidade do sistema, cujo resultado pode ser visualizado abaixo:





As tabelas utilizadas foram as mesmas da disciplina *Vamos Manter as Informações* conforme imagem abaixo:



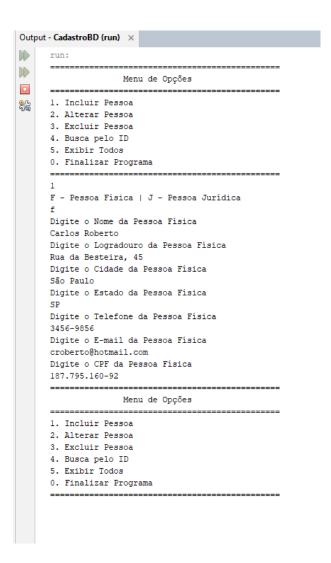
4.1.1 Análise e Conclusão:

- a) Qual a importância dos componentes de middleware, como o JDBC?
 - Eles facilitam a comunicação entre a aplicação e o banco de dados permitindo a portabilidade do aplicativo entre diversos bancos precisando apenas pequenos ajustes dentre outras funcionalidades.
- b) Qual a diferença no uso de Statement ou PreparedStatement para a manipulação de dados?
 - Utilizamos parâmetros nas consultas do PreparedStatement além dele assegurar um tratamento seguro desses parâmetros aumentando a segurança das consultas o que não ocorre no Statement que executa consultas fixas.
- c) Como o padrão DAO melhora a manutenibilidade do software?
 - Dentre os benefícios do DAO podemos citar a reutilização do código evitando a duplicação de procedimentos nas operações com o banco e a facilidade de implementação de testes unitários usando mocks isolando o banco real.
- d) Como a herança é refletida no banco de dados, quando lidamos com um modelo estritamente relacional?
 - Utilizamos tabelas com chave estrangeira para a tabela que seria a classe pai sendo utilizado a cardinalidade 1x1.

4.2. Alimentando a Base

Nesta etapa, foi realizado o preenchimento do banco de dados por meio do terminal. Configuramos o sistema para permitir o registro de informações e disponibilizamos um menu de opções, conforme ilustrado na imagem abaixo:





4.2.1 Análise e Conclusão:

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

Temos como algumas das diferenças a forma de guardar a informação aonde persistência em banco armazena as informações em um Sistema de Gerenciamento de Banco de Dados e a persistência em arquivo grava as informações em um arquivo em um local do disco além possuírem forma de recuperação de bem diferentes sendo a



- persistência em banco tendo uma gama maior de possibilidade de manipulação dos dados (SQL, API) e a persistência em arquivo é leitura e escrita direta no arquivo.
- b) Como o uso de operador lambda simplificou a impressão dos valores contidos nas entidades, nas versões mais recentes do Java?
 - Antes se usava o for ou foreach para impressão de valores de coleções, com o operador lambda ficou mais simples tornando o código mais conciso e legível onde essa expressão pode ser passada como argumento para funções de ordem superior.
- c) d) Por que métodos acionados diretamente pelo método main, sem o uso de um objeto, precisam ser marcados como static?
 - Porque o método main é static e um método acionado diretamente por ele pertence à classe em vez de pertencer a instâncias específicas dessa classe.

5. CONCLUSÃO

Nesta prática, realizamos com sucesso a conexão da aplicação Java ao banco de dados MSSQL e implementamos um sistema de cadastro que permite realizar operações CRUD (Create, Read, Update, Delete) no banco de dados. O código completo utilizado nesta prática, juntamente com uma cópia do banco de dados, estará disponível no GitHub.