



Estácio

Campus: Conceição - São Paulo/SP

GRADUAÇÃO DESENVOLVIMENTO FULL STACK

Disciplina: Nível 3: Back-end Sem Banco Não Tem

Turma: 2022.03 - Mundo 3

Aluna: Fernanda G. Vargas

Matrícula: 202208836305



Objetivos 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. As práticas devem ser feitas individualmente.

1º Procedimento | Mapeamento Objeto-Relacional e DAO

1- Título da Prática;

Mapeamento Objeto-Relacional e DAO

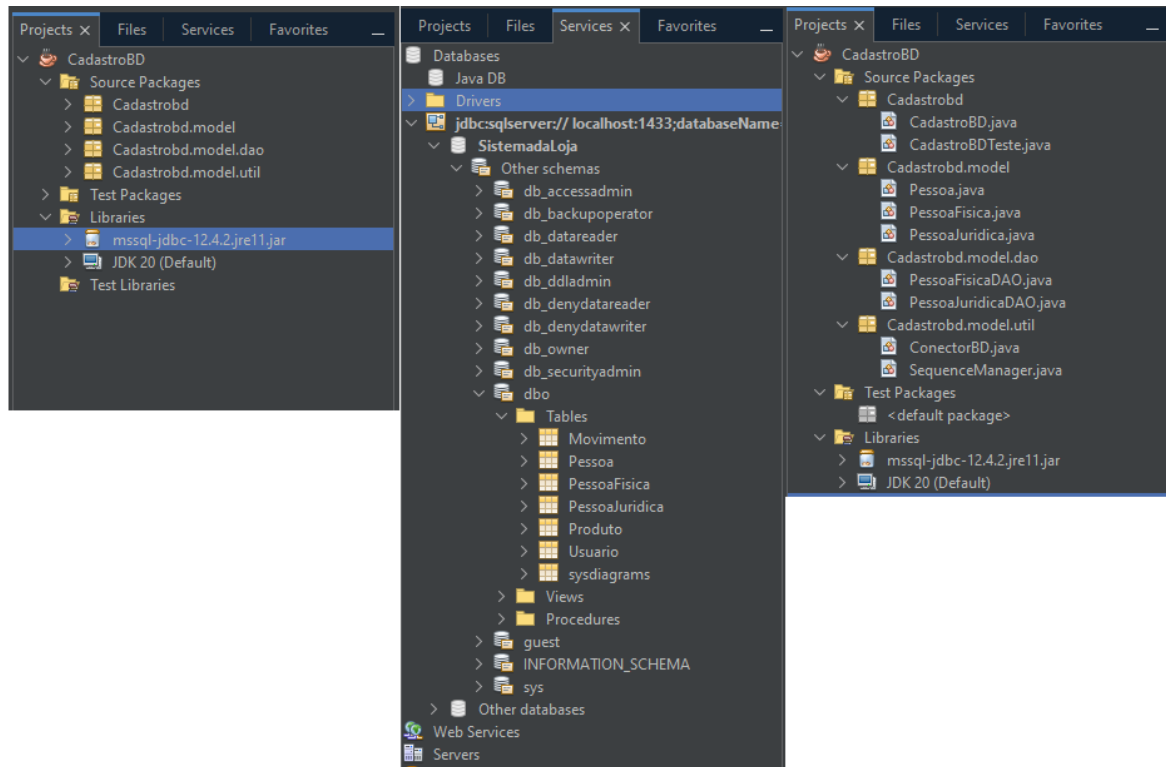
2- Objetivo da Prática;

Criar novo projeto modelo Ant pela IDE Apache NetBeans adicionar o drive JDBC para SQL Services na libraries pela opção Jar. Baixar e descompactar arq zip. Configurar conexão ao BD criano nível 2 para iniciar projeto de CadastroBD.

3- Todos os códigos solicitados neste roteiro de aula;

Disponível: <https://github.com/FerGVargas>

4- Os resultados da execução dos códigos também devem ser apresentados;



5- Análise e Conclusão:

5.a- Qual a importância dos componentes de middleware, como o JDBC?

O Java Database Connectivity é importante e necessária para desenvolver aplicativos facilitando a comunicação e integração de diferentes partes dos sistema.

5.b- Qual a diferença no uso de Statement ou PreparedStatement para a manipulação de dados?

São interfaces no JDBC usadas para executar instruções SQL em BD. As principais diferenças: **Statement** Estática; Não aceita parâmetros; Cada execução é compilada e otimizada pelo BD no momento da execução; Desempenho mais lento se executada repetidas vezes; Vulnerável a ataques de injeção de SQL se os parâmetros da consulta não forem devidamente tratados; Concatenar diretamente valores de entrada na string SQL pode expor o código a riscos de segurança; Menos legível e mais difícil de manter, especialmente em consultas SQL complexas; e Requer manipulação manual de caracteres especiais e formatação.

PreparedStatement A consulta SQL é dinâmica; Aceita parâmetros; Consulta pré-compilada e otimizada pelo BD uma vez; Pode ser reutilizada com diferentes conjuntos de parâmetros; Melhor desempenho para consultas repetidas com diferentes parâmetros, pois evita a recompilação; Usa parâmetros de forma segura; Os parâmetros são tratados separadamente da instrução SQL principal; e Protege automaticamente contra injeção de SQL, tornando a aplicação mais segura.

5.c- Como o padrão DAO melhora a manutenibilidade do software?

O DAO é a camada do sistema (pacotes, classes e métodos) que abstrai todo o acesso ao BD separadamente da lógica de negócio da aplicação. É no DAO que implementamos os métodos do CRUD (Create – Read – Update – Delete). A persistência de dados tem como garantia de que um dado foi salvo e que poderá ser recuperado quando necessário no futuro. Assim a manutenibilidade de software pode ser facilmente modificada ou adaptada e quantificada em termos do tempo médio requerido para efetivar a revisão do software para eliminar erros.

5.d- Como a herança é refletida no banco de dados, quando lidamos com um modelo estritamente relacional?

Herança é um conceito de BD orientado a objetos onde a classe filho recebe a herança de uma classe pai. No BD relacional representamos esse metodo atraves da chave estrangeira vinculando uma entidade a outra para receber os dados.

2º Procedimento | Alimentando a Base

1- Título da Prática;

Alimentando a Base

2- Objetivo da Prática;

Alterar o método main para apresentação de opções (incluir, altera, excluir, oter, obter todos e sair) através do DAO.

3- Todos os códigos solicitados neste roteiro de aula;

Disponivel: <https://github.com/FerGVargas>

4- Os resultados da execução dos códigos também devem ser apresentados;

```

run:
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====

1
F - Pessoa Fisica | J - Pessoa Juridica
f
Inserir dados da Pessoa Fisica:
Id:
10
Nome:
Fernanda
Logradouro:
Rua Jose Alves
Cidade:
São Paulo
Estado:
Sp
Telefone:
11 991478978
Email:
fernanda@hotmail.com
Cpf:
12345678910
=====

1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====

1
F - Pessoa Fisica | J - Pessoa Juridica
j
Inserir dados da Pessoa Juridica:
Id:
10010
Nome:
BatMoveis SA
Logradouro:
Av. Ipiranga
Cidade:
Sao Paulo
Estado:
SP
Telefone:
11 5689-1020
Email:
batmoveis@batmoveis.br
CNPJ:
12123456000110
=====

run:
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====

5
F - Pessoa Fisica | J - Pessoa Juridica
j
=====
Exibindo dados de todas as Pessoas Juridicas...
=====
Id: 6
Nome: JJC
Logradouro: Rua A, Centro
Cidade: Riacho do Sul
Estado: PA
Telefone: 1212-1212
Email: jjc@riacho.com
CNPJ: null
=====

```

5- Análise e Conclusão:

5.a- Quais as diferenças entre a persistência em arquivo e a persistência em banco de dados?

Persistência em Arquivo Dados armazenados em arquivo(txt, binário, etc); Eficiente para pequenas quantidades de dados ou para sistemas simples; O desempenho pode degradar-se à medida que os dados e a complexidade do sistema aumentam; Pesquisa e consulta exigem leitura do arquivo e processamento manual tornando difícil executar consultas complexas ou buscar dados de maneira eficiente; Difícil gerenciar sistemas baseados em arquivos, principalmente em ambientes multiusuários; Controle de acesso limitado e depende do sistema operacional dificultando a segurança; Difícil alterar a estrutura de dados principalmente com muitos arquivos. Persistência em Banco de

Dados armazenados em tabelas. Cada tabela representa um tipo de entidade, e as relações entre as tabelas são definidas para manter a integridade e a consistência dos dados; Eficiente para grandes volumes de dados e consultas complexas. Os SGBD são otimizados para consultas e transações em larga escala; Consultas SQL permitem buscar, filtrar e ordenar dados de maneira eficiente. Índices e otimizações internas dos SGBD facilitam a recuperação rápida de informações; Suporte nativo a transações, operações consistentes e seguras em ambientes multiusuários; Possível definir permissões em nível de tabela e campo, proporcionando maior segurança.

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

Como o operador lambda usando o método `forEach` o código fica mais simples e o resultado é mais objetivo.

5.c- Por que métodos acionados diretamente pelo método `main`, sem o uso de um objeto, precisam ser marcados como `static`?

O método `main` é estático e pertence a classe e é o ponto de entrada para o programa, deve ser estático porque é invocado pelo sistema antes que qualquer instância da classe seja criada.