Persistência de Dados - JAVA Data Acess Aobject (DAO)

Um projeto de implementação por camadas é organizado em 5 partes: execução, negocio, persistência, visão e vo.

- Execução: é camada responsável por realizar a execução do código realizando a interação com o usuário.
- Negócio: é a camada onde deve-se aplicar as regras de negócio e validação dos dados, comumente interligado coma camada de persistência.
- Persistência: camada que responsável por realizar a conexão com o banco de dados para realizar as operações de inclusão, busca e alteração.
- Visão: camada responsável por implementar a interface com o usuário.
- VO: camada responsável por conter as classes das entidades.

Exemplo: AlunoVO AlunoNegocio AlunoDAO Principal Academico Controle de Execução Validação de dados e Persistência e Arm azenam ento dos Interface com Usuário regras de negócio recuperação dos dados do banco de dados A partir dos dados digitados Recebe AlunoVO e Recebe AlunoVO Armazena dados na tabela aluno da base de pelo usuário, cria instância de valida dados e regras Cria script SQL AlunoVO e envia para dados academico de negócio Extrai dados do Inclusão AlunoNegocio Envia AlunoVO para AlunoVO AlunoDAO Inseri no script SQL Envia script SQL para o Banco de Dados Idem Idem Idem Alteração Idem Obtém a matrícula do aluno a Recebe a matrícula Recebe a matrícula Exclui o registro ser excluído do usuário Envia matrícula para Cria script SQL correspondente Envia a matrícula para AlunoDAO Inseri a matrícula no Exclusão AlunoNegocio script SQL Envia script SQL para o Banco de Dados Obtém a matrícula do aluno a Consulta a tabela aluno Recebe a matrícula Recebe a matrícula ser excluído do usuário Envia matrícula para Cria script SQL de para localizar o aluno Envia a matrícula para AlunoDAO consulta correspondente à AlunoNegocio Inseri a matrícula no matrícula no script Recebe AlunoVO de script SQL Devolve dados (um ou Recebe AlunoVO de AlunoDAO Envia script SQL para zero registros) Pesquisar por AlunoNegocio Devolve AlunoVO o Banco de Dados Matrícula Exibe dados contidos em para Principal Recebe dados do banco AlunoVO Monta AlunoVO com dados do registro recebido Devolve AlunoVO para Aluno Negocio

Explicação(caso não funcione, remova o segundo monitor para o áudio funcionar):



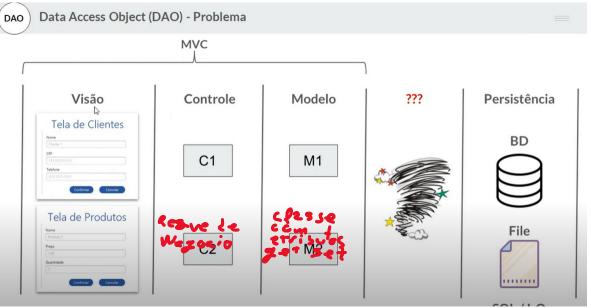
Implement...

por...

Gravação de áudio iniciada: 15:35 domingo, 20 de junho de 2021

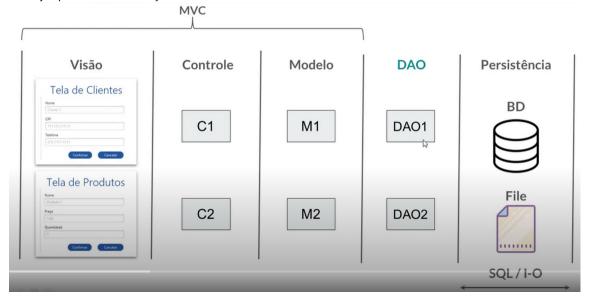
Persistência de Dados - Java DAO (Data Access Object)





Arquitetura MVC - Modelo Controle e Visão.

Dado a imagem acima, onde deveriamos adicionar o SQL para conversar com o banco de dados? A solução para isto seria a criação da camada DAO.



Para cada classe de Modelo teremos uma classe DAO para relacionar com o banco de dados. Exemplo:

Modelo: Cliente.java

```
public class Cliente {
      private String nome;
      private String cpf;
      private String telefone;
      public String getNome() {
          return nome;
      public void setNome (String nome) {
          this.nome = nome;
      public String getCpf() {
          return cpf;
      public void setCpf(String cpf) {
          this.cpf = cpf;
      public String getTelefone() {
          return telefone;
      public void setTelefone(String telefone) {
          this.telefone = telefone;
Y = 9
```

DAO: ClienteDAO.java

```
public class MainSelect {
    public static void main(String[] args) {
       ClienteDAO clienteDAO = new ClienteDAO();
       List<Cliente> lista = clienteDAO.listar();
        for (Cliente cliente : lista)
            System.out.println(cliente);
public class MainInsert {
   public static void main(String[] args) {
       Cliente cliente = new Cliente("Cliente 4", "444.444.444-44", "(44) 4444-4444");
       ClienteDAO clienteDAO = new ClienteDAO();
        clienteDAO.inserir(cliente);
public class MainUpdate {
   public static void main(String[] args) {
       Cliente cliente = new Cliente(1, "Cliente 0", "000.000.000-00", "(00) 0000-0000");
       ClienteDAO clienteDAO = new ClienteDAO();
       clienteDAO.alterar(cliente);
public class MainDelete {
   public static void main(String[] args) {
       ClienteDAO clienteDAO = new ClienteDAO();
        clienteDAO.remover(1);
```









CREATE

READ

UPDATE

DELETE

C

R

U

D