

JDBC

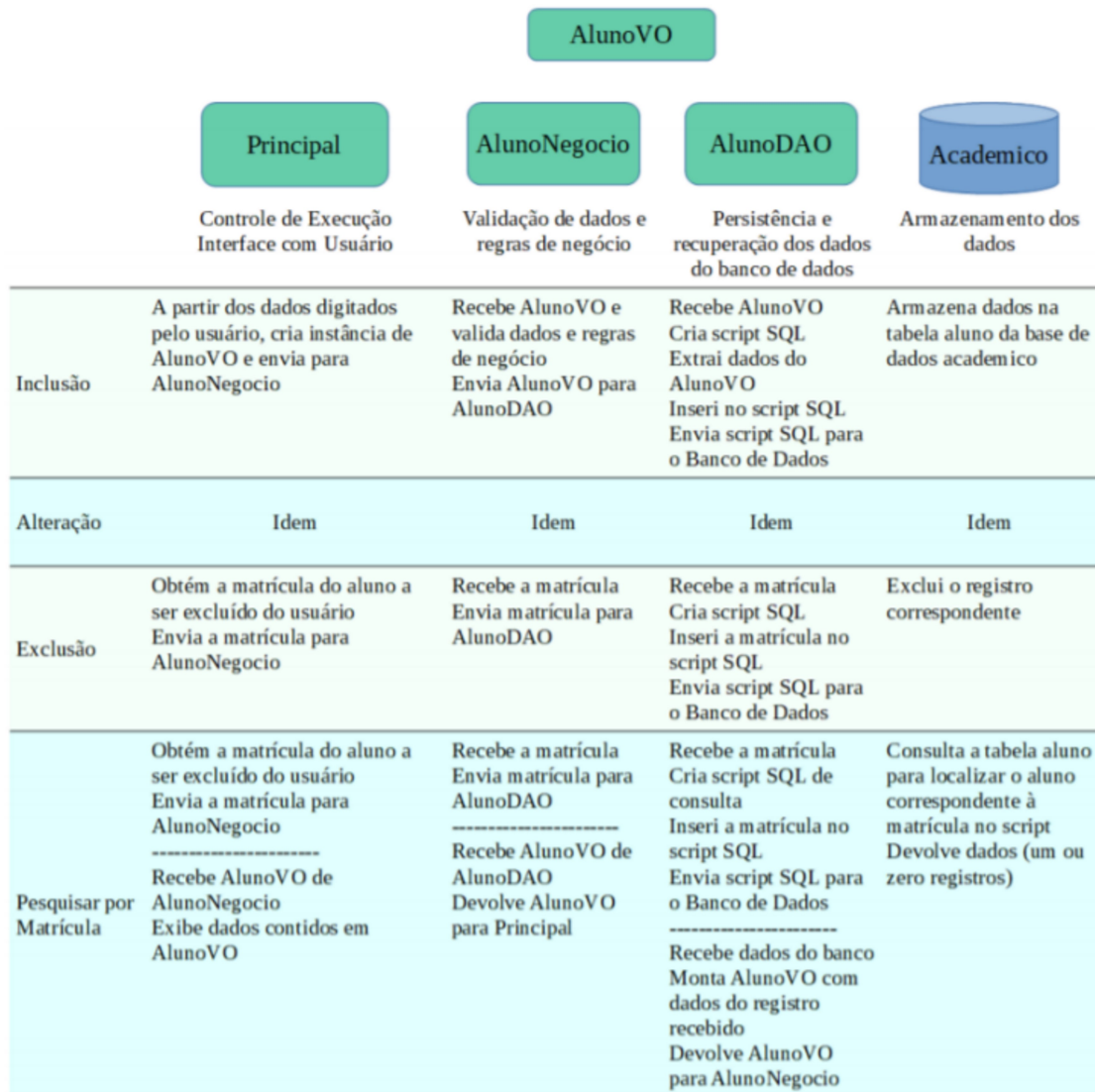
domingo, 20 de junho de 2021 15:22

Persistência de Dados - JAVA Data Access 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:



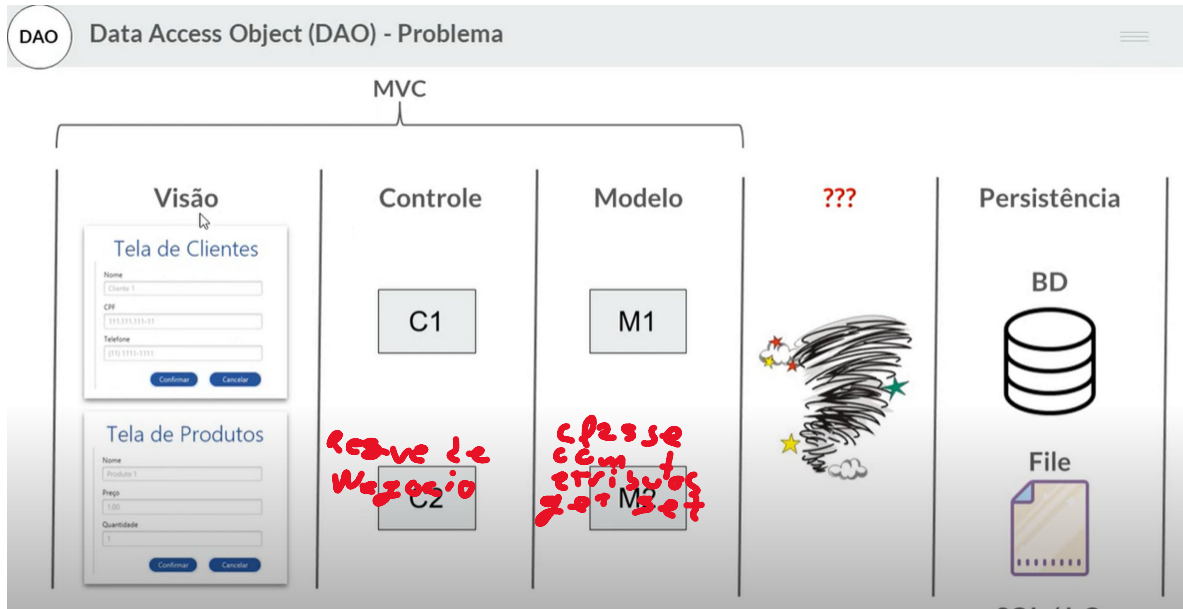
Explicação(caso não funcione, remova o segundo monitor para o áudio funcionar):



Implement...
por...

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

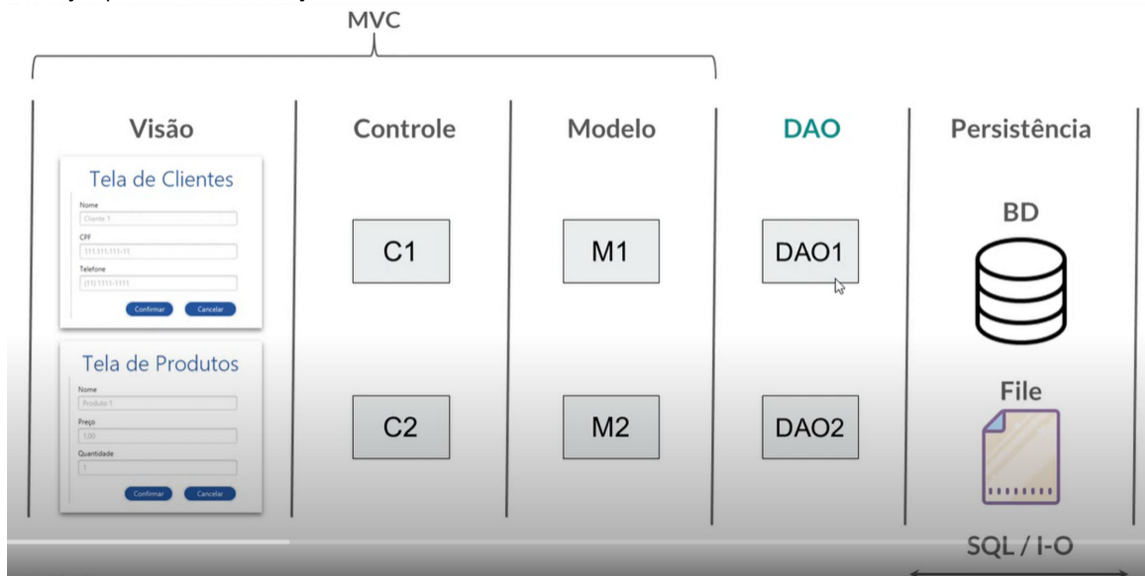
[Persistência de Dados - Java DAO \(Data Access Object\)](#)



Arquitetura MVC - Modelo Controle e Visão.

Dado a imagem acima, onde deveríamos 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;
    }
}
```

DAO: ClienteDAO.java

```
public class ClienteDAO {

    private Connection connection;

    public boolean inserir(Cliente cliente){
        String sql = "INSERT INTO CLIENTES (nome, cpf, telefone) VALUES (?, ?, ?)";
        ...
    }

    public boolean alterar(Cliente cliente){
        String sql = "UPDATE CLIENTES SET nome=?, cpf=?, telefone=? WHERE ID=?";
        ...
    }

    public boolean remover(Cliente cliente){
        String sql = "DELETE FROM CLIENTES WHERE ID=?";
        ...
    }

    public List<Cliente> listar(Cliente cliente){
        String sql = "SELECT * FROM CLIENTES";
        ...
    }

    public Cliente buscar(Cliente cliente){
        String sql = "SELECT * FROM CLIENTES WHERE ID=?";
        ...
    }
}
```

```
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