

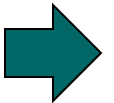
Java DataBase Connectivity (JDBC)

Vinicius Rosalen

Parte III

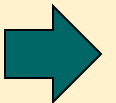
O padrão de projeto DAO

- Blz,
 - Vimos um monte de coisas interessantes sobre persistência de dados com Java...
- Também aprendemos como podemos modelar a nossa aplicação em camadas e utilizar uma simplificação do modelo de comunicação MVC entre elas...
- Será que eu consigo usar esse conceito para melhorar ainda mais a estrutura do meu sistema para acessar uma banco de dados???....
 - Como utilizar JDBC de forma organizada?
 - Como realizar a persistência dos meus objetos de negócio?



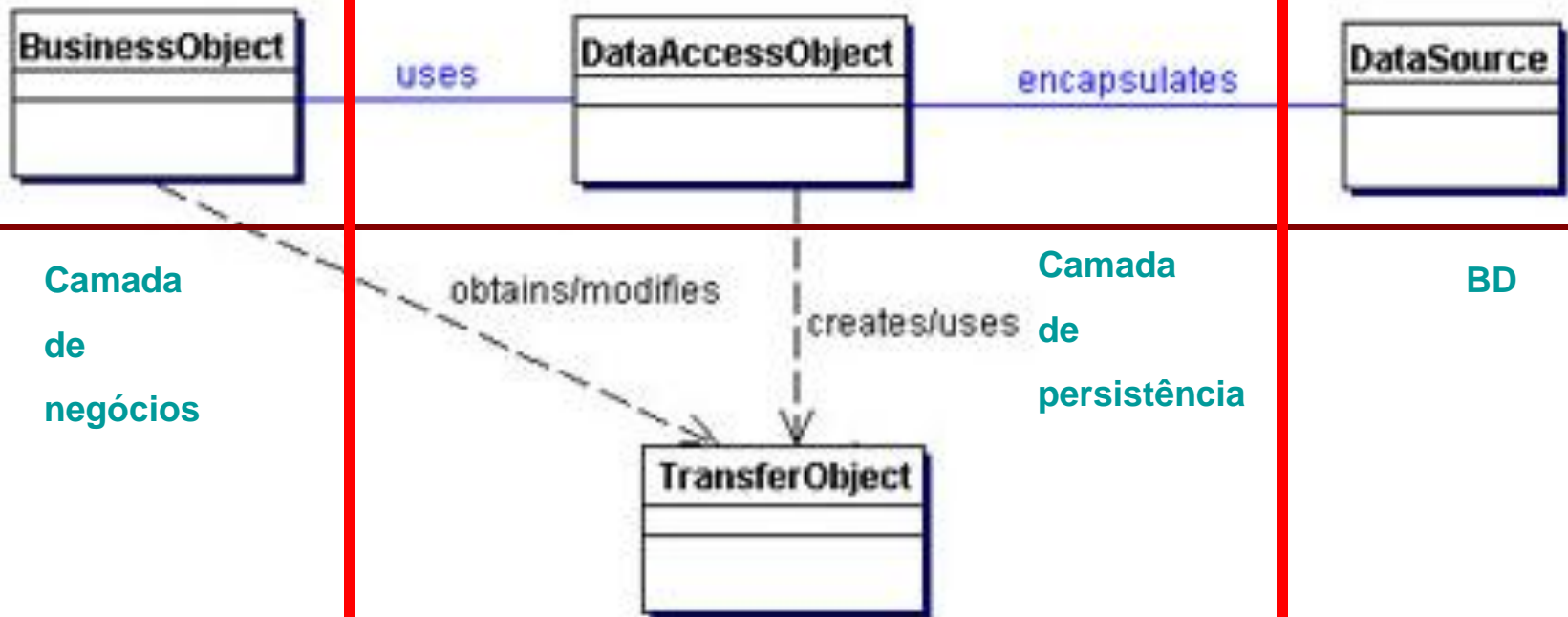
O padrão de projeto DAO

- Podemos usar o padrão de projetos DAO
 - Data Access Object (DAO)
- DAO é um padrão de projeto responsável por encapsular operações de acesso a dados em classes dedicadas exclusivamente a isso.....
 - Com isso evita-se que o código de interação com o banco de dados fique espalhado pelo código da sua aplicação.
- Também permite que a aplicação facilmente implemente suporte a diversos bancos de dados
 - Sem impactar no restante do código.



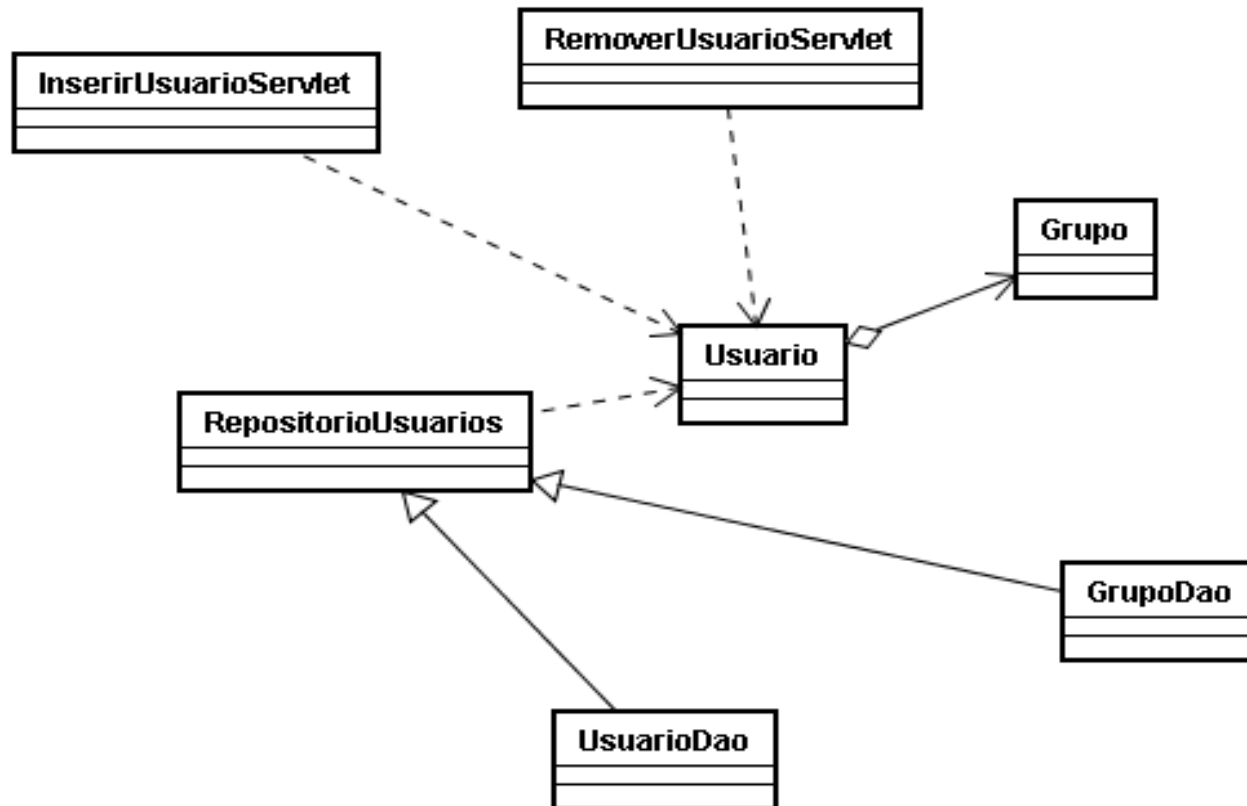
O padrão de projeto DAO

- **Estruturação**
 - A figura abaixo mostra um diagrama de classes que representa os relacionamentos no padrão de projeto DAO



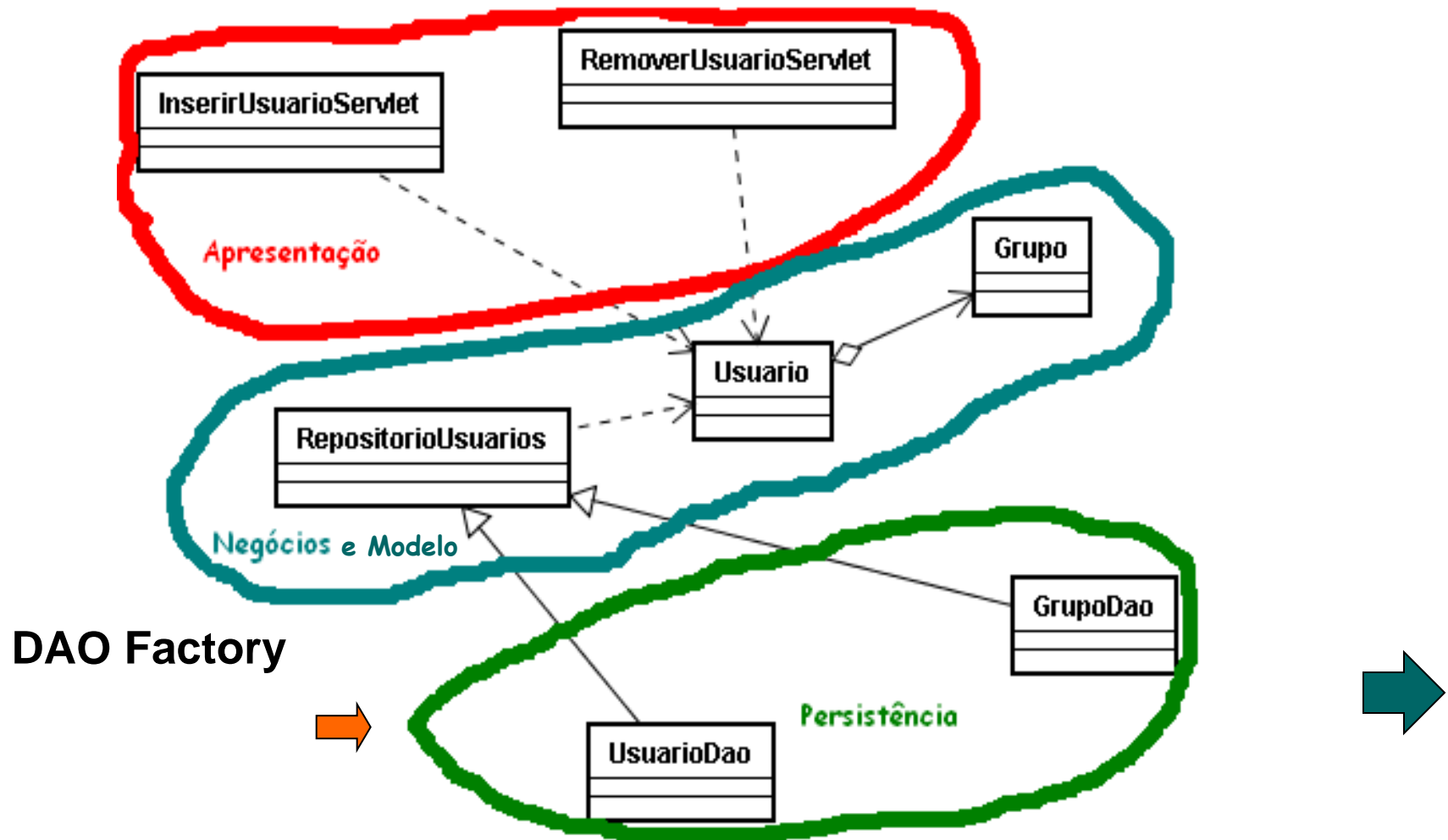
O padrão de projeto DAO

- Vamos analisar um exemplo...
 - ➔ O diagrama abaixo mostra os objetos em uma aplicação simples.
 - Neste caso não há separação lógica qualquer.



O padrão de projeto DAO

- Aplicando a técnica de Camadas e o padrão DAO
 - ➡ Vamos agrupar os objetos com responsabilidade semelhante.



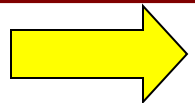
O padrão de projeto DAO

➡ Em alguns projetos pequenos

- Podemos utilizar um “mini” padrão DAO, deixando de criar a DAO Factory

➡ Desta forma, para cada classe de negócio que criarmos (Cliente, Produto, Venda, etc).....

➡ Criaremos também uma versão DAO da mesma (ClienteDAO, ProdutoDAO, VendaDAO,
– Onde iremos ter os códigos de manipulação com base de dados

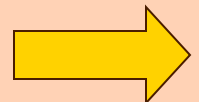


O padrão de projeto DAO

- O objetivo é separar o domínio do problema
 - De sua representação física no banco de dados.
- Fazendo isso, para armazenar um novo cliente na base de dados por exemplo...
 - ... poderíamos fazer o seguinte:

```
public void novoCliente() {  
    Cliente cli = new Cliente ();  
    cli.setNome ("Fulano de tal");  
    cli.setEndereco ("Rua tal de tal, S/N");  
    cli.setFone ("5555 - 1234");  
  
    // Armazena o novo cliente no banco de dados  
    ClienteDAO cliDAO = new ClienteDAO();  
    cliDAO.Create (cli);  
}
```

Pseudo código demonstrando o padrão DAO

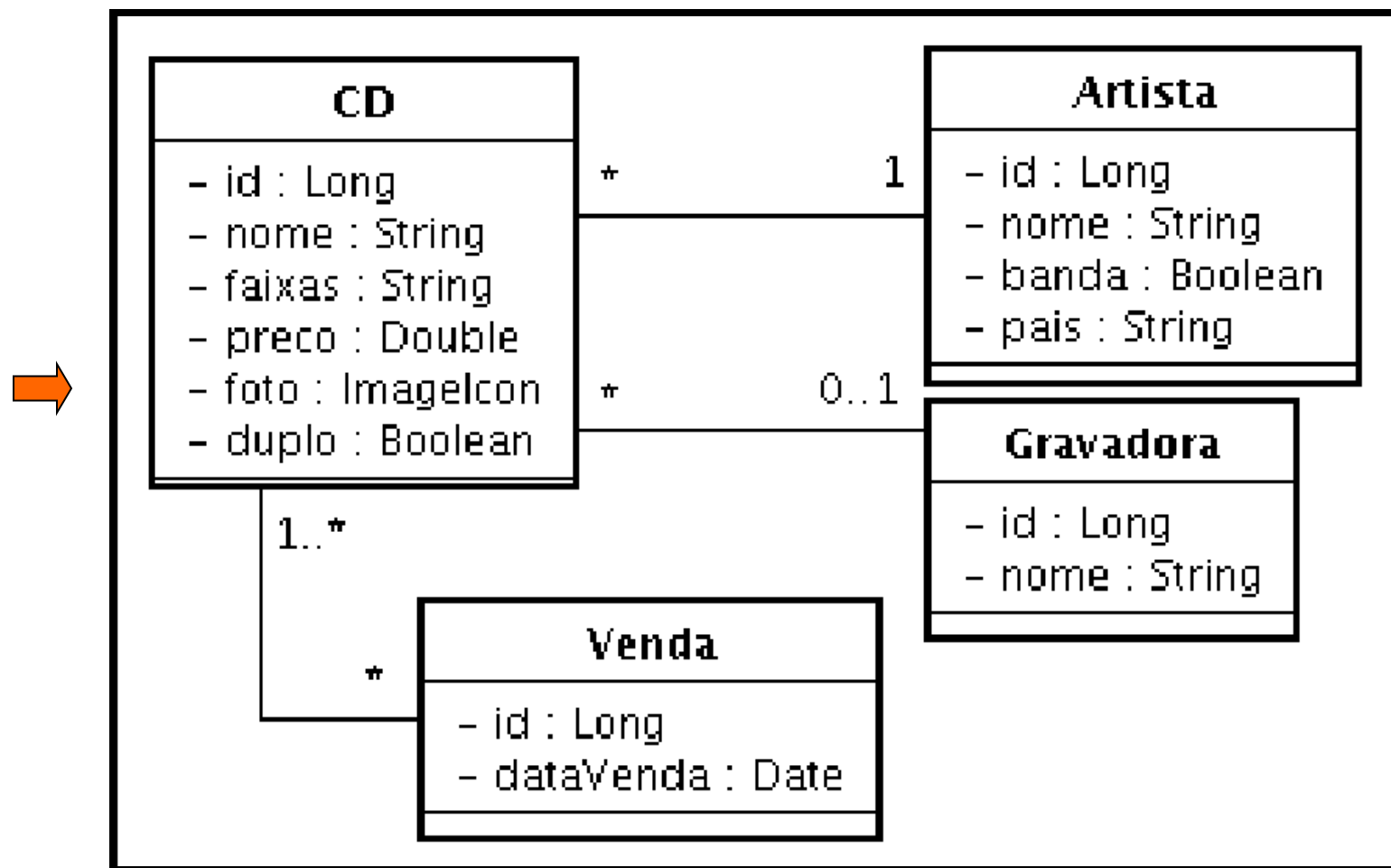


O padrão de projeto DAO

- Importante Reforçar: Cada DAO é responsável pelo CRUD
 - Acrônimo em inglês para as operações básicas de manipulação de dados em linguagem SQL: CReate, Uppdate, Delete).

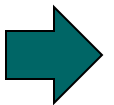
O padrão de projeto DAO

➡ Lembrem da loja de CD JavaDiscos que usamos anteriormente...



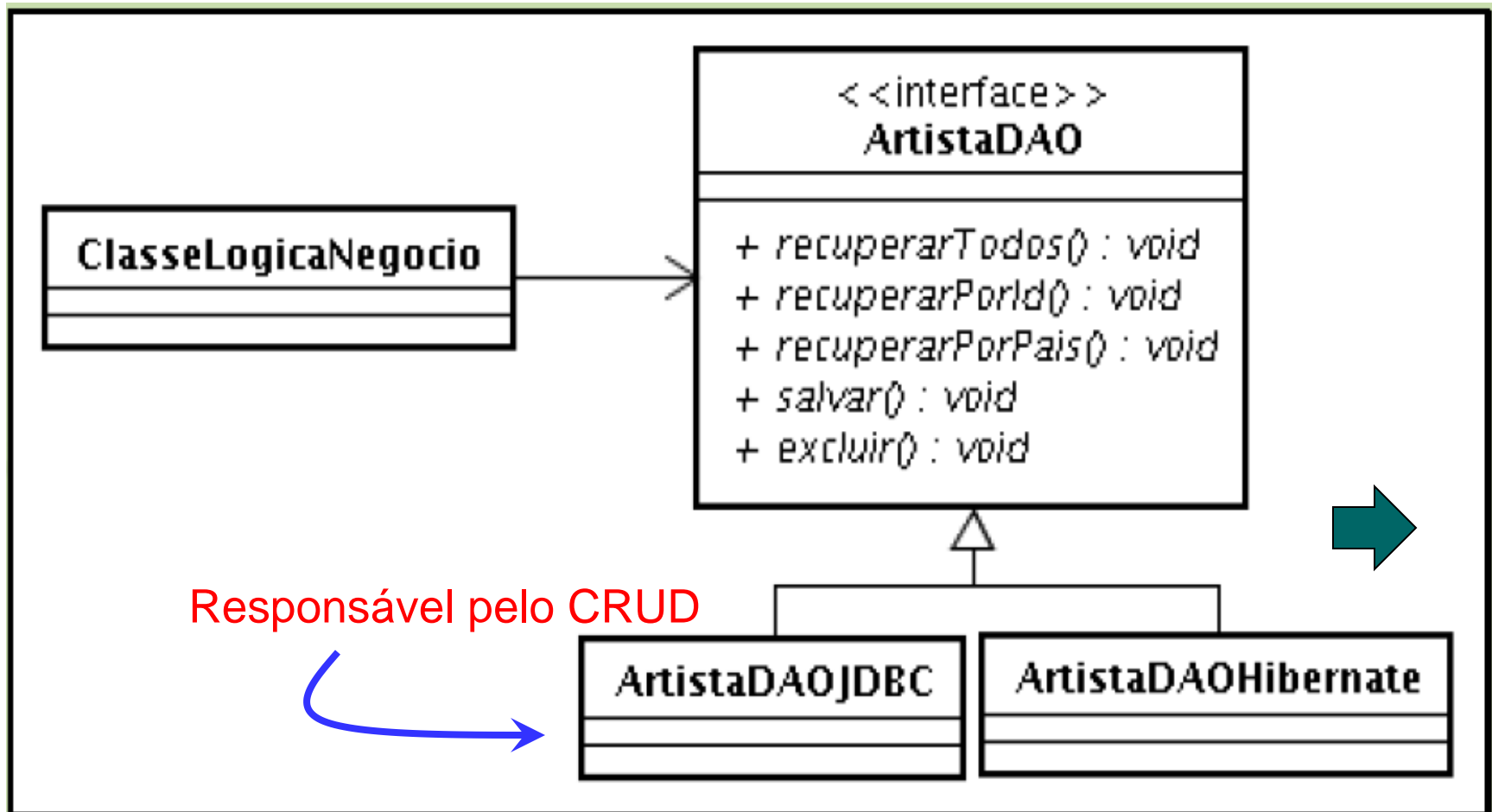
O padrão de projeto DAO

- Adotando o DAO na loja de CD JavaDiscos
 - Delega-se uma classe separada a tarefa de persistência de uma classe de negócio;
- Para cada classe de domínio há:
 - ➡ Uma interface que define as operações do DAO para aquela classe;
 - ➡ Uma implementação para cada tecnologia de persistência (JDBC, Hibernate, etc.).
- Por exemplo, se quisermos criar a persistência da classe Artista da loja de CD JavaDiscos
 - Podemos ter a seguinte estruturação...



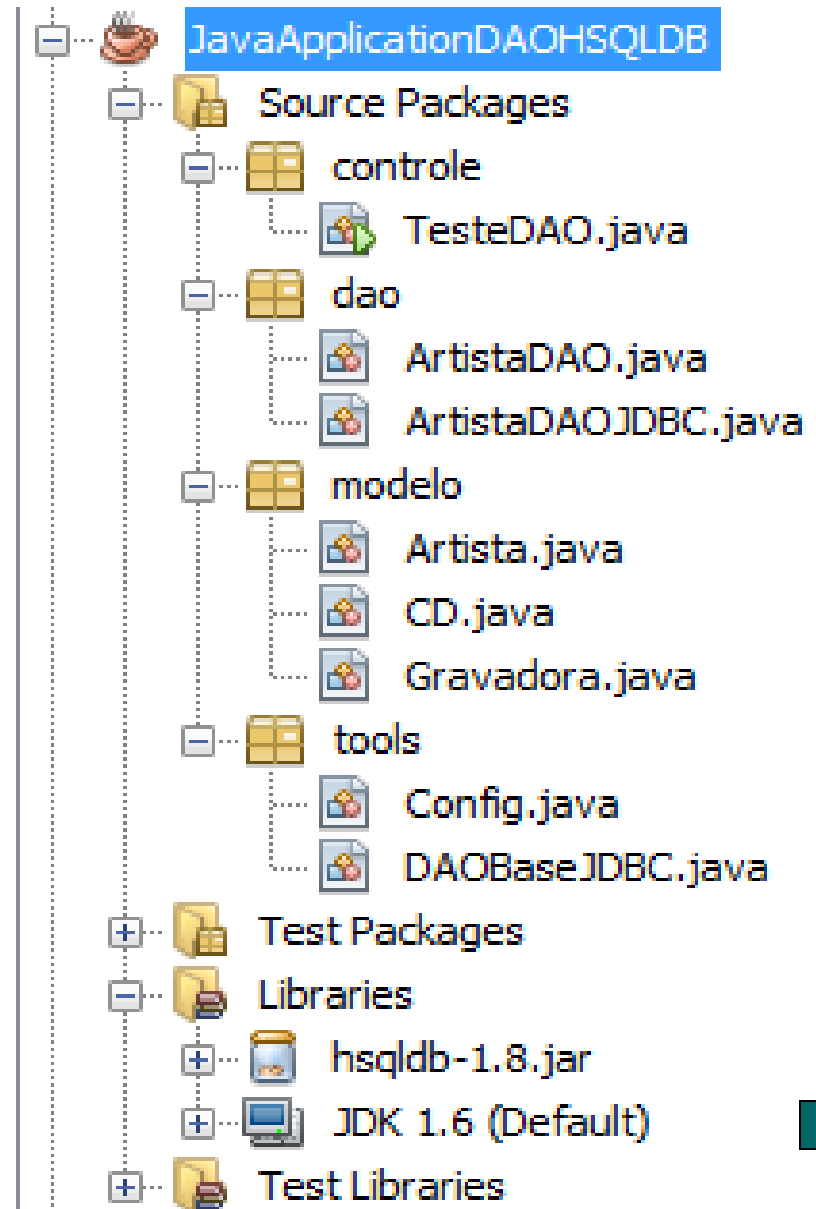
O padrão de projeto DAO

- Estruturação para a persistência da classe Artista da loja de CD JavaDiscos



O padrão de projeto DAO

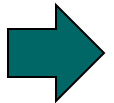
- Projeto no NetBeans
 - Vamos fazer juntos....



O padrão de projeto DAO

Config.java

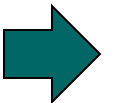
```
public interface Config {  
    public static final String NOME_DRIVER = "org.hsqldb.jdbcDriver";  
    public static final String BD_URL = "jdbc:hsqldb:hsqldb://localhost/javadiscos";  
    public static final String BD_LOGIN = "sa";  
    public static final String BD_SENHA = "";  
}
```



O padrão de projeto DAO

DAOBaseJDBC.java

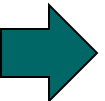
```
public class DAOBaseJDBC {  
  
    protected static Connection conn;  
  
    static {  
        try {  
            // Carrega o driver do HSQLDB e conecta.  
            Class.forName(Config.NOME_DRIVER);  
            conn = DriverManager.getConnection(Config.BD_URL, Config.BD_LOGIN, Config.BD_SENHA);  
        }  
        catch (ClassNotFoundException e) {  
            System.out.println("FATAL: driver não encontrado.");  
            System.exit(1);  
        }  
        catch (SQLException e) {  
            System.out.println("Erro SQL: " + e.getMessage());  
            System.exit(1);  
        }  
    }  
}
```



O padrão de projeto DAO

Artista.java

```
public class Artista {  
    private Long id;  
  
    private String nome;  
  
    private Boolean banda;  
  
    private String pais;
```



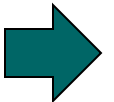
Artista	
💡 id:	BIGINT
💎 nome:	VARCHAR(100)
💎 banda:	BIT
💎 pais:	VARCHAR(50)

```
public Boolean getBanda() {...}  
public void setBanda(Boolean banda) {...}  
public Long getId() {...}  
public void setId(Long id) {...}  
public String getNome() {...}  
public void setNome(String nome) {...}  
public String getPais() {...}  
public void setPais(String pais) {...}  
@Override  
public String toString() {...}
```

O padrão de projeto DAO

ArtistaDAO.java

```
public interface ArtistaDAO {  
    Collection obterTodos();  
  
    void salvar(Artista artista);  
    void excluir(Artista artista);  
}
```



O padrão de projeto DAO

ArtistaDAOJDBC.java

```
public class ArtistaDAOJDBC extends DAOBaseJDBC implements ArtistaDAO {  
  
    public void excluir(Artista artista) {  
        try {  
            if (artista.getNome() != null) {  
                PreparedStatement stmt = conn.prepareStatement("DELETE FROM Artista WHERE nome = ?");  
                stmt.setString(1, artista.getNome());  
                stmt.executeUpdate();  
            }  
        } catch (SQLException e) {  
            System.out.println("Erro SQL: " + e.getMessage());  
            System.exit(1);  
        }  
    }  
}
```



O padrão de projeto DAO

ArtistaDAOJDBC.java

```
public void salvar(Artista artista) {
    PreparedStatement stmt;

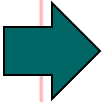
    try {
        if (artista.getId() == null) {
            stmt =
                conn.prepareStatement(
                    "INSERT INTO Artista (nome, banda, pais) VALUES (?, ?, ?)");
        } else {
            stmt =
                conn.prepareStatement(
                    "UPDATE Artista SET nome = ?, banda = ?, pais = ? WHERE id = ?");
            stmt.setLong(4, artista.getId());
        }

        stmt.setString(1, artista.getNome());
        stmt.setBoolean(2, artista.getBanda().booleanValue());
        stmt.setString(3, artista.getPais());
        stmt.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Erro SQL: " + e.getMessage());
        System.exit(1);
    }
}
```

O padrão de projeto DAO

ArtistaDAOJDBC.java

```
public Collection obterTodos() {  
    Set set = new HashSet();  
    ResultSet rset = null;  
    try {  
        PreparedStatement stmt =  
            conn.prepareStatement("SELECT id, nome, banda, pais FROM Artista");  
        rset = stmt.executeQuery();  
        while (rset.next()) {  
            Artista artista = new Artista();  
            artista.setId(new Long(rset.getLong("id")));  
            artista.setNome(rset.getString("nome"));  
            artista.setBanda(new Boolean(rset.getBoolean("banda")));  
            artista.setPais(rset.getString("pais"));  
            set.add(artista);  
        }  
    } catch (SQLException e) {  
        System.out.println("Erro SQL: " + e.getMessage());  
        System.exit(1);  
    }  
    return set;  
}
```



O padrão de projeto DAO

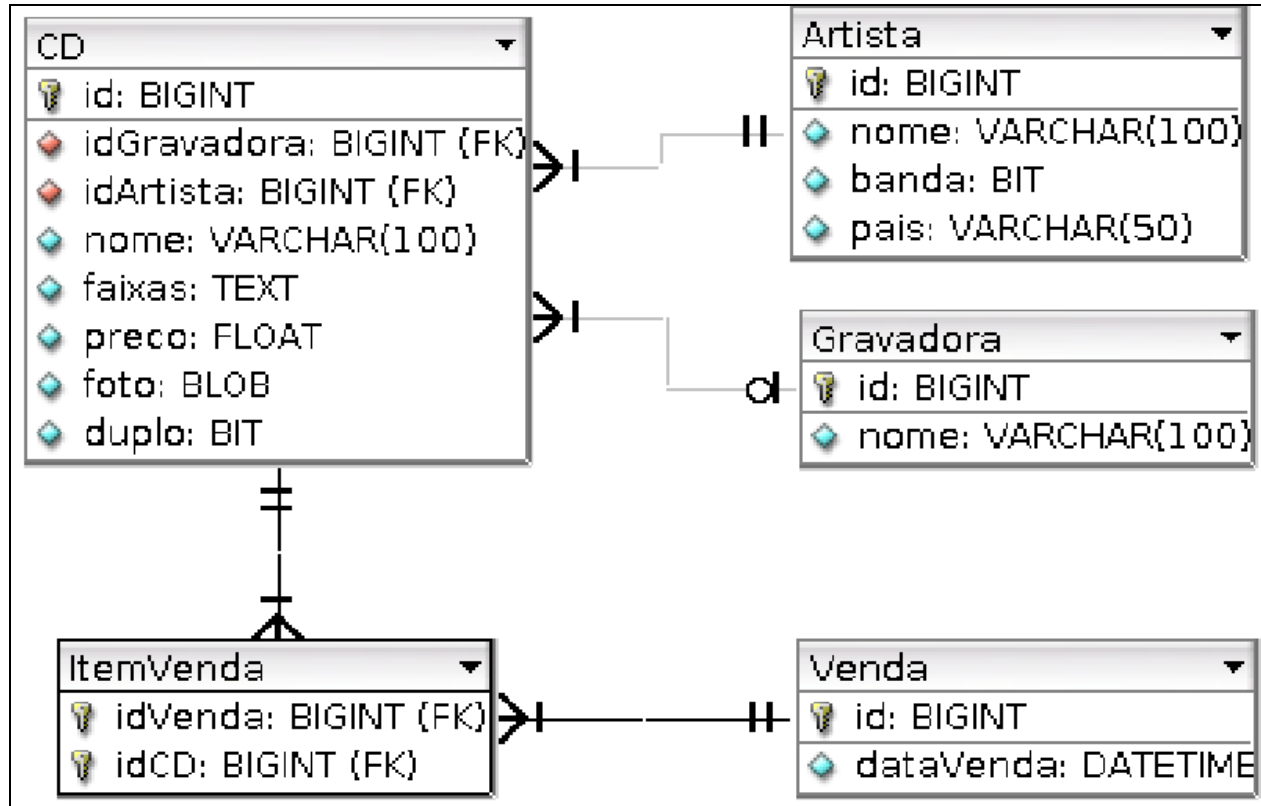
TesteDAO.java

```
public class TesteDAO {  
  
    public static void main(String[] args) {  
  
        ArtistaDAO dao = new ArtistaDAOJDBC();  
  
        Artista artista = new Artista();  
        artista.setNome("Men at Work");  
        artista.setBanda(true);  
        artista.setPais("Australia");  
        System.out.println("Salvando " + artista.getNome());  
        dao.salvar(artista);  
  
        Artista artista2 = new Artista();  
        artista2.setNome("Pancadão");  
        artista2.setBanda(false);  
        artista2.setPais("Braziiilzilzill");  
        System.out.println("Salvando " + artista2.getNome());  
        dao.salvar(artista2);  
    }  
}
```

```
        System.out.println("=====Imprimindo...=====");  
        Collection artistas = dao.obterTodos();  
        for(Object autor: artistas){  
            System.out.println((Artista)autor);  
        }  
  
        System.out.println("\nExcluindo " + artista.getNome());  
        dao.excluir(artista);  
        artistas = dao.obterTodos();  
        System.out.println("=====Imprimindo...=====");  
        for(Object autor: artistas){  
            System.out.println((Artista)autor);  
        }  
    }  
}
```

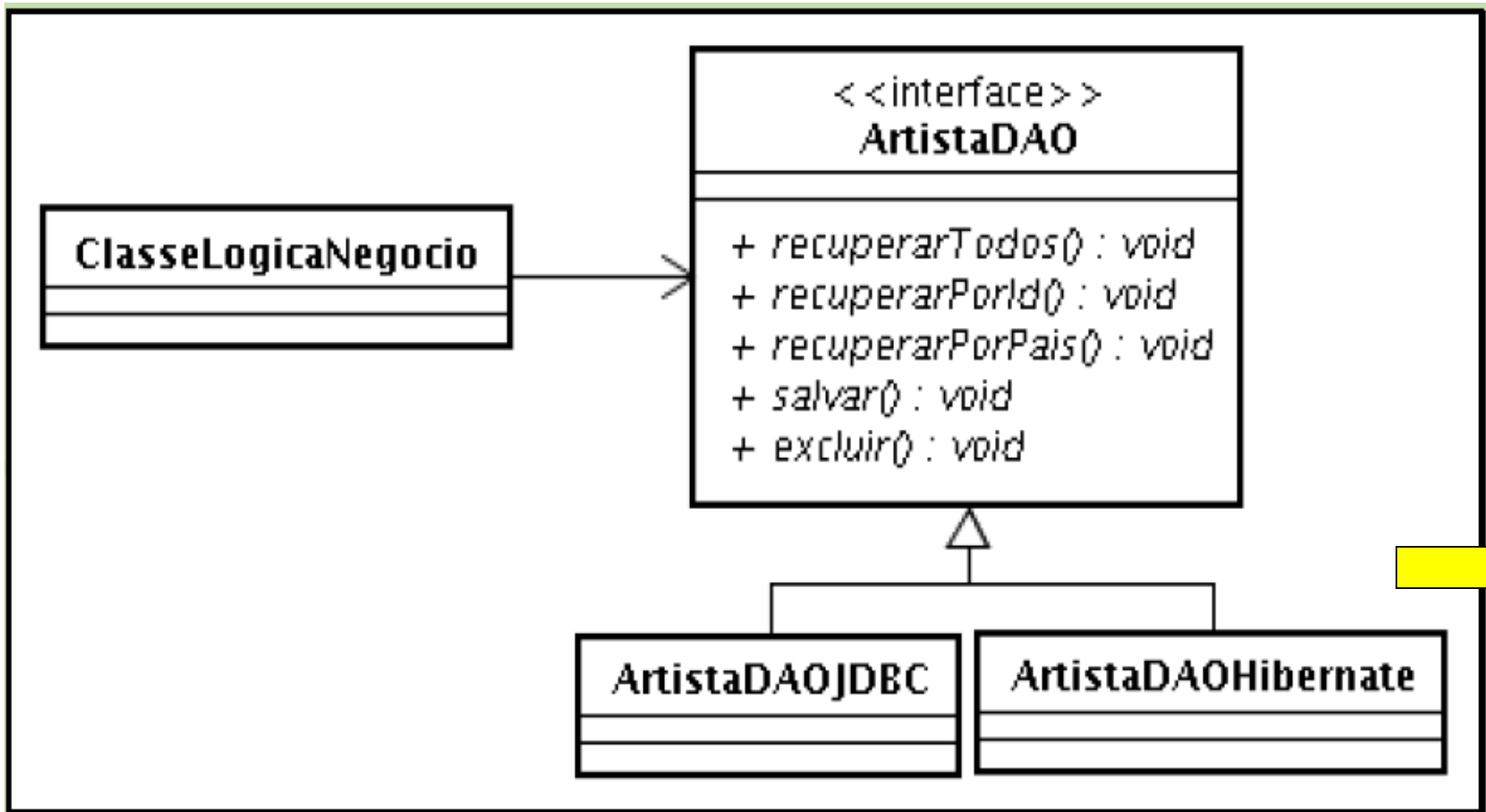
Exercício

- Blz,
 - Chega de conversa, vamos trabalhar...



O padrão de projeto DAO

- Dada a estruturação para a persistência da classe Artista da loja de CD JavaDiscos



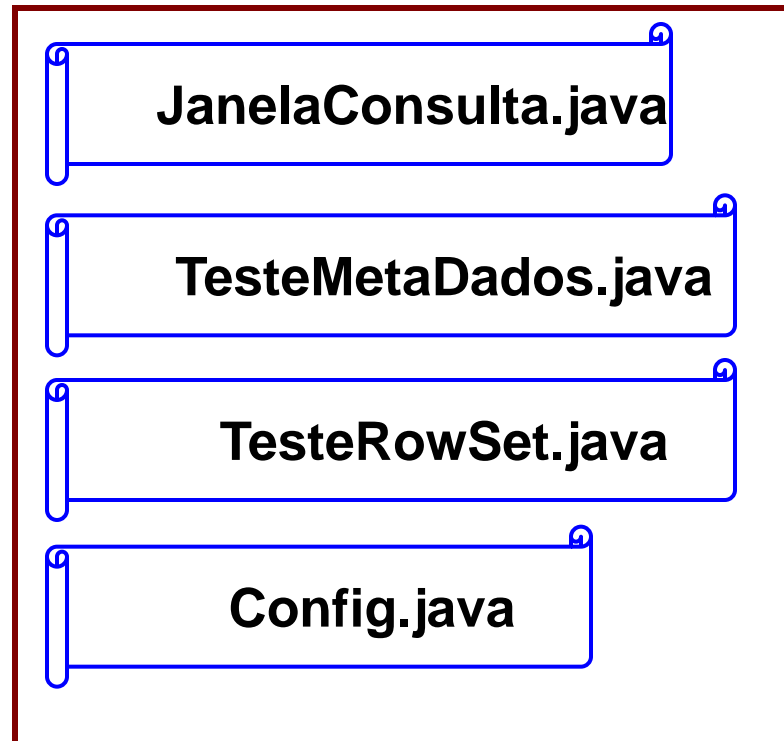
O padrão de projeto DAO

Exercício (IMPORTANTE FAZER ESTE!!!)

- 1. Implemente outros métodos para ArtistaDAO
 - Tipos recuperarPorID, recuperarPorPais, etc
- 2. Montar o padrão DAO para persistir a entidade de negócio Gravadora
 - Estrutura semelhante a Artista

Outros

- Execute as implementações apresentadas abaixo...



**Montar a
estrutura no
NetBeans...**