

ALPOO


Teoria - Java

Persistência

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

1



Disciplina: ALPOO


Tema: Persistência de Dados

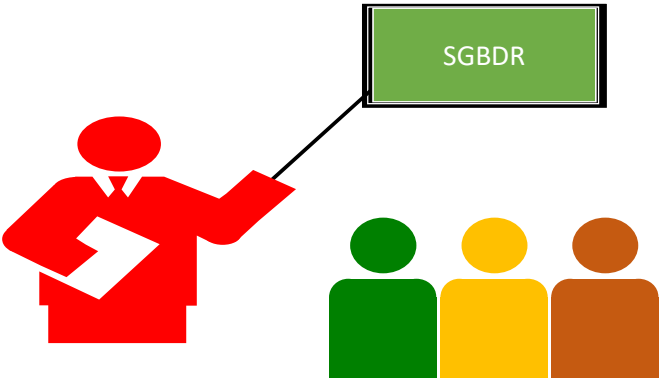
Professor: Ricardo Drudi

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

2






Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

3



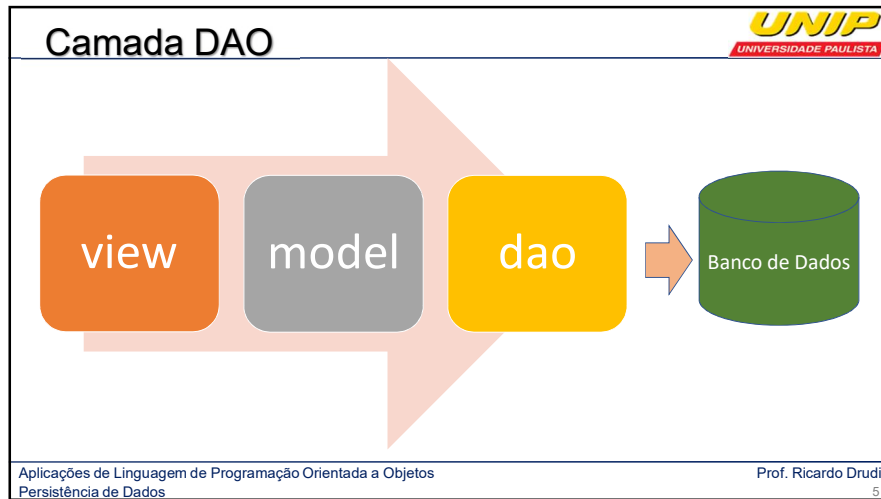
SGBD

- Um **Sistema Gerenciador de Banco de Dados** é um programa cujo objetivo é garantir a persistência dos dados entre as sessões de execução dos aplicativos.
- A persistência de dados é, do ponto de vista de uma aplicação Java, um serviço prestado pelo sistema computacional.
- O serviço de persistência é acessado por um número IP e uma porta.
- Para o Java, não interessa como a persistência é feita. Só interessa qual a *interface* pública (métodos disponíveis) que o servidor de persistência disponibiliza.

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

4



DAO

- DAO: Data Access Objects: Objetos de Acesso à Dados.
- O DAO é o objeto que conhece a implementação do Banco de Dados, e se comunica com este segundo a linguagem do banco.
- Para SGBDR, a linguagem de comunicação utilizada é o SQL.
- Outros padrões de bases de dados (NoSQL, OODB) utilizam linguagens de comunicação diferentes.
- O objeto do DAO é isolar o BD da aplicação. Toda requisição de dados da aplicação deve ser feita ao DAO, que se encarrega de obter os dados do banco.

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

6

DAO

- De forma geral, cada Model persistente tem o seu DAO.
- O DAO deve disponibilizar métodos para o CRUD do respectivo Model, além de outros métodos de acesso ao BD que se façam necessários.
- É útil criar uma interface para a implementação do DAO, pois dessa forma assegura-se que os métodos básicos serão implementados.
- O acesso ao BD é feito por conexões. Há um *design pattern* muito utilizado para essa tarefa, o *pattern* Factory.

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

7

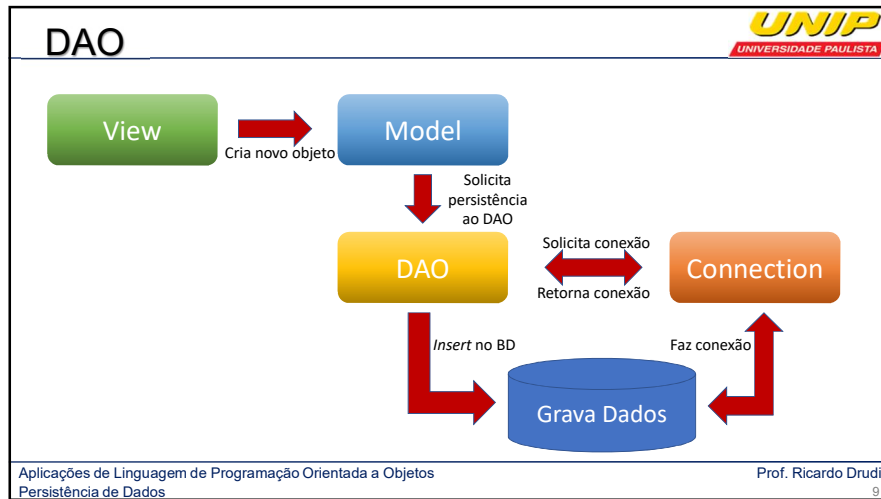
Interface Persistencia

```
public interface Persistencia<T> {  
    public boolean create (T obj); // grava  
    public List<T> read (); // seleciona todos  
    public boolean update (T obj); // atualiza  
    public boolean delete (T obj); // exclui  
    public T findByName (String nome); // retorna somente 1  
}
```

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

8



SGBDR

- Elementos de uma conexão com BD relacional:

```

Connection con; // conexão com o BD, fornecido pela Factory
PreparedStatement pst; // o statement SQL
ResultSet rs; // uma lista com o ResultSet de um Select
  
```

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
10

ConnectionFactory

```

public class ConnectionFactory {
    private static final String URL = "jdbc:mysql://localhost:3306/banco";
    private static final String USER = "usuario";
    private static final String PASSWORD = "senha";

    public static Connection getConnection() {
        try {
            return DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (SQLException e) {
            throw new RuntimeException("Erro na conexão");
        }
    }

    public static void closeConnection(Connection con) {
        try {
            if (con != null) con.close();
        } catch (SQLException e) {
            throw new RuntimeException("Erro no fechamento da conexão");
        }
    }
}
  
```

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
11

Exemplo Insert

```


public static void gravarTabela(Usuario obj) {
    Connection con = ConnectionFactory.getConnection();
    PreparedStatement pst = null;

    try {
        pst = con.prepareStatement("INSERT INTO Usuario (Login, Senha) VALUES (?, ?)");
        pst.setString(1, obj.getLogin());
        pst.setString(2, obj.getSenha());
        pst.executeUpdate();
    } catch (SQLException ex) {
        throw new RuntimeException("Erro no INSERT.");
    } finally {
        ConnectionFactory.closeConnection(con, pst);
    }
}
  
```

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
12

Exemplo Select




```
public static void lerTabela() {  
    Connection con = ConnectionFactory.getConnection();  
    PreparedStatement pst = null;  
    ResultSet rs = null;  
    try {  
        pst = con.prepareStatement("SELECT * FROM Usuario");  
        rs = pst.executeQuery();  
        while (rs.next()) {  
            String login = rs.getLogin("Login");  
            String senha = rs.getSenha("Senha");  
            new Usuario(login, senha);  
        }  
    } catch (SQLException ex) {  
        throw new RuntimeException("Erro no SELECT.");  
    } finally {  
        ConnectionFactory.closeConnection(con, pst, rs);  
    }  
}
```

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

13



Muito obrigado a todos.

Copyright © 2018 Prof. Ricardo Drudi

Todos direitos reservados. A reprodução ou divulgação total ou parcial deste documento é expressamente proibida sem o consentimento formal, por escrito, do professor Ricardo Drudi.

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

14