Wellington Cesar Fonseca

DESENVOLVIMENTO FULL STACK

MISSÃO PRÁTICA | NÍVEL 3 | MUNDO 3

RPG0016 - BACKEND SEM BANCO NÃO TEM

OBJETIVO

1. Implementar persistência com base no middleware JDBC.
2. Utilizar o padrão DAO (Data Access Object) no manuseio de dados.
3. Implementar o mapeamento objeto-relacional em sistemas Java.
4. Criar sistemas cadastrais com persistência em banco relacional.
5. No final do exercício, o aluno terá criado um aplicativo cadastral com uso do SQL Server na persistência de dados.

1° PROCEDIMENTO | MAPEAMENTO OBJETO-RELACIONAL E DAO

Qual a importância dos componentes de middleware, como o JDBC?
R: Os componentes de middleware JDBC, facilitou o processo de comunicação entre banco e aplicação. Neste nosso caso nos permitiu executar querys de forma eficiente, para se conectar só foi necessário informar a abertura e o fechamento da conexão. Ou seja, deixando pra gente que programa, se concentrar no projeto, sem precisar reinventar a roda da conexão

Qual a diferença no uso de Statement ou PreparedStatement para a manipulação de dados?
R: o Statement é na verdade a query crua (por assim dizer), onde os parâmetro são passado diretamente na string já o PreparedStatement é um "tradutor" para Statement, porém os parâmetros não são informados diretamente na string, eles passam por um processo de inserção, validação e tratamento e somente depois inseridos no statement, também é o mais seguro contra SQL Injection

Como o padrão DAO melhora a manutenibilidade do software?
R: O processo de DAO separa o processo de responsabilidade isolando o processos de manipulação de dados e regras de negócio. Com eles podemos também reutiliza-los em diferentes partes do projeto, evitando duplicidade de uso, centralizando a manutenção

Como a herança é refletida no banco de dados, quando lidamos com um modelo estritamente relacional?
Podem ser refletidas como **Uma tabela para Cada Classe:** cada classe tem a sua própria tabela; **Uma tabela por Subclasse:** Cada classe da hieraquia tem a sua própria tabela; **Uma tabela para Toda Hierarquia:** Todas as classes e hierarquias são armazenadas e uma única tabela (que foi a mesma que eu usei).

Observação! Não consegui instalar no mac o sql server, então estou utilizando o **Postgree** como banco de dados

CadastroBDTeste.java

```
package cadastrobd.model;

/**
 *
 * @author wellingtonfonseca
 */
public class CadastroBDTeste {
    public static void main(String[] args) {
        PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO();

        System.out.println("-------------------------");
        System.out.println("Pessoa Fisica incluir");
        System.out.println("-------------------------");
        PessoaFisica pf = new PessoaFisica(0, "João Silva", "Rua A", "Cidade A", "Estado A",
"123456789", "joao@exemplo.com", "111.222.333-44");
        pf = pessoaFisicaDAO.incluir(pf);
        System.out.println("OK");
        System.out.println("-------------------------");

        System.out.println("-------------------------");
        System.out.println("Pessoa Fisica alterar");
        System.out.println("-------------------------");
        pf.setNome("João Silva Alterado");
        pessoaFisicaDAO.alterar(pf);
        System.out.println("OK");
        System.out.println("-------------------------");

        System.out.println("-------------------------");
        System.out.println("Pessoa Fisica listar");
        System.out.println("-------------------------");
        // Consultar todas as pessoas Fisicas do banco de dados e listar no console
        for (PessoaFisica pessoa : pessoaFisicaDAO.getPessoas()) {
            System.out.println(" ");
            pessoa.exibir();
            System.out.println(" ");
        }
        System.out.println("-------------------------");

        System.out.println("-------------------------");
        System.out.println("Pessoa Fisica exluir");
        System.out.println("-------------------------");
        pessoaFisicaDAO.excluir(pf.getId());
        System.out.println("OK");
        System.out.println("-------------------------");

        PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO();

        System.out.println("-------------------------");
        System.out.println("Pessoa Juridica incluir");
        System.out.println("-------------------------");
        PessoaJuridica pj = new PessoaJuridica(0, "Empresa X", "Rua B", "Cidade B",
"Estado B", "987654321", "empresa@exemplo.com", "12.345.678/0001-99");
        pessoaJuridicaDAO.incluir(pj);
        System.out.println("OK");
        System.out.println("-------------------------");

        System.out.println("-------------------------");
        System.out.println("Pessoa Juridica alterar");
        System.out.println("-------------------------");
        pj.setNome("Empresa X Alterada");
        pessoaJuridicaDAO.alterar(pj);
        System.out.println("OK");
        System.out.println("-------------------------");

        System.out.println("-------------------------");
        System.out.println("Pessoa Juridica listar");
        System.out.println("-------------------------");
        for (PessoaJuridica pessoa : pessoaJuridicaDAO.getPessoas()) {
            System.out.println(" ");
            pessoa.exibir();
            System.out.println(" ");
        }
        System.out.println("-------------------------");

        System.out.println("-------------------------");
        System.out.println("Pessoa Juridica exluir");
        System.out.println("-------------------------");
        pessoaJuridicaDAO.excluir(pj.getId());
        System.out.println("OK");
        System.out.println("-------------------------");
    }
}
```

Pessoa.java

```
package cadastrobd.model;

/**
 *
 * @author wellingtonfonseca
 */
public class Pessoa {
    private int id;
    private String nome;
    private String logradouro;
    private String cidade;
    private String estado;
    private String telefone;
    private String email;

    public void setId(int id) {
        this.id = id;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public void setLogradouro(String logradouro) {
        this.logradouro = logradouro;
    }

    public void setCidade(String cidade) {
        this.cidade = cidade;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public void setTelefone(String telefone) {
        this.telefone = telefone;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public int getId() {
        return id;
    }

    public String getNome() {
        return nome;
    }

    public String getLogradouro() {
        return logradouro;
    }

    public String getCidade() {
        return cidade;
    }

    public String getEstado() {
        return estado;
    }

    public String getTelefone() {
        return telefone;
    }

    public String getEmail() {
        return email;
    }

    public Pessoa(
            int id,
            String nome,
            String logradouro,
            String cidade,
            String estado,
            String telefone,
            String email
    ) {
        this.id = id;
        this.nome = nome;
        this.logradouro = logradouro;
        this.cidade = cidade;
        this.estado = estado;
        this.telefone = telefone;
        this.email = email;
    }

    public void exibir() {
        System.out.println("id: " + id);
        System.out.println("nome: " + nome);
        System.out.println("logradouro: " + logradouro);
        System.out.println("cidade: " + cidade);
        System.out.println("estado: " + estado);
        System.out.println("telefone: " + telefone);
        System.out.println("email: " + email);
    }

}
```

PessoaFisica.java

```
package cadastrobd.model;


/**
 *
 * @author wellingtonfonseca
 */
public class PessoaFisica extends Pessoa {
    private String cpf;

    public void setCpf(String cpf) {
        this.cpf = cpf;
    }

    public String getCpf() {
        return cpf;
    }

    public PessoaFisica(
            int id,
            String nome,
            String logradouro,
            String cidade,
            String estado,
            String telefone,
            String email,
            String cpf
    ) {
        super(
            id,
            nome,
            logradouro,
            cidade,
            estado,
            telefone,
            email
        );
        this.cpf = cpf;
    }

    @Override
    public void exibir() {
        super.exibir();
        System.out.println("cpf: " + cpf);
    }
}
```

PessoaFisicaDAO.java

```
package cadastrobd.model;

import cadastrobd.model.util.ConectorBD;
import cadastrobd.model.util.SequenceManager;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author wellingtonfonseca
 */
public class PessoaFisicaDAO {
    public PessoaFisica getPessoa(int id) {
        PessoaFisica pf = null;

        String sql = "select * from pessoa p where p.tipo = 'pf' p.id = ?";

        try (Connection conn = ConectorBD.getConnection();
             PreparedStatement stmt = ConectorBD.getPrepared(conn, sql)) {
            stmt.setInt(1, id);
            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {
                pf = new PessoaFisica(
                        rs.getInt("id"),
                        rs.getString("nome"),
                        rs.getString("logradouro"),
                        rs.getString("cidade"),
                        rs.getString("estado"),
                        rs.getString("telefone"),
                        rs.getString("email"),
                        rs.getString("cpf_cnpj")
                );
            }
            ConectorBD.close(rs);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return pf;
    }

    public List<PessoaFisica> getPessoas() {
        List<PessoaFisica> pessoas = new ArrayList<>();

        String sql = "select * from pessoa p where p.tipo = 'pf'";

        try (Connection conn = ConectorBD.getConnection();
             ResultSet rs = ConectorBD.getSelect(conn, sql)) {
            while (rs.next()) {
                PessoaFisica pf = new PessoaFisica(
                        rs.getInt("id"),
                        rs.getString("nome"),
                        rs.getString("logradouro"),
                        rs.getString("cidade"),
                        rs.getString("estado"),
                        rs.getString("telefone"),
                        rs.getString("email"),
                        rs.getString("cpf_cnpj")
                );
                pessoas.add(pf);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return pessoas;
    }
```

PessoaFisicaDAO.java

```
    public PessoaFisica incluir(PessoaFisica pf) {
        String sqlPessoa = "insert into pessoa (id, tipo, nome, logradouro, email, telefone,
cpf_cnpj, cidade, estado) VALUES (?, 'pf', ?, ?, ?, ?, ?, ?, ?)";

        try (Connection conn = ConectorBD.getConnection()) {
            conn.setAutoCommit(false);

            int id = SequenceManager.getValue("pessoa_id_seq");

            try (PreparedStatement stmtPessoa = ConectorBD.getPrepared(conn,
sqlPessoa)) {

                stmtPessoa.setInt(1, id);
                stmtPessoa.setString(2, pf.getNome());
                stmtPessoa.setString(3, pf.getLogradouro());
                stmtPessoa.setString(4, pf.getEmail());
                stmtPessoa.setString(5, pf.getTelefone());
                stmtPessoa.setString(6, pf.getCpf());
                stmtPessoa.setString(7, pf.getCidade());
                stmtPessoa.setString(8, pf.getEstado());

                stmtPessoa.executeUpdate();

                conn.commit();

                pf.setId(id);

            } catch (SQLException e) {
                conn.rollback();
                e.printStackTrace();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return pf;
    }

    public void alterar(PessoaFisica pf) {
        String sqlPessoa = "update pessoa set nome = ?, logradouro = ?, email = ?,
telefone = ?, cpf_cnpj = ?, cidade = ?, estado = ? WHERE tipo = 'pf' and id = ?";

        try (Connection conn = ConectorBD.getConnection()) {
            conn.setAutoCommit(false);

            try (PreparedStatement stmtPessoa = ConectorBD.getPrepared(conn,
sqlPessoa)) {
                stmtPessoa.setString(1, pf.getNome());
                stmtPessoa.setString(2, pf.getLogradouro());
                stmtPessoa.setString(3, pf.getEmail());
                stmtPessoa.setString(4, pf.getTelefone());
                stmtPessoa.setString(5, pf.getCpf());
                stmtPessoa.setString(6, pf.getCidade());
                stmtPessoa.setString(7, pf.getEstado());
                stmtPessoa.setInt(8, pf.getId());

                stmtPessoa.executeUpdate();

                conn.commit();
            } catch (SQLException e) {
                conn.rollback();
                e.printStackTrace();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void excluir(int id) {
        String sqlPessoa = "delete from pessoa where tipo = 'pf' and id = ?";

        try (Connection conn = ConectorBD.getConnection()) {
            conn.setAutoCommit(false);

            try (PreparedStatement stmtPessoa = ConectorBD.getPrepared(conn,
sqlPessoa)) {
                stmtPessoa.setInt(1, id);
                stmtPessoa.executeUpdate();

                conn.commit();
            } catch (SQLException e) {
                conn.rollback();
                e.printStackTrace();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

PessoaJuridica.java

```
package cadastrobd.model;


/**
 *
 * @author wellingtonfonseca
 */
public class PessoaJuridica extends Pessoa {
    private String cnpj;

    public void setCnpj(String cnpj) {
        this.cnpj = cnpj;
    }

    public String getCnpj() {
        return cnpj;
    }

    public PessoaJuridica(
            int id,
            String nome,
            String logradouro,
            String cidade,
            String estado,
            String telefone,
            String email,
            String cnpj
    ) {
        super(
            id,
            nome,
            logradouro,
            cidade,
            estado,
            telefone,
            email
        );
        this.cnpj = cnpj;
    }

    @Override
    public void exibir() {
        super.exibir();
        System.out.println("cnpj: " + cnpj);
    }
}
```

PessoaJuridicaDAO.java

```
package cadastrobd.model;

import cadastrobd.model.util.ConectorBD;
import cadastrobd.model.util.SequenceManager;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author wellingtonfonseca
 */
public class PessoaJuridicaDAO {
    public PessoaJuridica getPessoa(int id) {
        PessoaJuridica pj = null;

        String sql = "select * from pessoa p where p.tipo = 'pj' p.id = ?";

        try (Connection conn = ConectorBD.getConnection();
             PreparedStatement stmt = ConectorBD.getPrepared(conn, sql)) {
            stmt.setInt(1, id);
            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {
                pj = new PessoaJuridica(
                        rs.getInt("id"),
                        rs.getString("nome"),
                        rs.getString("logradouro"),
                        rs.getString("cidade"),
                        rs.getString("estado"),
                        rs.getString("telefone"),
                        rs.getString("email"),
                        rs.getString("cpf_cnpj")
                );
            }
            ConectorBD.close(rs);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return pj;
    }

    public List<PessoaJuridica> getPessoas() {
        List<PessoaJuridica> pessoas = new ArrayList<>();

        String sql = "select * from pessoa p where p.tipo = 'pj'";

        try (Connection conn = ConectorBD.getConnection();
             ResultSet rs = ConectorBD.getSelect(conn, sql)) {
            while (rs.next()) {
                PessoaJuridica pj = new PessoaJuridica(
                        rs.getInt("id"),
                        rs.getString("nome"),
                        rs.getString("logradouro"),
                        rs.getString("cidade"),
                        rs.getString("estado"),
                        rs.getString("telefone"),
                        rs.getString("email"),
                        rs.getString("cpf_cnpj")
                );
                pessoas.add(pj);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return pessoas;
    }
```

PessoaJuridicaDAO.java

```
    public PessoaJuridica incluir(PessoaJuridica pj) {
        String sqlPessoa = "insert into pessoa (id, tipo, nome, logradouro, email, telefone,
cpf_cnpj, cidade, estado) VALUES (?, 'pj', ?, ?, ?, ?, ?, ?, ?)";

        try (Connection conn = ConectorBD.getConnection()) {
            conn.setAutoCommit(false);

            int id = SequenceManager.getValue("pessoa_id_seq");

            try (PreparedStatement stmtPessoa = ConectorBD.getPrepared(conn,
sqlPessoa)) {

                stmtPessoa.setInt(1, id);
                stmtPessoa.setString(2, pj.getNome());
                stmtPessoa.setString(3, pj.getLogradouro());
                stmtPessoa.setString(4, pj.getEmail());
                stmtPessoa.setString(5, pj.getTelefone());
                stmtPessoa.setString(6, pj.getCnpj());
                stmtPessoa.setString(7, pj.getCidade());
                stmtPessoa.setString(8, pj.getEstado());

                stmtPessoa.executeUpdate();

                conn.commit();

                pj.setId(id);

            } catch (SQLException e) {
                conn.rollback();
                e.printStackTrace();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return pj;
    }

    public void alterar(PessoaJuridica pj) {
        String sqlPessoa = "update pessoa set nome = ?, logradouro = ?, email = ?,
telefone = ?, cpf_cnpj = ?, cidade = ?, estado = ? WHERE tipo = 'pj' and id = ?";

        try (Connection conn = ConectorBD.getConnection()) {
            conn.setAutoCommit(false);

            try (PreparedStatement stmtPessoa = ConectorBD.getPrepared(conn,
sqlPessoa)) {
                stmtPessoa.setString(1, pj.getNome());
                stmtPessoa.setString(2, pj.getLogradouro());
                stmtPessoa.setString(3, pj.getEmail());
                stmtPessoa.setString(4, pj.getTelefone());
                stmtPessoa.setString(5, pj.getCnpj());
                stmtPessoa.setString(6, pj.getCidade());
                stmtPessoa.setString(7, pj.getEstado());
                stmtPessoa.setInt(8, pj.getId());

                stmtPessoa.executeUpdate();

                conn.commit();
            } catch (SQLException e) {
                conn.rollback();
                e.printStackTrace();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void excluir(int id) {
        String sqlPessoa = "delete from pessoa where tipo = 'pj' and id = ?";

        try (Connection conn = ConectorBD.getConnection()) {
            conn.setAutoCommit(false);

            try (PreparedStatement stmtPessoa = ConectorBD.getPrepared(conn,
sqlPessoa)) {
                stmtPessoa.setInt(1, id);
                stmtPessoa.executeUpdate();

                conn.commit();
            } catch (SQLException e) {
                conn.rollback();
                e.printStackTrace();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

ConectorBD.java

```
package cadastrobd.model.util;
import java.sql.*;

/**
 *
 * @author wellingtonfonseca
 */
public class ConectorBD {
    private static final String URL = "jdbc:postgresql://xxx";
    private static final String USER = "xxx";
    private static final String PASSWORD = "xxx";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }

    public static PreparedStatement getPrepared(
        Connection conn,
        String sql
    ) throws SQLException {
        return conn.prepareStatement(sql);
    }

    public static ResultSet getSelect(
        Connection conn,
        String sql
    ) throws SQLException {
        Statement stmt = conn.createStatement();
        return stmt.executeQuery(sql);
    }

    public static void close(
        Connection conn
    ) {
        try {
            if (conn != null && !conn.isClosed()) {
                conn.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public static void close(
        Statement stmt
    ) {
        try {
            if (stmt != null && !stmt.isClosed()) {
                stmt.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public static void close(
        ResultSet rs
    ) {
        try {
            if (rs != null && !rs.isClosed()) {
                rs.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

SequenceManager.java

```
package cadastrobd.model.util;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

/**
 *
 * @author wellingtonfonseca
 */
public class SequenceManager {
    public static int getValue(
        String sequenceName
    ) {
        int value = -1;
        String sql = "select nextval(?)";
        try (Connection conn = ConectorBD.getConnection();
             PreparedStatement stmt = ConectorBD.getPrepared(conn, sql)) {
            stmt.setString(1, sequenceName);
            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {
                value = rs.getInt(1);
            }
            ConectorBD.close(rs);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return value;
    }
}
```