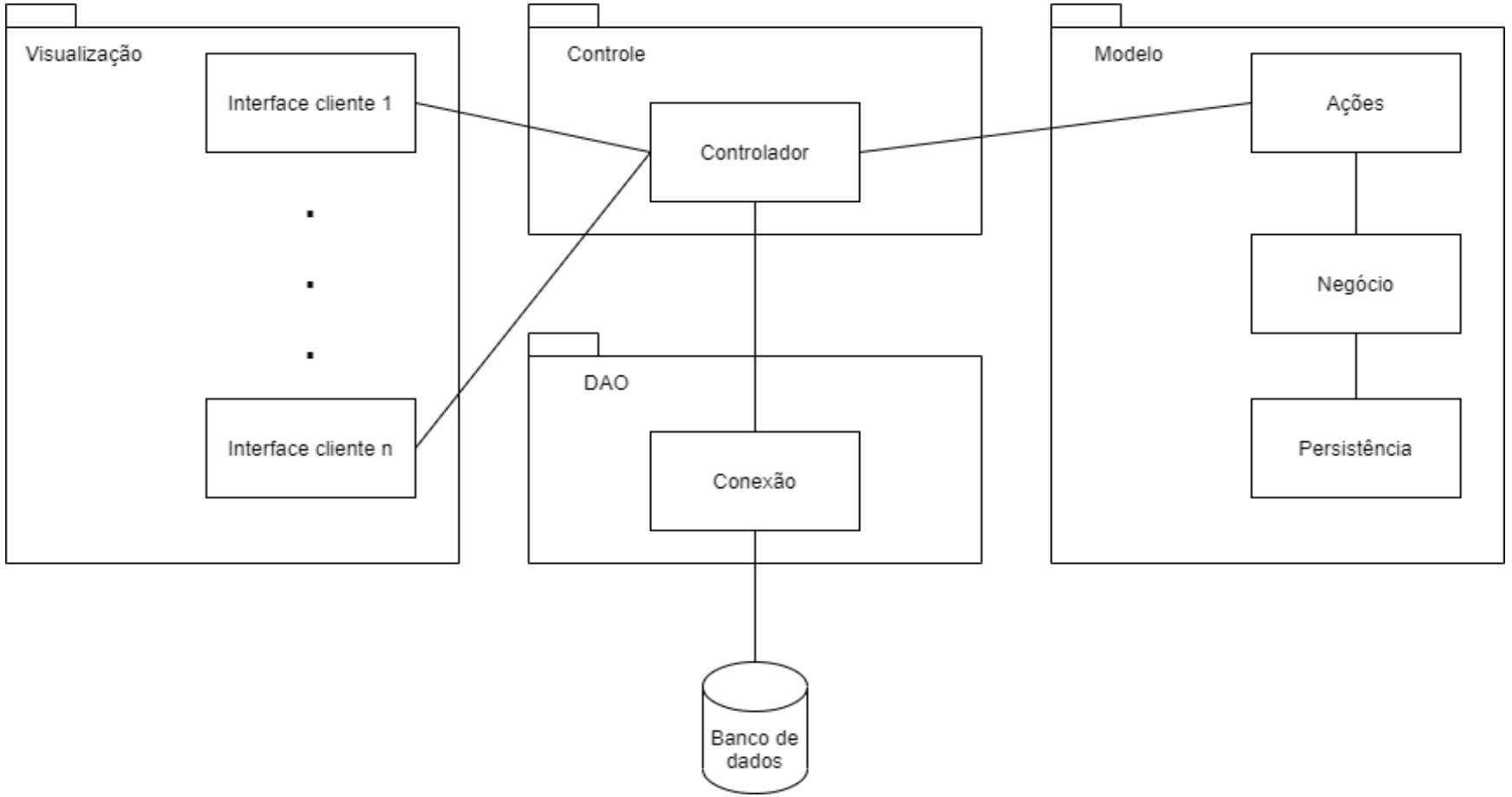
UNIP
UNIVERSIDADE PAULISTA

UNICLUSTER

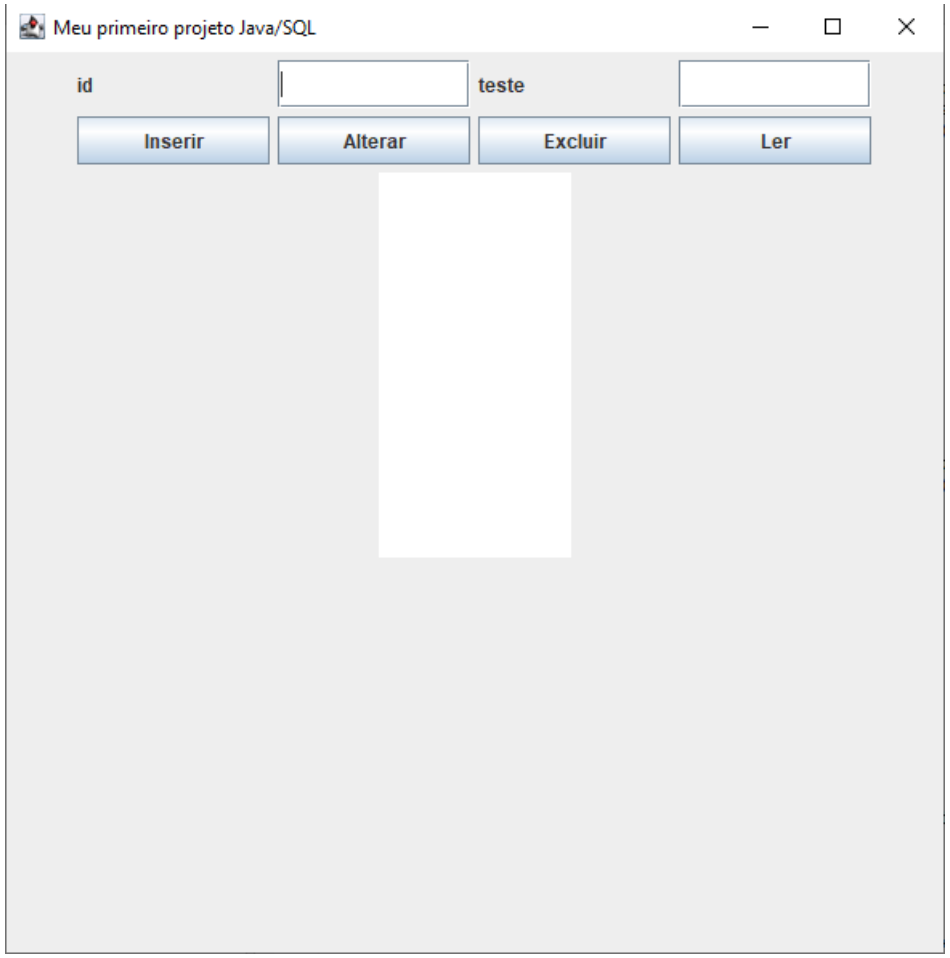
Prj MVC-DAO

Carlos Arruda Baltazar
UNIP – Cidade Universitária

Prj MVC-DAO


Visualização
Interface cliente 1
Interface cliente n
Controle
Controlador
DAO
Conexão
Banco de dados
Modelo
Ações
Negócio
Persistência

Prj MVC-DAO


Meu primeiro projeto Java/SQL
id
teste
Inserir
Alterar
Excluir
Ler

Model


```java
package Pck_Model;

public class TesteModel
{
    private int pkId;
    private String dsTeste;

    public TesteModel()
    {

    }

    public int getPkId() {
        return pkId;
    }

    public void setPkId(int pkId) {
        this.pkId = pkId;
    }

    public String getDsTeste() {
        return dsTeste;
    }

    public void setDsTeste(String dsTeste) {
        this.dsTeste = dsTeste;
    }
}
```

DAO


```
package Pck_DAO;

import java.sql.Connection;

public class MySqlConnection
{
    private static String status = "Não conectou...";
    private static String serverName = "127.0.0.1";
    private static String mydatabase ="lpbd";
    private static String url = "jdbc:mysql://" + serverName + "/" + mydatabase;
    private static String username = "aluno";
    private static String password = "4@11";

    public MySqlConnection() {}

    public java.sql.Connection getMySqlConnection()
    {
            Connection connection = null;
            try
            {
                String driverName = "com.mysql.jdbc.Driver";
                Class.forName(driverName);
                connection = DriverManager.getConnection(url, username, password);
                if (connection != null)
                {
                    status = ("STATUS--->Conectado com sucesso!");
                    System.out.println(status);
                }
                else
                {
                    status = ("STATUS--->Não foi possivel realizar conexão");
                    System.out.println(status);
                }
                return connection;
            }
            catch (ClassNotFoundException e)
            {
                System.out.println("O driver expecificado nao foi encontrado.");
                return null;
            }
            catch (SQLException e)
            {
                System.out.println("Nao foi possivel conectar ao Banco de Dados.");
                return null;
            }
        }

    public String statusConection()
    {
        return status;
    }
    public boolean CloseConnection()
    {
        try
        {
            this.getMySqlConnection().close();
            return true;
        }
        catch (SQLException e)
        {
            return false;
        }
    }
    public java.sql.Connection RestartConnection()
    {
        this.CloseConnection();
        return this.getMySqlConnection();
    }
}
```

View


```java
package Pck_View;

import java.awt.Dimension;

public class MyWindow extends JFrame
{
    private JLabel jl_teste, jl_id;
    private JTextField jtf_teste, jtf_id;
    private JTextArea jta_read;
    private JButton jb_insert, jb_update, jb_remove, jb_read;

    private TesteController testeController;

    public MyWindow()
    {
        this.testeController = new TesteController();

        this.CreateGUI();
        this.CreateEvents();
    }
```

```java
public void CreateGUI()
{
    this.setSize(new Dimension(600, 600));
    this.setTitle("Meu primeiro projeto Java/SQL");
    this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    this.setLayout(new FlowLayout());

    this.jl_id = new JLabel("id");
    this.jl_id.setPreferredSize(new Dimension(120,30));
    this.getContentPane().add(jl_id);

    this.jtf_id = new JTextField();
    this.jtf_id.setPreferredSize(new Dimension(120,30));
    this.getContentPane().add(jtf_id);

    this.jl_teste = new JLabel("teste");
    this.jl_teste.setPreferredSize(new Dimension(120,30));
    this.getContentPane().add(jl_teste);

    this.jtf_teste = new JTextField();
    this.jtf_teste.setPreferredSize(new Dimension(120,30));
    this.getContentPane().add(jtf_teste);

    this.jb_insert = new JButton("Inserir");
    this.jb_insert.setPreferredSize(new Dimension(120,30));
    this.getContentPane().add(jb_insert);

    this.jb_update = new JButton("Alterar");
    this.jb_update.setPreferredSize(new Dimension(120,30));
    this.getContentPane().add(jb_update);

    this.jb_remove = new JButton("Excluir");
    this.jb_remove.setPreferredSize(new Dimension(120,30));
    this.getContentPane().add(jb_remove);

    this.jb_read = new JButton("Ler");
    this.jb_read.setPreferredSize(new Dimension(120,30));
    this.getContentPane().add(jb_read);

    this.jta_read = new JTextArea();
    this.jta_read.setPreferredSize(new Dimension(120,240));
    this.getContentPane().add(jta_read);
}
```

View


```
    public void CreateEvents()
    {
        this.jb_insert.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                testeController.InsertTeste(jtf_teste.getText());
            }
        });

        this.jb_update.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                testeController.UpdateTeste(Integer.parseInt(jtf_id.getText()),
                                             jtf_teste.getText());
            }
        });

        this.jb_remove.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                testeController.DeleteTeste(Integer.parseInt(jtf_id.getText()));
            }
        });

        this.jb_read.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                jta_read.setText(testeController.ReadTeste());
            }
        });
    }
}
```

```
package Pck_View;

public class Main
{
    public static void main(String [] args)
    {
        MyWindow myWindow = new MyWindow();
        myWindow.setVisible(true);
    }
}
```

Controller


```java
package Pck_Controller;

import java.sql.PreparedStatement;

public class TesteController
{
    private MySqlConnection mySqlConnection;
    private PreparedStatement preparedStatement;
    private TesteModel testeModel;
    private ResultSet resultSet;
    private String insertQuery = "INSERT INTO teste (ds_teste) VALUES(?);";
    private String updateQuery = "UPDATE teste SET ds_teste = ? WHERE pk_id = ?;";
    private String deleteQuery = "DELETE FROM teste WHERE pk_id = ?";
    private String readQuery = "SELECT * FROM teste;";

    public TesteController ()
    {
        this.mySqlConnection = new MySqlConnection();
        this.testeModel = new TesteModel();
    }
```

```java
public void InsertTeste(String dsTeste)
{
    this.testeModel.setDsTeste(dsTeste);

    try
    {
        this.preparedStatement = this.mySqlConnection.getMySqlConnection().prepareStatement(insertQuery);
        this.preparedStatement.setString (1, testeModel.getDsTeste());
        this.preparedStatement.execute();

        this.mySqlConnection.CloseConnection();
    }
    catch (SQLException e)
    {
        System.out.println("Query error: " + e.toString());
    }
}

public void UpdateTeste(int pkId, String dsTeste)
{
    this.testeModel.setPkId(pkId);
    this.testeModel.setDsTeste(dsTeste);
    try
    {
        this.preparedStatement = this.mySqlConnection.getMySqlConnection().prepareStatement(updateQuery);
        this.preparedStatement.setString (1, testeModel.getDsTeste());
        this.preparedStatement.setInt(2, testeModel.getPkId());
        this.preparedStatement.execute();

        this.mySqlConnection.CloseConnection();
    }
    catch (SQLException e)
    {
        System.out.println("Query error: " + e.toString());
    }
}
```

Controller


```
    public void DeleteTeste(int pkId)
    {
        this.testeModel.setPkId(pkId);
        try
        {
            this.preparedStatement = this.mySqlConnection.getMySqlConnection().prepareStatement(deleteQuery);
            this.preparedStatement.setInt(1, testeModel.getPkId());
            this.preparedStatement.execute();

            this.mySqlConnection.CloseConnection();
        }
        catch (SQLException e)
        {
            System.out.println("Query error: " + e.toString());
        }
    }

    public String ReadTeste()
    {
        String result = "";
        try
        {
            this.resultSet = this.mySqlConnection.getMySqlConnection().createStatement().executeQuery(readQuery);

            result += "id\tteste\n";

            while (this.resultSet.next())
            {
                result += this.resultSet.getInt("pk_id") + "\t" + this.resultSet.getString("ds_teste") + "\n";
            }

            this.mySqlConnection.CloseConnection();

            return result;
        }
        catch (SQLException e)
        {
            System.out.println("Query error: " + e.toString());

            return result;
        }
    }
}
```

OBRIGADO