

SITUAÇÃO DE APRENDIZAGEM III

Utilizaremos um banco de dados chamado "mydatabase" com uma tabela chamada "contacts" contendo os campos "id", "name" e "email".

Estrutura do Projeto:

- ProjetoMVC
- src
- main
- java
- controllers
- ContactController.java
- models
- Contact.java
- utils
- DBUtil.java
- webapp
- WEB-INF
- views
- add.jsp
- edit.jsp
- list.jsp
- index.jsp
- web.xml

Contact.java (Model):

```
package models;
```

```
public class Contact {  
    private int id;  
    private String name;  
    private String email;
```

```
    // Getters and setters
```

```
    // Constructors
```

```
}
```

DBUtil.java (Utilitário de Banco de Dados):

```
package utils;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBUtil {
    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/mydatabase";
    private static final String JDBC_USER = "your_username";
    private static final String JDBC_PASSWORD = "your_password";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(JDBC_URL, JDBC_USER, JDBC_PASSWORD);
    }
}
```

ContactController.java (Controller):

```
package controllers;

import models.Contact;
import utils.DBUtil;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

@WebServlet("/contacts/*")
public class ContactController extends HttpServlet {
    @Override
```

```

protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String pathInfo = request.getPathInfo();

    if (pathInfo == null || pathInfo.equals("/")) {
        listContacts(request, response);
    } else if (pathInfo.equals("/add")) {
        // Exibir formulário para adicionar contato
        request.getRequestDispatcher("/WEB-INF/views/add.jsp").forward(request,
response);
    } else if (pathInfo.matches("/\\d+")) {
        int contactId = Integer.parseInt(pathInfo.substring(1));
        showContact(request, response, contactId);
    }
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String pathInfo = request.getPathInfo();

    if (pathInfo == null || pathInfo.equals("/")) {
        response.sendRedirect(request.getContextPath() + "/contacts");
    } else if (pathInfo.equals("/add")) {
        addContact(request, response);
    } else if (pathInfo.matches("/\\d+/edit")) {
        int contactId = Integer.parseInt(pathInfo.substring(1, pathInfo.indexOf("/edit")));
        editContact(request, response, contactId);
    } else if (pathInfo.matches("/\\d+/delete")) {
        int contactId = Integer.parseInt(pathInfo.substring(1, pathInfo.indexOf("/delete")));
        deleteContact(request, response, contactId);
    }
}

private void listContacts(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    try (Connection connection = DBUtil.getConnection()) {
        List<Contact> contacts = new ArrayList<>();
        String sql = "SELECT * FROM contacts";
        try (PreparedStatement statement = connection.prepareStatement(sql);
            ResultSet resultSet = statement.executeQuery()) {
            while (resultSet.next()) {
                Contact contact = new Contact();
                contact.setId(resultSet.getInt("id"));
                contact.setName(resultSet.getString("name"));
                contact.setEmail(resultSet.getString("email"));
                contacts.add(contact);
            }
        }
    }
}

```

```

    }
    request.setAttribute("contacts", contacts);
    request.getRequestDispatcher("/WEB-INF/views/list.jsp").forward(request,
response);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```

private void showContact(HttpServletRequest request, HttpServletResponse response, int
contactId)
    throws ServletException, IOException {
    try (Connection connection = DBUtil.getConnection()) {
        String sql = "SELECT * FROM contacts WHERE id = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, contactId);
            try (ResultSet resultSet = statement.executeQuery()) {
                if (resultSet.next()) {
                    Contact contact = new Contact();
                    contact.setId(resultSet.getInt("id"));
                    contact.setName(resultSet.getString("name"));
                    contact.setEmail(resultSet.getString("email"));
                    request.setAttribute("contact", contact);
                    request.getRequestDispatcher("/WEB-INF/views/show.jsp").forward(request,
response);
                } else {
                    response.sendRedirect(request.getContextPath() + "/contacts");
                }
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```

private void addContact(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String name = request.getParameter("name");
    String email = request.getParameter("email");

    try (Connection connection = DBUtil.getConnection()) {
        String sql = "INSERT INTO contacts (name, email) VALUES (?, ?)";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(1, name);
            statement.setString(2, email);
            statement.executeUpdate();
        }
        response.sendRedirect(request.getContextPath() + "/contacts");
    }
}

```

```

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

private void editContact(HttpServletRequest request, HttpServletResponse response, int
contactId)
    throws ServletException, IOException {
    try (Connection connection = DBUtil.getConnection()) {
        String sql = "SELECT * FROM contacts WHERE id = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, contactId);
            try (ResultSet resultSet = statement.executeQuery()) {
                if (resultSet.next()) {
                    Contact contact = new Contact();
                    contact.setId(resultSet.getInt("id"));
                    contact.setName(resultSet.getString("name"));
                    contact.setEmail(resultSet.getString("email"));
                    request.setAttribute("contact", contact);
                    request.getRequestDispatcher("/WEB-INF/views/edit.jsp").forward(request,
response);
                } else {
                    response.sendRedirect(request.getContextPath() + "/contacts");
                }
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

private void deleteContact(HttpServletRequest request, HttpServletResponse response,
int contactId)
    throws ServletException, IOException {
    try (Connection connection = DBUtil.getConnection()) {
        String sql = "DELETE FROM contacts WHERE id = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, contactId);
            statement.executeUpdate();
        }
        response.sendRedirect(request.getContextPath() + "/contacts");
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}

```

index.jsp (View - Página Inicial):

```
<!DOCTYPE html>
<html>
<head>
  <title>Contacts</title>
</head>
<body>
  <h1>Contacts</h1>
  <a href="${request.contextPath}/contacts/add">Add Contact</a>
  <ul>
    <c:forEach items="${contacts}" var="
```