

# EXEMPLOS MVC

## index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Sistema Acadêmico</title>
</head>
<body>
  <h1>Sistema Acadêmico</h1>
  <a href="html\incluir.html">Incluir Alunos</a><br/>
  <a href="html\atualizar.html">Alterar Alunos</a><br/>
  <a href="html\excluir.html">Excluir Alunos</a><br/>
  <a href="html\consultar.html">Consultar Alunos</a><br/>
  <a href="ServletAluno?cmd=Listar">Listar todos</a><br/>
</body>
</html>
```

# incluir.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Sistema Acadêmico</title>
</head>
<body>
    <form action="../../ServletAluno" method="get">
        <table border="1">
            <input type="hidden" name="cmd" value="incluir">
            <tr><th colspan="2"><h1>Cadastro de Alunos</h1></th></tr>
            <tr>
                <td>RA:</td>
                <td><input type="text" name="txtRa" size="60" /></td>
            </tr>
            <tr>
                <td>Nome:</td>
                <td><input type="text" name="txtNome" size="60" /></td>
            </tr>
            <tr>
                <td>E-mail:</td>
                <td><input type="email" size="60" name="txtEmail" /></td>
            </tr>
            <tr>
                <td>Endereço:</td>
                <td><input type="text" size="60" name="txtEndereco" /></td>
            </tr>
            <tr>
                <td>Data de Nascimento:</td>
                <td><input type="date" size="10" name="txtData" /></td>
            </tr>
            <tr>
                <td>Período</td>
                <td><select size="1" name="cmbPeriodo">
                    <option>Manhã</option>
                    <option>Tarde</option>
                    <option>Noite</option>
                </select></td>
            </tr>
            <tr>
                <td colspan="2"><input type="submit" value="Enviar"></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

# atualizar.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Sistema Acadêmico</title>
</head>
<body>
  <form action="../../ServletAluno?cmd=atu" method="post">
    <table border="1">
      <tr>
        <th colspan="2"><h2>Atualizar Aluno</h2></th>
      </tr>
      <tr>
        <td>RA do Aluno:</td>
        <td><input type="text" name="txtRa" size="10" /></td>
      </tr>
      <tr>
        <th colspan="2"><input type="submit" value="Enviar"></th>
      </tr>
    </table>
  </form>
</body>
</html>
```

# consultar.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Sistema Acadêmico</title>
</head>
<body>
  <form action="../../ServletAluno?cmd=con" method="post">
    <table border="1">
      <tr>
        <th colspan="2"><h2>Consultar Alunos</h2></th>
      </tr>
      <tr>
        <td>RA do Aluno:</td>
        <td><input type="text" name="txtRa" size="10" /></td>
      </tr>
      <tr>
        <th colspan="2"><input type="submit" value="Enviar"></th>
      </tr>
    </table>
  </form>
</body>
</html>
```

# excluir.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Sistema Acadêmico</title>
</head>
<body>
  <form action="../../ServletAluno?cmd=exc" method="post">
    <table border="1">
      <tr>
        <th colspan="2"><h2>Excluir Aluno</h2></th>
      </tr>
      <tr>
        <td>RA do Aluno:</td>
        <td><input type="text" name="txtRa" size="10" /></td>
      </tr>
      <tr>
        <th colspan="2"><input type="submit" value="Enviar"></th>
      </tr>
    </table>
  </form>
</body>
</html>
```

# mostrarAlunos.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="java.text.SimpleDateFormat"%>
<%@ page import="java.util.ArrayList"%>
<%@ page import="java.util.List"%>
<%@ page import="br.edu.exemplo.model.Aluno"%>
<!DOCTYPE html>
<html>
<head>
    <title>Lista Alunos</title>
</head>
<body>
    <table width="60%" border="1">
        <tr>
            <th colspan="6"><h1>Lista de Alunos</h1></th>
        </tr>
        <tr>
            <th>ID - Atualizar</th>
            <th>Nome</th>
            <th>Endereço</th>
            <th>E-mail</th>
            <th>Nascimento</th>
            <th>Período</th>
        </tr>
        <%
SimpleDateFormat data = new SimpleDateFormat("dd/MM/yyyy");
List<Aluno> lista = new ArrayList<Aluno>();
lista = (ArrayList) request.getAttribute("alunosList");
for (Aluno a : lista) {
%>
            <tr>
                <td><%=a.getRa()%></td>
                <td><%=a.getNome()%></td>
                <td><%=a.getEndereco()%></td>
                <td><%=a.getEmail()%></td>
                <td><%=data.format(a.getDataNascimento())%></td>
                <td><%=a.getPeriodo()%></td>
            </tr>
        <% } %>
        <tr>
            <th colspan="6"><a href="index.html">Página Principal</a></th>
        </tr>
    </table>
</body>
</html>
```

# atualizarAluno.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="UTF-8"%>
<%@page import="java.text.SimpleDateFormat"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Sistema Acadêmico</title>
</head>
<body>
    <h2>Atualizar Aluno</h2>
    <jsp:useBean id="aluno" scope="session" class="br.edu.exemplo.model.Aluno" />
    <% SimpleDateFormat data = new SimpleDateFormat("dd/MM/yyyy");%>
    <form action="ServletAluno?cmd=atualizar" method="post">
        <table border="1">
            <tr>
                <td>RA:</td>
                <td><input type="text" size="60" name="txtRa" value="<%=aluno.getRa()%>"
                    readonly="readonly" /></td>
            </tr>
            <tr>
                <td>Nome:</td>
                <td><input type="text" size="60" name="txtNome" value="<%=aluno.getNome()%>"
                    /></td>
            </tr>
            <tr>
                <td>E-mail:</td>
                <td><input type="text" size="60" name="txtEmail" value="<%=aluno.getEmail()%>"
                    /></td>
            </tr>
            <tr>
                <td>Endereço:</td>
                <td><input type="text" size="60" name="txtEndereco"
                    value="<%=aluno.getEndereco()%>" /></td>
            </tr>
            <tr>
                <td>Nascimento (dd/mm/aaaa):</td>
                <td><input type="date" size="60" name="txtData"
                    value="<%=aluno.getDataNascimento()%>" /></td>
            </tr>
            <tr>
                <td>Período:</td>
                <td><select name="cmbPeriodo" size="1">
                    <option><%=aluno.getPeriodo()%></option>
                    <option>Manhã</option>
                    <option>Tarde</option>
                    <option>Noite</option>
                </select>
            </td>
            </tr>
            <tr>
                <th colspan="2"><input type="submit" value="Confirmar Atualização" /></th>
            </tr>
            <tr>
                <th colspan="2"><a href="index.html">Página Principal</a></th>
            </tr>
        </table>
    </form>
</body>
</html>
```

# consultarAluno.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="UTF-8"%>
<%@page import="java.text.SimpleDateFormat"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Sistema Acadêmico</title>
</head>
<body>
    <h2>Consultar Aluno</h2>
    <jsp:useBean id="aluno" scope="session" class="br.edu.exemplo.modelo.Aluno" />
    <% SimpleDateFormat data = new SimpleDateFormat("dd/MM/yyyy"); %>
    <form>
        <table border="1">
            <tr>
                <td>RA:</td>
                <td><input type="text" size="60" name="txtRa"
                    value="<%=aluno.getRa()%>" readonly="readonly" /></td>
            </tr>
            <tr>
                <td>Nome:</td>
                <td><input type="text" size="60" name="txtNome"
                    value="<%=aluno.getNome()%>" readonly="readonly" /></td>
            </tr>
            <tr>
                <td>E-mail:</td>
                <td><input type="text" size="60" name="txtEmail"
                    value="<%=aluno.getEmail()%>" readonly="readonly" /></td>
            </tr>
            <tr>
                <td>Endereço:</td>
                <td><input type="text" size="60" name="txtEndereco"
                    value="<%=aluno.getEndereco()%>" readonly="readonly" /></td>
            </tr>
            <tr>
                <td>Nascimento (dd/mm/aaaa):</td>
                <td><input type="text" size="60" name="txtData" readonly="readonly"
                    value="<%=data.format(aluno.getDataNascimento())%>" /></td>
            </tr>
            <tr>
                <td>Período:</td>
                <td><input type="text" size="60" name="txtIdade"
                    value="<%=aluno.getPeriodo()%>" readonly="readonly" /></td>
            </tr>
            <tr>
                <th colspan="2"><a href="index.html">Página Principal</a></th>
            </tr>
        </table>
    </form>
</body>
</html>
```

# excluirAluno.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="UTF-8"%>
<%@page import="java.text.SimpleDateFormat"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Sistema Acadêmico</title>
</head>
<body>
    <h2>Excluir Aluno</h2>
    <jsp:useBean id="aluno" scope="session" class="br.edu.exemplo.modelo.Aluno" />
    <% SimpleDateFormat data = new SimpleDateFormat("dd/MM/yyyy"); %>
    <form action="ServletAluno?cmd=excluir" method="post">
        <table border="1">
            <tr>
                <td>RA:</td>
                <td><input type="text" size="60" name="txtRa"
                    value="<%=aluno.getRa()%>" readonly="readonly" /></td>
            </tr>
            <tr>
                <td>Nome:</td>
                <td><input type="text" size="60" name="txtNome"
                    value="<%=aluno.getNome()%>" /></td>
            </tr>
            <tr>
                <td>E-mail:</td>
                <td><input type="text" size="60" name="txtEmail"
                    value="<%=aluno.getEmail()%>" /></td>
            </tr>
            <tr>
                <td>Endereço:</td>
                <td><input type="text" size="60"
                    name="txtEndereco" value="<%=aluno.getEndereco()%>" /></td>
            </tr>
            <tr>
                <td>Nascimento (dd/mm/aaaa):</td>
                <td><input type="text" size="60" name="txtData"
                    value="<%=data.format(aluno.getDataNascimento())%>" /></td>
            </tr>
            <tr>
                <td>Período:</td>
                <td><input type="text" size="60" name="txtIdade"
                    value="<%=aluno.getPeriodo()%>" readonly="readonly" /></td>
            </tr>
            <tr>
                <th colspan="2"><input type="submit" value="Confirmar Exclusão" /></th>
            </tr>
            <tr>
                <th colspan="2"><a href="index.html">Página Principal</a></th>
            </tr>
        </table>
    </form>
</body>
</html>
```



# ConnectionFactory.java

```
package br.edu.exemplo.util;
import java.sql.*;
public class ConnectionFactory {
    public static Connection getConnection() throws Exception {
        try {
            // indica qual é o banco de dados que estou utilizando e seu driver
            Class.forName("com.mysql.jdbc.Driver");
            // estabelece a conexão e retorna uma conexão
            String login = "root";
            String senha = "";
            String url = "jdbc:mysql://localhost:3306/academico";
            return DriverManager.getConnection(url,login,senha);
        } catch (Exception e) {
            throw new Exception(e.getMessage());
        }
    }
    // fecha uma conexão de três formas: conn, stmt, rs
    public static void closeConnection(Connection conn, Statement stmt,
    ResultSet rs) throws Exception {
        close(conn, stmt, rs);
    }
    public static void closeConnection(Connection conn, Statement stmt)
    throws Exception {
        close(conn, stmt, null);
    }
    public static void closeConnection(Connection conn) throws Exception {
        close(conn, null, null);
    }
    private static void close(Connection conn, Statement stmt, ResultSet rs)
    throws Exception {
        try {
            if (rs != null)
                rs.close();
            if (stmt != null)
                stmt.close();
            if (conn != null)
                conn.close();
        } catch (Exception e) {
            throw new Exception(e.getMessage());
        }
    }
}
```

# Aluno.java

```
package br.edu.exemplo.modelo;
import java.util.Date;
public class Aluno {
    /* Java Bean - É uma classe que contenha todos os atributos privados, possua getters e setters
    para seus atributos e usada para encapsular e abstrair uma entidade */
    // atributos
    private int ra;
    private String nome;
    private String email;
    private String endereco;
    private Date dataNascimento;
    private String periodo;
    // construtor
    public Aluno(int ra, String nome, String email, String endereco, Date dataNascimento, String
    periodo) {
        this.ra = ra;
        this.nome = nome;
        this.email = email;
        this.endereco = endereco;
        this.dataNascimento = dataNascimento;
        this.periodo = periodo;
    }
    // construtor vazio
    public Aluno() {}
    // getters e setters
    public int getRa() {
        return ra;
    }
    public void setRa(int ra) {
        this.ra = ra;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getEndereco() {
        return endereco;
    }
    public void setEndereco(String endereco) {
        this.endereco = endereco;
    }
    public Date getDataNascimento() {
        return dataNascimento;
    }
    public void setDataNascimento(Date dataNascimento) {
        this.dataNascimento = dataNascimento;
    }
    public String getPeriodo() {
        return periodo;
    }
    public void setPeriodo(String periodo) {
        this.periodo = periodo;
    }
}}
```

# ServletAluno.java

```
package br.edu.exemplo.controller;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;
import br.edu.exemplo.dao.AlunoDAO;
import br.edu.exemplo.model.Aluno;
/**
 * Servlet implementation class ServletAluno
 */
@WebServlet("/ServletAluno")
public class ServletAluno extends HttpServlet {
    private static final long serialVersionUID = 1L;
    // metodo para conversao de String para data
    private Date strToDate(String data) throws Exception {
        SimpleDateFormat formato = new SimpleDateFormat("yyyy-MM-dd");
        Date dataF = formato.parse(data);
        return dataF;
    }
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        Aluno aluno = new Aluno();
        AlunoDAO dao;
        String cmd = request.getParameter("cmd");
        try {
            if (cmd.equals("incluir") || cmd.equals("atualizar") ) {
                aluno.setRa(Integer.parseInt(request.getParameter("txtRa")));
                aluno.setNome(request.getParameter("txtNome"));
                aluno.setEmail(request.getParameter("txtEmail"));
                aluno.setEndereco(request.getParameter("txtEndereco"));
                aluno.setPeriodo(request.getParameter("cmbPeriodo"));
                aluno.setDataNascimento(strToDate(request.getParameter("txtData")));
            } else {
                aluno.setRa(Integer.parseInt(request.getParameter("txtRa")));
            }
        } catch (Exception e) {
            //System.out.println("Erro na data");
            System.out.println(e.getMessage());
        }
        try {
            dao = new AlunoDAO();
            // Dispatcher - direciona para uma nova página
            RequestDispatcher rd = null;
            if (cmd.equalsIgnoreCase("incluir")) {
                dao.salvar(aluno);
                rd = request.getRequestDispatcher("ServletAluno?cmd=listar");
            } else if (cmd.equalsIgnoreCase("listar")) {
                List<Aluno> alunosList = dao.todosAlunos();
                request.setAttribute("alunosList", alunosList);
                rd = request.getRequestDispatcher("jsp/mostrarAlunos.jsp");
            } else if (cmd.equalsIgnoreCase("atu")) {
                aluno = dao.procurarAluno(aluno.getRa());
                HttpSession session = request.getSession(true);
                session.setAttribute("aluno", aluno);
            }
        }
    }
}
```

```

        rd = request.getRequestDispatcher("jsp/atualizarAluno.jsp");
    } else if (cmd.equalsIgnoreCase("atualizar")) {
        dao.atualizar(aluno);
        rd = request.getRequestDispatcher("ServletAluno?cmd=listar");
    } else if (cmd.equalsIgnoreCase("con")) {
        aluno = dao.procurarAluno(aluno.getRa());
        HttpSession session = request.getSession(true);
        session.setAttribute("aluno", aluno);
        rd = request.getRequestDispatcher("jsp/consultarAluno.jsp");
    } else if (cmd.equalsIgnoreCase("exc")) {
        aluno = dao.procurarAluno(aluno.getRa());
        HttpSession session = request.getSession(true);
        session.setAttribute("aluno", aluno);
        rd = request.getRequestDispatcher("jsp/excluirAluno.jsp");
    } else if (cmd.equalsIgnoreCase("excluir")) {
        dao.excluir(aluno);
        rd = request.getRequestDispatcher("ServletAluno?cmd=listar");
    }
    // executa a ação de direcionar para a página JSP
    rd.forward(request, response);
} catch (Exception e) {
    System.out.println(e.getMessage());
}
}

protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}
}

```

# AlunoDAO.java

```
package br.edu.exemplo.dao;
import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import br.edu.exemplo.model.Aluno;
import br.edu.exemplo.util.ConnectionFactory;

public class AlunoDAO {
    // classes de banco de dados
    private Connection conn; // abre a conexao do banco de dados
    private PreparedStatement ps; // permite que scripts SQL sejam executados a partir do Java
    private ResultSet rs; // representa as tabelas
    // classe JavaBean
    private Aluno aluno;

    public AlunoDAO() throws Exception {
        // chama a classe ConnectionFactory e estabelece uma conexao
        try {
            this.conn = ConnectionFactory.getConnection();
        } catch (Exception e) {
            throw new Exception("erro: \n" + e.getMessage());
        }
    }

    // metodo de salvar
    public void salvar(Aluno aluno) throws Exception {
        if (aluno == null)
            throw new Exception("O valor passado nao pode ser nulo");
        try {
            String SQL = "INSERT INTO alunos (ra, nome, email, endereco, datanascimento, periodo) values (?, ?, ?, ?, ?, ?)";
            ps = conn.prepareStatement(SQL);
            ps.setInt(1, aluno.getRa());
            ps.setString(2, aluno.getNome());
            ps.setString(3, aluno.getEmail());
            ps.setString(4, aluno.getEndereco());
            ps.setDate(5, new java.sql.Date(aluno.getDataNascimento().getTime()));
            ps.setString(6, aluno.getPeriodo());
            ps.executeUpdate();
        } catch (SQLException sqle) {
            throw new Exception("Erro ao inserir dados " + sqle);
        } finally {
            ConnectionFactory.closeConnection(conn, ps);
        }
    }

    // metodo de atualizar
    public void atualizar(Aluno aluno) throws Exception {
        if (aluno == null)
            throw new Exception("O valor passado nao pode ser nulo");
        try {
            String SQL = "UPDATE alunos set nome=?, email=?, endereco=?, datanascimento=?, "
                + "periodo=? WHERE ra=?";
            ps = conn.prepareStatement(SQL);
            ps.setString(1, aluno.getNome());
            ps.setString(2, aluno.getEmail());
            ps.setString(3, aluno.getEndereco());
            ps.setDate(4, new java.sql.Date(aluno.getDataNascimento().getTime()));
        }
    }
}
```

```

        ps.setString(5, aluno.getPeriodo());
        ps.setInt(6, aluno.getRa());
        ps.executeUpdate();
    } catch (SQLException sqle) {
        throw new Exception("Erro ao alterar dados " + sqle);
    } finally {
        ConnectionFactory.closeConnection(conn, ps);
    }
}

// metodo de excluir
public void excluir(Aluno aluno) throws Exception {
    if (aluno == null)
        throw new Exception("O valor passado nao pode ser nulo");
    try {
        String SQL = "DELETE FROM alunos WHERE ra = ?";
        ps = conn.prepareStatement(SQL);
        ps.setInt(1, aluno.getRa());
        ps.executeUpdate();
    } catch (SQLException sqle) {
        throw new Exception("Erro ao excluir dados " + sqle);
    } finally {
        ConnectionFactory.closeConnection(conn, ps);
    }
}

// Procurar Aluno
public Aluno procurarAluno(int raAluno) throws Exception {
    try {
        String SQL = "SELECT * FROM alunos WHERE ra=?";
        ps = conn.prepareStatement(SQL);
        ps.setInt(1, raAluno);
        rs = ps.executeQuery();
        if (rs.next()) {
            int ra = rs.getInt("ra");
            String nome = rs.getString("nome");
            String email = rs.getString("email");
            String endereco = rs.getString("endereco");
            Date dataNascimento = rs.getDate("dataNascimento");
            String periodo = rs.getString("periodo");
            aluno = new Aluno(ra, nome, email, endereco, dataNascimento, periodo);
        }
        return aluno;
    } catch (SQLException sqle) {
        throw new Exception(sqle);
    } finally {
        ConnectionFactory.closeConnection(conn, ps, rs);
    }
}

// Listar todos os alunos
public List<Aluno> todosAlunos() throws Exception {
    try {
        ps = conn.prepareStatement("SELECT * FROM alunos");
        rs = ps.executeQuery();
        List<Aluno> list = new ArrayList<Aluno>();
        while (rs.next()) {
            int ra = rs.getInt("ra");
            String nome = rs.getString("nome");
            String email = rs.getString("email");
            String endereco = rs.getString("endereco");
            Date dataNascimento = rs.getDate("dataNascimento");
            String periodo = rs.getString("periodo");
            list.add(new Aluno(ra, nome, email, endereco, dataNascimento, periodo));
        }
        return list;
    }
}

```

```
    } catch (SQLException sqle) {  
        throw new Exception(sqle);  
    } finally {  
        ConnectionFactory.closeConnection(conn, ps, rs);  
    }  
}  
}
```