

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE MATO GROSSO DO SUL
CÂMPUS COXIM
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA
INTERNET
PERÍODO NOTURNO
TURMA 32110
PROGRAMAÇÃO PARA SERVIDORES WEB**

Maicon Cezar Azambuja da Cunha

**PRIMEIRA ATIVIDADE: DESENVOLVIMENTO DE UM EXEMPLO
DE JSF**

INTRODUÇÃO

Este trabalho é uma atividade prática da disciplina “Programação de Servidores Web”, do curso superior de Tecnologia em Sistemas para Internet, ministrada pelo professor Gustavo Maruyama, no 5º semestre, no Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul – Câmpus Coxim.

O objetivo da atividade foi aprofundar o aprendizado em JavaServer Faces (JSF), aplicando conceitos de desenvolvimento de aplicações web dinâmicas, com foco na separação entre lógica de negócio e apresentação e na utilização de ManagedBeans com diferentes tipos de escopo.

A prática consistiu na criação de um aplicativo para cálculo do Índice de Massa Corporal (IMC), que recebe peso e altura do usuário e retorna a classificação do IMC conforme a tabela da Wikipedia.

DESENVOLVIMENTO

1. Pesquisa sobre JavaServer Faces

O JavaServer Faces (JSF) é um framework Java para desenvolvimento de interfaces web dinâmicas, parte da plataforma Jakarta EE. JSF permite criar componentes reutilizáveis de interface e gerenciar eventos e interações do usuário, simplificando a integração entre front-end (HTML/CSS) e back-end (Java).

Bibliotecas necessárias:

- *jakarta.jakartaee-api 10.0.0* (fornece todas as APIs do Jakarta EE 10, incluindo JSF 3).
- Servidor de aplicação compatível: Payara 6 (ou GlassFish 6).

2. ManagedBean e Tipos de Escopo

Um *Managed Bean* é uma classe Java que serve como controlador da aplicação, gerenciando dados do usuário e ações da interface.

Exemplo de bean usado no projeto:

Java

```
package bean;

import jakarta.inject.Named;
import jakarta.enterprise.context.SessionScoped;
import java.io.Serializable;

@Named("imcBean")
@SessionScoped
public class ImcBean implements Serializable {
    private double peso;
    private double altura;
    private double imc;
    private String classificacao;

    public void calcular() {
        imc = peso / (altura * altura);

        if (imc < 18.5) classificacao = "Abaixo do peso";
        else if (imc < 24.9) classificacao = "Peso normal";
        else if (imc < 29.9) classificacao = "Sobrepeso";
    }
}
```

```

else if (imc < 34.9) classificacao = "Obesidade grau I";
else if (imc < 39.9) classificacao = "Obesidade grau II";
else classificacao = "Obesidade grau III";
}

// Getters e Setters
public double getPeso() { return peso; }
public void setPeso(double peso) { this.peso = peso; }
public double getAltura() { return altura; }
public void setAltura(double altura) { this.altura = altura; }
public double getImc() { return imc; }
public String getClassificacao() { return classificacao; }
}

```

Tipos de escopo em JSF:

- *@RequestScoped*: o bean é criado a cada requisição HTTP. Ideal para operações rápidas e sem persistência de dados entre páginas.
- *@ViewScoped*: mantém o estado do bean enquanto o usuário permanece na mesma página.
- *@SessionScoped*: mantém o bean durante toda a sessão do usuário. Usado no nosso projeto para preservar os dados do IMC até o usuário sair da aplicação.
- *@ApplicationScoped*: único bean compartilhado entre todos os usuários. Usado para dados globais da aplicação.

3. Implementação do formulário

index.xhtml:

```

Java
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
  <title>Calculadora de IMC</title>
</h:head>
<h:body>
  <h1>Calcular IMC</h1>
  <h:form>
    <h:outputLabel value="Peso (kg): " for="peso"/>

```

```

<h:inputText id="peso" value="#{imcBean.peso}" /><br/><br/>

<h:outputLabel value="Altura (m): " for="altura"/>
<h:inputText id="altura" value="#{imcBean.altura}" /><br/><br/>

<h:commandButton value="Calcular" action="resultado.xhtml" />
</h:form>
</h:body>
</html>

```

resultado.xhtml:

```

Java
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
  <title>Resultado do IMC</title>
</h:head>
<h:body>
  <h1>Resultado do IMC</h1>
  <p>IMC: #{imcBean.imc}</p>
  <p>Classificação: #{imcBean.classificacao}</p>
  <h:link outcome="index.xhtml" value="Voltar"/>
</h:body>
</html>

```

4. Observações Técnicas

- Arquivos .xhtml são interpretados pelo Faces Servlet configurado no web.xml.
- O bean ImcBean é SessionScoped para manter os dados entre páginas.
- O Maven garante que a dependência JSF esteja disponível, e o Payara executa a aplicação corretamente.

CONCLUSÃO

Com esta atividade, foi possível compreender o funcionamento do JSF e como ele facilita a criação de interfaces web dinâmicas. Também foi possível aplicar o conceito de *Managed Beans* e explorar os diferentes tipos de escopo, como Request, View, Session e Application. Durante o desenvolvimento, criou-se uma aplicação prática de cálculo de IMC, consolidando a integração entre front-end e back-end. Além disso, aprendeu-se sobre a configuração de um projeto Maven Web com Payara 6, utilizando arquivos .xhtml, o web.xml e realizando a navegação entre páginas. Essa experiência reforçou conceitos importantes, como a separação de responsabilidades, a persistência de dados em sessões e o uso do JSF como framework de apresentação, que são essenciais para aplicações web corporativas em Java.