

---

# Programação Web

**Professor: Diego Oliveira**



**Conteúdo 19:  
JSF5**





# Navegação JSF

- A navegação é o ato de ir de uma página a outra
- As duas formas de navegação do JSF são conhecidas como:
  - Explícita
  - Implícita
- A navegação também pode ser estática, através de links, ou dinâmica, que analisa o retorno dos métodos do ManagedBean





# Navegação Implícita

- A Navegação Implícita é a mais simples de se executar
- Um sinal (*outcome*) é enviado ao framework e um tratador de navegação identifica qual tela deve ser apresentada como resposta
- Esse *outcome* é uma String de retorno:
  - se um método do MB retornar uma String
  - uma String no atributo **action** de um `commandButton` ou `commandLink`
  - uma String no atributo **outcome** de um link





# Navegação Implícita

- Um link em uma página pode levar a uma outra página através do atributo outcome

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:body>
VOCÊ ESTÁ NA PÁGINA INDEX <br></br>
<h:link outcome="pagina1">PAGINA 1</h:link>
</h:body>
</html>
```

- Essa é a forma mais simples de navegar



- Para entrar em pastas utilize a /

- Para voltar utilize ..



# Navegação Explícita

- Para a executar a Navegação Explícita é preciso configurar o arquivo faces-config.xml
- Precisamos de pelo menos 3 informações:
  - caminho da tela de origem
  - outcome
  - caminho da tela de destino
- Esta técnica tem vantagens e desvantagens
- Para grandes sistemas com muitas páginas o arquivo fica difícil de manter





# Navegação Explícita

- Exemplo:

```
<faces-config xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-facesconfig_1_2.xsd"
  version="1.2">
  <navigation-rule>
    <from-view-id>/index.xhtml</from-view-id>
    <navigation-case>
      <from-outcome>pagina1</from-outcome>
      <to-view-id>/pagina1.xhtml</to-view-id>
    </navigation-case>
  </navigation-rule>
</faces-config>
```





# Navegação Dinâmica

- Já vimos a navegação estática, onde se diz diretamente no link para qual página ele vai
- Agora veremos a dinâmica, na qual podemos decidir entre uma página e outra no mesmo botão ou link, utilizando o MB
- Na página index ficaria tudo igual, mas a avaliação passa a ser no MB, que irá decidir para qual página o sistema irá



- Vejamos um exemplo:



# Navegação Dinâmica

- Exemplo:

```
<h:body>
  VOCÊ ESTÁ NA PÁGINA INDEX <br></br>
  <!--<h:link outcome="paginal">PAGINA 1</h:link>-->
  <h:form>
    <h:inputText value="#{meuMB.nome}"></h:inputText>
    <h:commandButton action="#{meuMB.escolherPagina()}"
      value="VAI"></h:commandButton>
  </h:form>
</h:body>
```

- Se o usuário digitar 'diego' irá para a página 'diego.xhtml', senão irá para 'outro.xhtml'

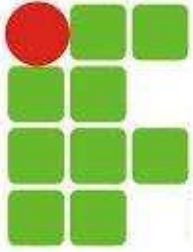


```
public class MeuMB {

    private String nome;

    public String escolherPagina() {
        if (nome.equals("diego")) {
            return "diego";
        } else {
            return "outro";
        }
    }
}
```

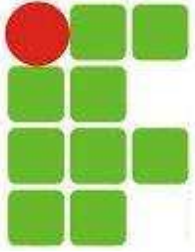




# Exercício

- Crie um formulário onde o aluno colocará as 4 notas tiradas ao longo do ano no IFRN
- Faça um formulário organizado em tabela
- Ao clicar em CALCULAR o sistema deverá verificar se o aluno foi APROVADO, REPROVADO ou está em RECUPERAÇÃO
- Utilize NAVEGAÇÃO DINÂMICA com métodos no MB
- Cada resultado deverá encaminhar para uma página diferente, com os respectivos designs (tela verde para aprovado, tela vermelha para reprovado e tela laranja para recuperação)
- Na tela de recuperação mostre a nota necessária para passar





---

# Perguntas?

---

