



PROGRAMAÇÃO SERVIDOR EM SISTEMAS WEB

SESSÃO E LOGIN

Prof. Dr. Daniel Caetano

2011 - 2

Visão Geral

1

- Estados de Componente

2

- Estados Relacionados ao Login

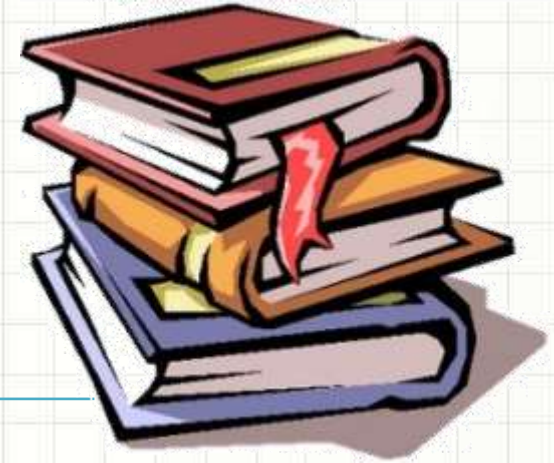
3

- A Sessão e Seu Ciclo de Vida

4

- Componente de Logon

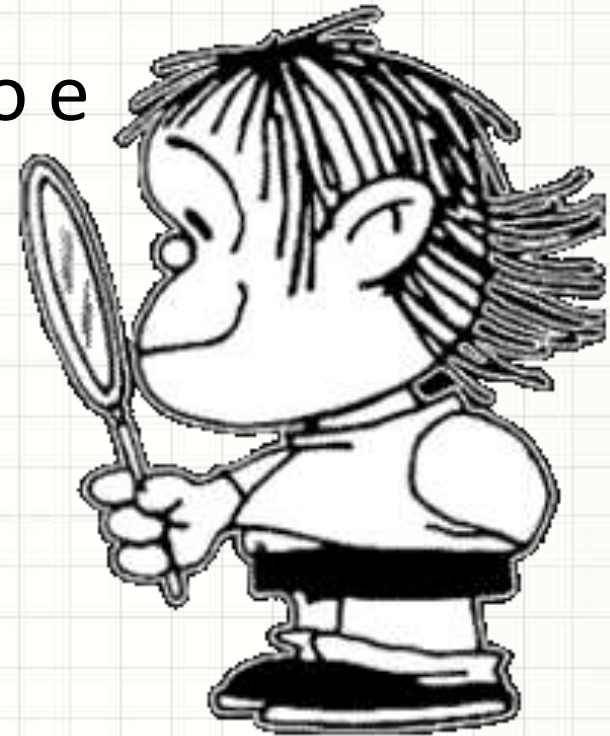
Material de Estudo



Material	Acesso ao Material
Notas de Aula	http://www.caetano.eng.br/aulas/psw/ (Aula 11)
Apresentação	http://www.caetano.eng.br/aulas/psw/ (Aula 11)
Material Didático	Padrões de Arquitetura de Aplicações Corporativas, páginas 93 a 98
Biblioteca Virtual	-

Objetivos

- Compreender o que são estados de um componente
- Compreender o que é a Sessão e como pode ser usada
- Capacitar para implementar elementos que armazenem estado na sessão
- **TAREFA!**
 - Aguardem... Atividade Estruturada 2!

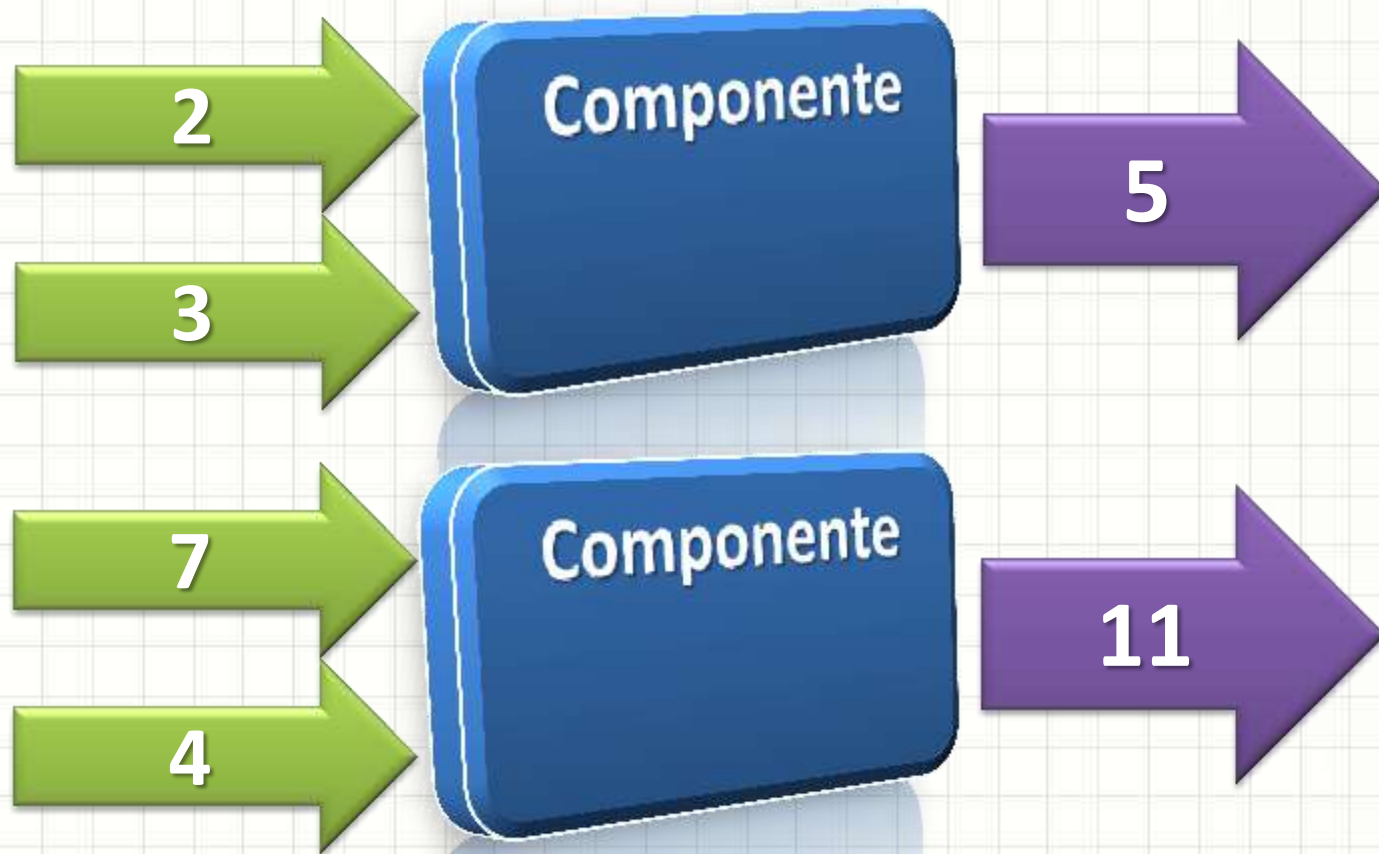




ESTADOS DE COMPONENTE

Estados de Componente

- Comum: mesmos dados \rightarrow mesma saída



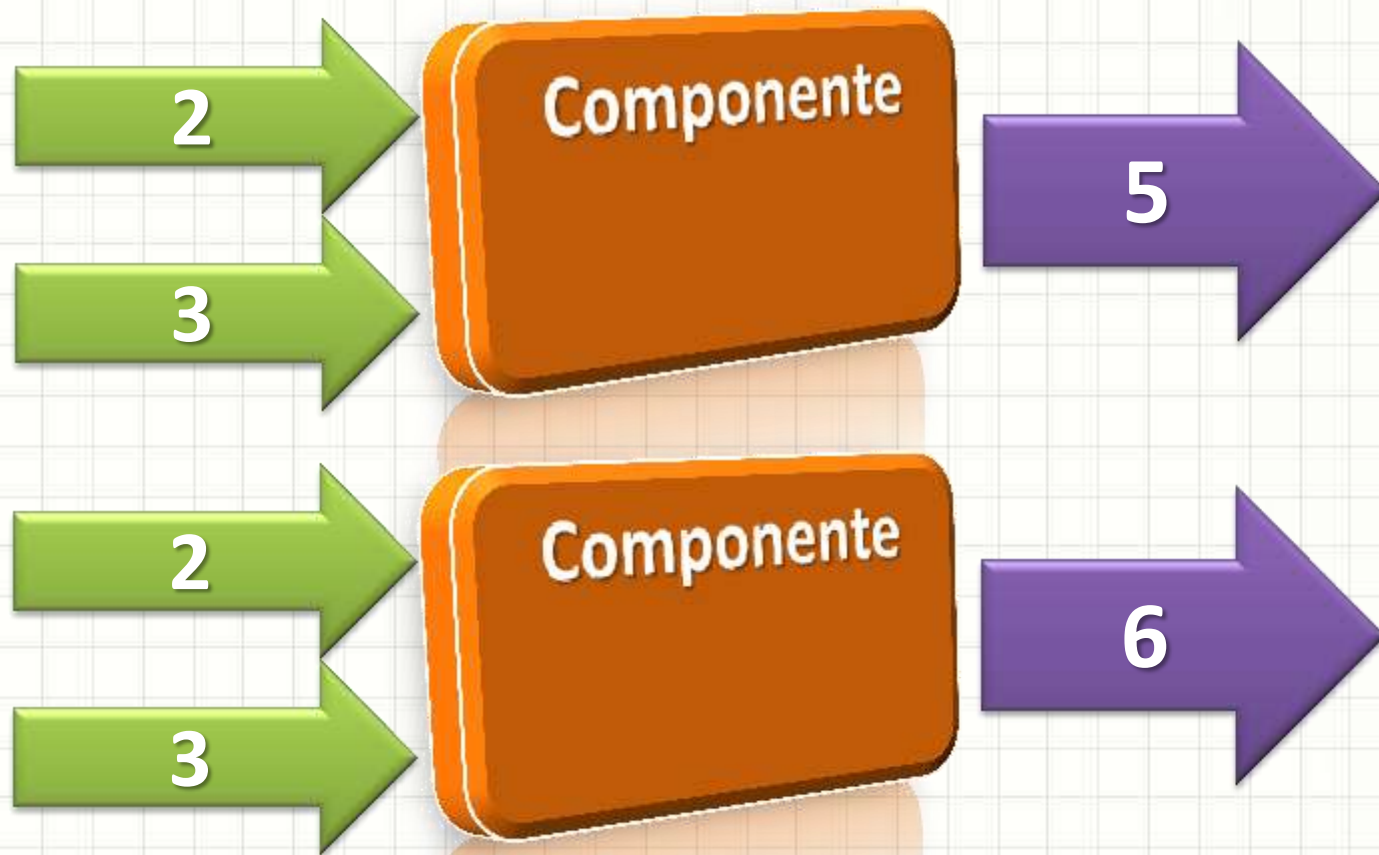
Estados de Componente

- Comum: mesmos dados → mesma saída



Estados de Componente

- Outro caso: mesmos dados \rightarrow saídas diferentes



Estados de Componente

- Outro caso: mesmos dados → saídas diferentes



Estados de Componente

- Analogia
 - Considere um pai muito bondoso que não nega nada a seu filho
 - O filho pede dinheiro ao pai
 - O que o pai responde?
- Depende...
- Depende do quê?
- Da situação financeira dele...
 - Do **estado** das finanças dele

Estados de Componente

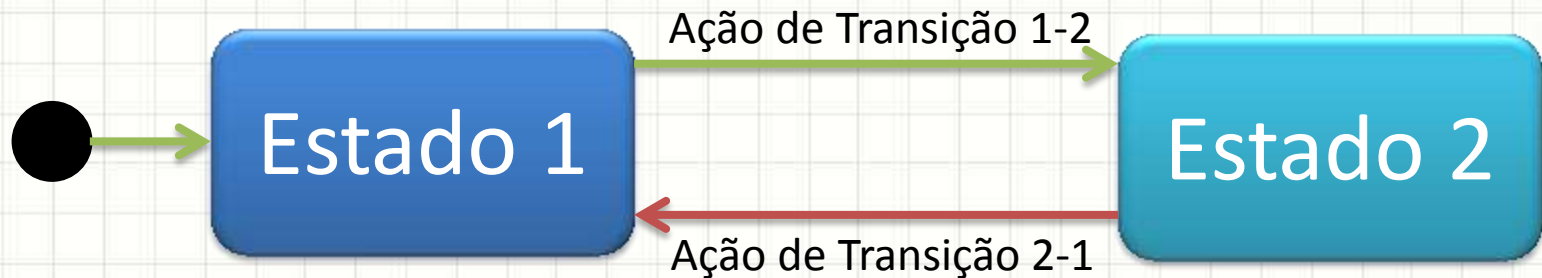
- Comportamento depende de estado?
 - Componente com **estado** → *stateless*
- Comportamento não depende do estado?
 - Componente **sem estado** → *statefull*
- A mudança do estado nem sempre depende do próprio componente
 - O pai pode não ter dinheiro porque gastou
 - O pai pode não ter dinheiro porque foi assaltado

Estados de Componente

- Existe estado em programação?
- Claro!
- O que ocorre se você marcar a página do seu extrato no Banco como “favorito” e depois tenta voltar lá?
- Você será encaminhado para a tela de login
- O componente de extrato tem dois estados:
 - Se o logon foi feito: mostra o extrato
 - Se o logon não foi feito: redireciona para a tela de logon

Estados de Componente

- Como indicar os estados de um componente?
- Diagrama de Estados



Estados de Componente

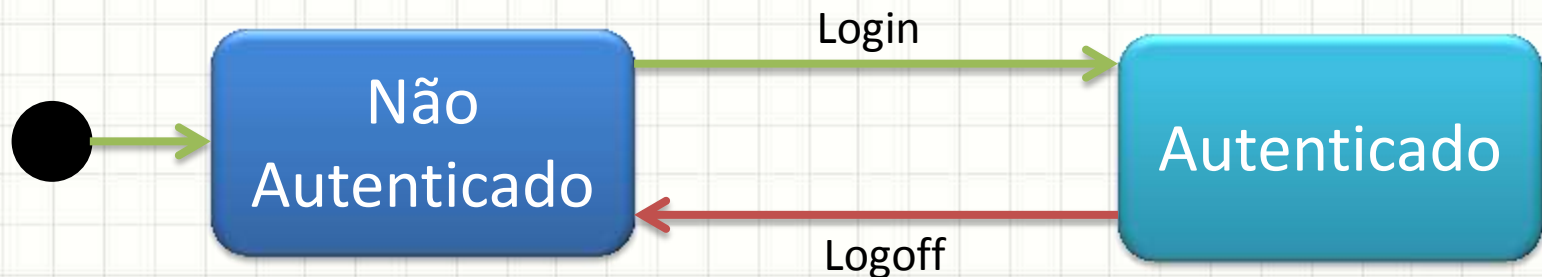
- Um componente precisa saber qual é seu estado para poder agir de acordo...
- Por exemplo...
 - Quais os estados de uma aplicação com Login?



ESTADOS DE LOGIN

Estados de Login

- O que é um login?
- Sistema com Controle de Acesso...
 - Identificação
 - Autenticação
- Os componentes precisam saber QUEM usa
 - Estado 1: Sem Autenticação
 - Estado 2: Com Autenticação



Estados de Login

- **Não autenticado:**
 - Encaminha execução para a tela de login
- **Autenticado:**
 - Componente faz sua função
- Como mudar de um estado para outro?
 - Não autenticado → Autenticado: **Login**
 - Autenticado → Não autenticado: **Logout**
 - Autenticado → Não autenticado: **Timeout**

Estados de Login

- Todo componente: precisa saber estado
 - Como o componente pode saber isso?
- **Armazenar a informação em lugar especial**
- Imagine que exista uma caixa...



- E que tudo que colocarmos nela, fica disponível para qualquer Servlet da aplicação
- Seria perfeito!



SESSÃO

Sessão

- Existe uma “caixa” assim...
- Essa caixa chama-se...
 - **Sessão** (session)
- Temos que criá-la?
- Não... ela já existe...
- Só precisamos pegar a referência para ela:

```
HttpSession session = request.getSession();
```

- Ok... mas como usar isso?



Sessão

- Uma vez que tenhamos a referência...
 - Usamos do mesmo jeito que a request

- Armazenar um objeto

`session.setAttribute("etiqueta",obj)`

- Recuperar um objeto

`Object obj = (Object) session.getAttribute("etiqueta")`

Sessão

- Mas... é o mesmo que guardar na **request**?
- **NÃO!**
- Qual a diferença?
- O que eu guardo em uma request só é passado para um outro servlet com o uso de um **RequestDispatcher.forward()**
- Quando colocamos na sessão, o objeto fica disponível para todos os outros servlets... automaticamente!

Sessão

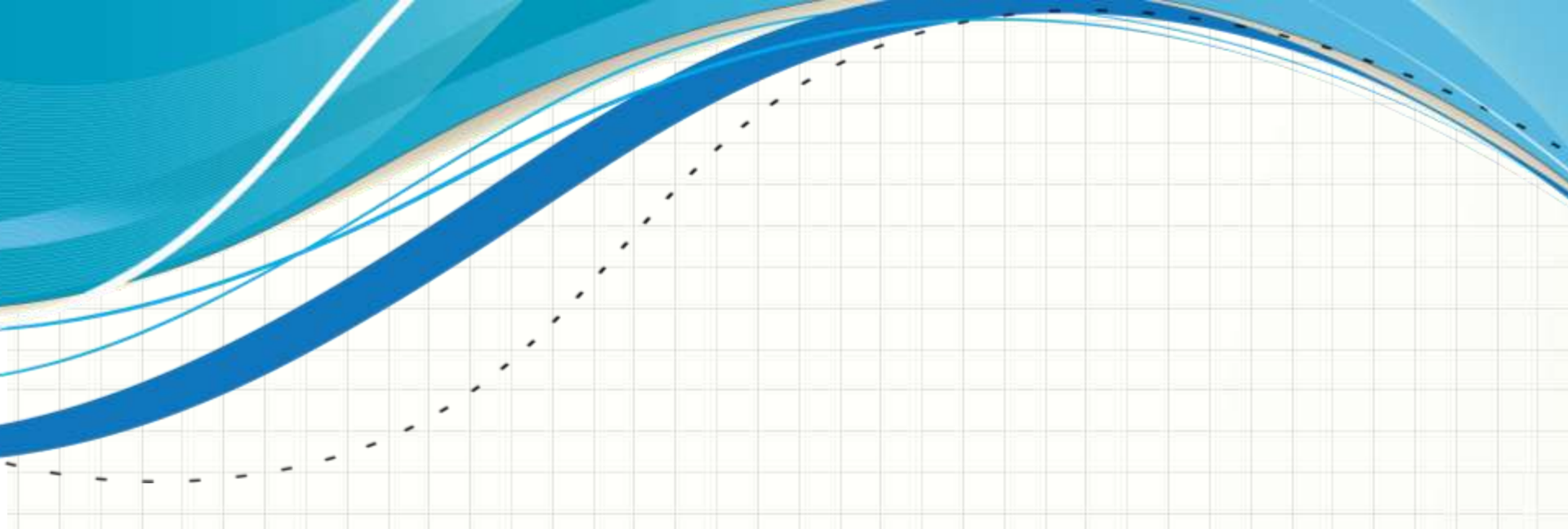
- Até quando um objeto fica na sessão?
- Até quando o programador removê-lo...

`session.removeAttribute("etiqueta")`

- Até quando acabar o tempo da sessão (*timeout*)
 - web.xml
 - Ou...

`session.invalidate()`

- Até o usuário fechar o navegador



IMPLEMENTANDO O COMPONENTE LOGON

Componente de Logon

- Um componente de logon é muito útil
- Acompanhe o professor na implementação
- Observe o uso da sessão!



CONCLUSÕES

Resumo

- Muitos componentes possuem estados
- Os mais comuns são os estados de autenticação de usuário
- Para que possam agir, os componentes precisam de uma “memória” de qual é seu estado
- O recurso usado para implementar essa “memória” é a sessão

Próxima Aula



- A “Sessão” serve só para isso?
 - O que mais podemos fazer com a sessão?



PERGUNTAS?



**BOM DESCANSO
A TODOS!**