

# Java Web

AULA 04 – SERVLETS PARTE II

## Objetivos e Conceitos

- Objetivos:
  - Apresentar a manipulação de cookies e sessões. Exemplo de Login.
- Conceitos:
  - Cookies, Sessões, Login

## Tópicos

---

- Cookies
- Sessões
- Login

## Cookies

---

# Cookies

Pequenas quantidades de informações que podem ser armazenadas no browser (cliente)

- Strings
- CHAVE=VALOR

Pode ser mantido mesmo depois do navegador fechado

Pode ter uma data de expiração

- Pode-se configurar quando expirar

Trafegam como *headers* nas mensagens HTTP

Servidor

- Envia *cookies* para os clientes
- Programador adiciona o código necessário para obter os *cookies* recebidos e enviar *cookies* para os clientes

Cliente

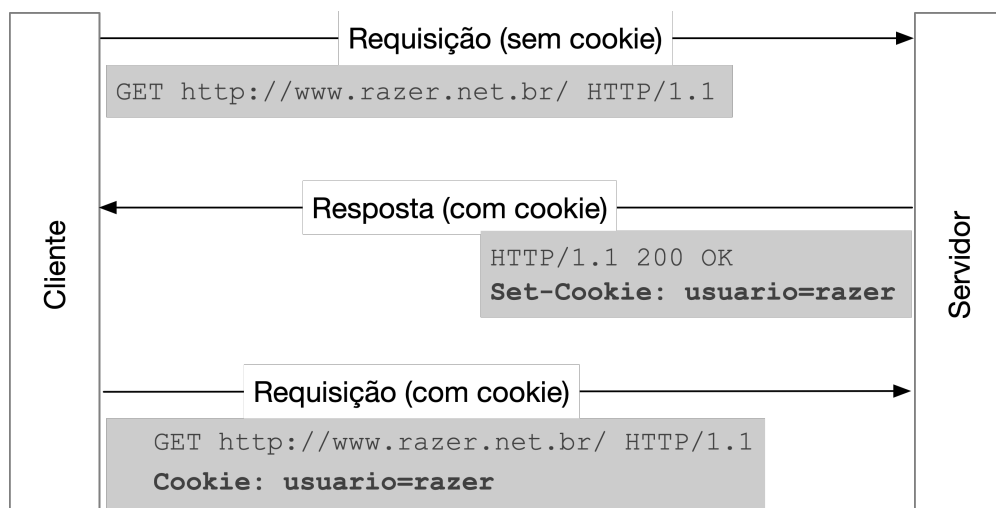
- Quando requisita em servidores para os quais há *cookies*, os envia
- Esse envio é automático

Prof. Dr. Razer A N R Montaña

JAVA WEB

5

# Cookies



Prof. Dr. Razer A N R Montaña

JAVA WEB

6

# Cookies

---

Requisição (sem *cookies* trafegando)

```
GET http://www.razer.net.br/ HTTP/1.1
```

Resposta (Servidor enviando cookies para o cliente)

```
HTTP/1.1 200 OK
Date: Wed, 01 Jan 2020 00:10:10 GMT
Server: Apache/2.2.21
Set-Cookie: usuario=razer
Content-Type: text/html
```

Requisição (Cliente enviando cookies para o servidor)

```
GET http://www.razer.net.br/ HTTP/1.1
Cookie: usuario=razer
```

# Cookies

---

Estes códigos são no servidor (Servlets)

Classe que representa um Cookie:

```
javax.servlet.http.Cookie
```

Para criar um cookie e enviar ao cliente (na resposta)

```
Cookie c = new Cookie("usuario", "razer");  
response.addCookie(c);
```

Pode-se obter todos os cookies enviados pelo cliente (na requisição)

```
Cookie[] cookies = request.getCookies();
```

## Cookies.

Para setar data de expiração (antes de invocar `addCookie()`):

```
c.setMaxAge(60*60*2); // tempo em segundos = 2 horas
response.addCookie(c);
```

Gera um header:

```
Set-Cookie: usuario=razer; max-age=7200
```

Para apagar um *cookie* usa-se 0 como MaxAge:

```
c.setMaxAge(0);
response.addCookie(c);
```

## Exemplo: index.html

Alterar o index.html para conter

```
<html>
<head><title>Dados</title></head>
<body>
  <form action="Processa" method="post">
    <input type="text" name="nome"/><br />
    <input type="password" name="senha"/><br />
    <input type="submit" value="Ok"/>
  </form>
</body>
</html>
```

## Exemplo: Processa.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

@WebServlet(urlPatterns = {"/Processa"})
public class Processa extends HttpServlet {
    public void doPost( HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        PrintWriter out = response.getWriter();
        response.setContentType("text/html");

        String usu = request.getParameter("nome");
        String sen = request.getParameter("senha");

        Cookie c = new Cookie("usuario", usu);
        c.setMaxAge(60 * 60); // 1 hora
        response.addCookie(c);

        out.println("<html><head><title>Teste</title></head><body>Cookie armazenado<br/>");
        out.println("<a href='\"Leitura\"'>Ver</a></body></html>");
        out.flush();
    }
}
```

Prof. Dr. Razer A N R Montañó

JAVA WEB

11

## Exemplo: Leitura.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

@WebServlet(urlPatterns = {"/Leitura"})
public class Leitura extends HttpServlet {
    public void doGet( HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        PrintWriter out = response.getWriter();
        response.setContentType("text/html");

        out.println("<html><head><title>Teste</title></head>");
        out.println("<body>Cookies:<br/>");

        Cookie[] cookies = request.getCookies();
        for (Cookie c: cookies)
            out.println(c.getName() + ": " + c.getValue() + "<br/>");

        out.println("</body></html>");
        out.flush();
    }
}
```

Prof. Dr. Razer A N R Montañó

JAVA WEB

12



## Exercícios..

---

1. Criar e executar a aplicação anterior e testar da seguinte forma:
  - a. Primeiro executar somente Leitura e constatar que o *cookie* `usuario` não existe
  - b. Rodar desde o `index.html` e constatar que o *cookie* `usuario` existe

## Sessões

---

## Sessões

---

### PROBLEMA!!!

- Via HTTP não há como saber se uma requisição vem de um usuário que está solicitando a primeira página ou já está a algum tempo no sítio

### EXEMPLO DE DIÁLOGO:

1. O **cliente** escolhe um colchão para compra
2. O **sistema** reserva o colchão e solicita o método de pagamento
3. O **cliente** escolhe o método de pagamento com cartão de crédito
4. O **sistema** efetua a venda do colchão escolhido no passo 1, via cartão de crédito

## Sessões

---

### Como Resolver?

- Usar componentes de guarda de estado EJB do Jakarta EE (**Stateful Session Beans**)
- Guardar os dados em um **Banco de Dados**

Em qualquer um destes casos, resta a pergunta:

- COMO O SERVIDOR SABE QUE A REQUISIÇÃO 4 VEM DO MESMO CLIENTE DA REQUISIÇÃO 1???



## Sessões

PROBLEMA PIOR AINDA!!!

➤ Temos vários clientes fazendo as mesmas requisições ao mesmo tempo

EXEMPLO DE DIÁLOGO:

1. A **MARIA** escolhe um colchão para compra
1. O **JOÃO** escolhe uma cadeira para compra
2. O **sistema** reserva a CADEIRA do **JOÃO** e solicita o método de pagamento
2. O **sistema** reserva o COLCHÃO da **MARIA** e solicita o método de pagamento
3. A **MARIA** escolhe o método de pagamento com CARTÃO DE CRÉDITO
3. O **JOÃO** escolhe o método de pagamento com BOLETO
4. O **sistema** efetua a venda do COLCHÃO da **MARIA** no passo 1, via CARTÃO DE CRÉDITO
4. O **sistema** efetua a venda da CADEIRA do **JOÃO** no passo 1, via BOLETO

Prof. Dr. Razer A N R Montaña

JAVA WEB

17

## Sessões

Para resolver deve haver uma forma do servidor identificar o requisitante

O servidor deve atribuir um ID a cada cliente

Esse ID é usado para identificar uma estrutura de dados: **SESSÃO**

- Cada usuário terá uma sessão diferente
- Todas as sessões possuem os mesmos dados, mas com valores diferentes

O cliente deve enviar esse ID a cada requisição

Por que não usar o endereço IP como esse ID identificador?

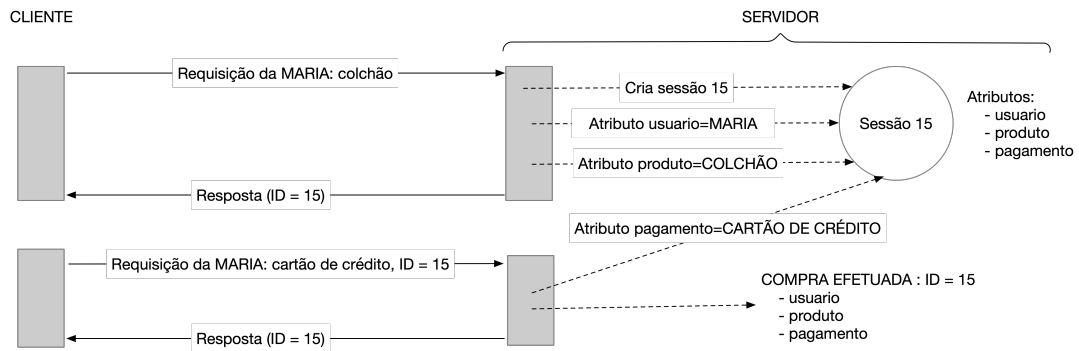
- Porque o IP não identifica o usuário unicamente, se ele estiver em uma rede local, por exemplo
- O IP seria o do roteador, igual para todos atrás desse roteador

Prof. Dr. Razer A N R Montaña

JAVA WEB

18

## Sessões: 1 cliente

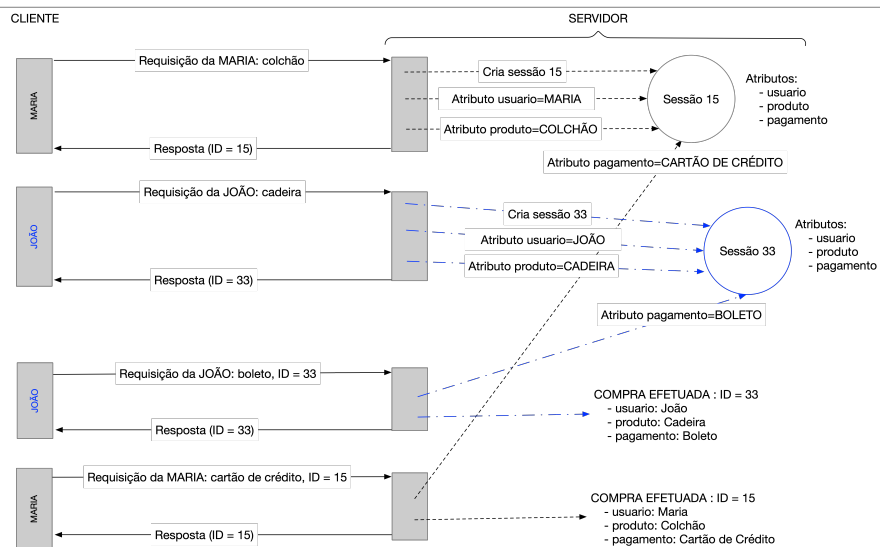


Prof. Dr. Razer A N R Montaña

JAVA WEB

19

## Sessões: Vários clientes



Prof. Dr. Razer A N R Montaña

JAVA WEB

20

## Sessões

---

Para cada usuário, a *servlet* pode criar um objeto **HttpSession**

Possui um identificador: **JSESSIONID**

Está associado ao usuário

Armazena atributos como pares: Chave/Objeto

É acessível a partir de outros *servlets* e JSPs do aplicativo

## Sessões

---

Trafega como um Cookie, mas o programador não precisa manipulá-lo

- Tudo é feito por **HttpSession**

Requisição (sem sessão)

```
GET http://www.razer.net.br/ HTTP/1.1
```

Resposta (Servidor criou a sessão)

```
HTTP/1.1 200 OK
Date: Wed, 01 Jan 2020 00:10:10 GMT
Server: Apache/2.2.21
Set-Cookie: JSESSIONID=AB013DBF14
Content-Type: text/html
```

Requisição (Cliente enviando seu ID)

```
GET http://www.razer.net.br/ HTTP/1.1
Cookie: JSESSIONID=AB013DBF14
```

## Sessões

---

Para obter uma Sessão

- A partir do `HttpServletRequest`
- Automaticamente o ID é criado e o *cookie* JSESSIONID anexado na resposta

Sintaxes

- Cria um nova sessão ou retorna uma já existente

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

- Se a sessão já existir, retorna-a.
- Se não existir e passar TRUE, cria a sessão.
- Se não existir e passar FALSE, não cria a sessão e retorna NULL

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

## Sessões

---

Sessões possuem atributos

- Dados que são dependentes do usuário
- Cada usuário tem um valor diferente para estes atributos

Para setar um atributo na sessão

```
session.setAttribute(<string>, <objeto>);
```

Para obter um atributo da sessão

```
<tipo> x = (<tipo>) session.getAttribute(<string>);
```

## Sessões

---

Exemplo: Setar um atributo objeto na sessão

```
Pessoa razer = new Pessoa();  
razer.setNome("Razer");  
session.setAttribute("eu", razer);
```

Exemplo: Obter um atributo objeto da sessão

```
Pessoa p = (Pessoa)session.getAttribute("eu");
```

## Sessões.

---

Para invalidar uma sessão (ex. logout)

```
HttpSession session = request.getSession(false);  
if (session != null) {  
    session.invalidate();  
}
```

## Exemplo: index.html

Alterar o index.html para conter

```
<html>
<head><title>Dados</title></head>
<body>
  <form action="Processa" method="post">
    <input type="text"      name="usuario"/><br />
    <input type="password" name="senha"/><br />
    <input type="submit"   value="Ok" />
  </form>
</body>
</html>
```

## Exemplo: Processa.java

```
// Importações

@WebServlet(urlPatterns = {"/Processa"})
public class Processa extends HttpServlet {
    public void doPost(HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");

        String usu = request.getParameter("usuario");
        String sen = request.getParameter("senha");

        HttpSession session = request.getSession();
        session.setAttribute("nome", usu);
        session.setAttribute("senha", sen);

        out.println("<html><head><title>Teste</title></head><body>");
        out.println("Dados armazenados<br/>");
        out.println("<a href=\"Leitura\">Ver</a></body></html>");
        out.flush();
    }
}
```

## Exemplo: Leitura.java.

```
// Importações

@WebServlet(urlPatterns = {"/Leitura"})

public class Leitura extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");

        HttpSession session = request.getSession();
        String nome = (String) session.getAttribute("nome");
        String senha = (String) session.getAttribute("senha");

        out.println("<html><head><title>Teste</title></head><body>");
        out.println("nome: " + nome + "<br/>");
        out.println("senha: " + senha + "<br/>");
        out.println("</body></html>");
        out.flush();
    }
}
```

Prof. Dr. Razer A N R Montañó

JAVA WEB

29



## Exercícios..

1. Criar e executar a aplicação anterior e testar da seguinte forma:
  - a. Primeiro executar somente Leitura e constatar que a sessão não contém dados
  - b. Rodar desde o index.html e constatar que a sessão é preenchida

Prof. Dr. Razer A N R Montañó

JAVA WEB

30

# Login

---

## Login Com Sessões

---

Formulário de Login que submete para uma Servlet

Ao submeter do formulário, cria a sessão e guarda uma informação

Arquivos

- `index.html`
- `LoginServlet`
- `OutraServlet`
- `pagina.jsp`
- `LogoutServlet`



## Login Com Sessões.

### `index.html`

- Formulário de login

### `LoginServlet`

- Verifica se o usuário/senha está OK
- Se estiver, armazena algo na sessão, indicando que está logado
- Se não estiver, mostra mensagem de erro

### Todas as demais Servlets/JSPs

- `OutraServlet` e `pagina.jsp`
- Verificar se o usuário está logado
- Se estiver, continua
- Se não estiver, mostra mensagem de erro

### `LogoutServlet`

- Se tem uma sessão ativa, invalida

## index.html

```
<html>
  <head><title>Login</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <form action="LoginServlet" method="post">
      Login: <input type="text" name="login" /><br/>
      Senha: <input type="text" name="senha" /><br/>
      <input type="submit" name="Logar" /><br/>
    </form>
  </body>
</html>
```

## LoginServlet: doPost()/processRequest()

```
String login = request.getParameter("login");
String senha = request.getParameter("senha");

if (loginValido(login, senha)) { // Banco de Dados!!!
    HttpSession session = request.getSession();
    session.setAttribute("logado", login);

    // Tela logado com sucesso
}
else {

    // Tela erro login
}
```

Prof. Dr. Razer A N R Montaña

JAVA WEB

35

## LoginServlet: doPost()/processRequest()

```
String login = request.getParameter("login");
String senha = request.getParameter("senha");
response.setContentType("text/html; charset=UTF-8");
if (login.equals(senha)) {
    HttpSession session = request.getSession();
    session.setAttribute("logado", login);

    try (PrintWriter out = response.getWriter()) {
        out.println("<html><head><title>Login</title></head><body>");
        out.println("<h1>Meu Portal - usuário " + login + " logado com sucesso</h1>");
        out.println("<a href=\"<code>OutraServlet</code>\">OutraServlet</a>");
        out.println("<a href=\"<code>pagina.jsp</code>\">Outra página</a>");
        out.println("</body></html>");
    }
}
else {
    try (PrintWriter out = response.getWriter()) {
        out.println("<html><head><title>Login</title></head><body>");
        out.println("<h1 style=\"color:red\">Usuário/Senha inválidos</h1>");
        out.println("<a href=\"<code>index.html</code>\">Login</a>");
        out.println("</body></html>");
    }
}
```

Prof. Dr. Razer A N R Montaña

JAVA WEB

36

## OutraServlet: doPost()/processRequest()

```
HttpSession session = request.getSession();
String logado = (String)session.getAttribute("logado");
response.setContentType("text/html;charset=UTF-8");
if (logado == null) {
    try (PrintWriter out = response.getWriter()) {
        out.println("<html><head><title>Login</title></head><body>");
        out.println("<h1 style='color:red'>Precisa estar logado</h1>");
        out.println("<a href='index.html'>Login</a>");
        out.println("</body></html>");
    }
}
else {
    try (PrintWriter out = response.getWriter()) {
        out.println("<html><head><title>Login</title></head><body>");
        out.println("<h1 style='color:blue'>Tela linda e maravilhosa</h1>");
        out.println("<a href='pagina.jsp'>Outra página</a>");
        out.println("<a href='LogoutServlet'>Logout</a>");
        out.println("</body></html>");
    }
}
```

Prof. Dr. Razer A N R Montañó

JAVA WEB

37

## pagina.jsp

```
<% if (session.getAttribute("logado") == null) { %>
    <h1 style="color:red">Você não está logado</h1>
    <a href="index.html">Login</a>
<% }
else { %>
    <h1 style="color:blue">Outra página linda</h1>
    <a href="OutraServlet">OutraServlet</a>
    <a href="LogoutServlet">Logout</a>
<% } %>
```

Prof. Dr. Razer A N R Montañó

JAVA WEB

38

## LogoutServlet: doGet() /processRequest().

```
HttpSession session = request.getSession(false);  
if (session != null) {  
    session.invalidate();  
}  
response.setContentType("text/html;charset=UTF-8");  
try (PrintWriter out = response.getWriter()) {  
    out.println("<html><head>");  
    out.println("<title>Servlet LogoutServlet</title></head>");  
    out.println("<body><h1>Deslogado com sucesso</h1>");  
    out.println("<a href=\"index.html\">Login</a>");  
    out.println("</body></html>");  
}
```



## Exercícios..

1. Criar e executar a aplicação anterior
2. SE JÁ SOUBER JDBC: Amplie a aplicação anterior para, somente armazenar a informação nome na sessão (indicando usuário logado) se o par usuário/senha estiverem cadastrados em uma tabela em um banco de dados
3. SE JÁ SOUBER JDBC: Amplie a aplicação anterior para criptografar a senha no banco de dados usando MD5