

Lab 5 - Formulários e Cookies

Neste laboratório iremos retomar o Projeto **Banco-WebView**, realizaremos interações entre a camada de visão e Servlets criados. Trabalharemos com manipulação de sessão e cookies.

Exercícios

Exercício 1: Criando o **Servlet** e **Altere o Formulário** que recebe somente dados inseridos.

Exercício 2: Manipulando **Cookies** em um Servlet.

Exercício 1 - Criando Servlet e Alterando o formulário que recebe os dados

1. Crie uma nova Classe java com o nome **ManterCliente**, realize as alterações para que fique conforme a especificação abaixo.

```
public class ManterCliente extends HttpServlet{

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

}
```

2. Abra o arquivo **web.xml** e adicione as configurações do servlet.

```
<servlet>
  <servlet-name>ManterCliente</servlet-name>
  <servlet-class>threeway.projeto.servlets.ManterCliente</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>ManterCliente</servlet-name>
  <url-pattern>/ManterCliente</url-pattern>
</servlet-mapping>
```

3. Altere o método **doGet()** para que fique igual a especificação abaixo.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub

    String nome = request.getParameter("nome");
    String telefone = request.getParameter("telefone");
    String endereco = request.getParameter("endereco");
    String rg = request.getParameter("rg");

    System.out.println("----- Saída de dados recebidos do Formulário -----");
    System.out.println("Nome: " + nome);
    System.out.println("Telefone: " + telefone);
    System.out.println("Endereco: " + endereco);
    System.out.println("Rg: " + rg);
    response.sendRedirect("manterCliente.jsp");
}
```

getParameter(): recupera os inputs mapeados no formulário através do atributo *name*, método do objeto request.
sendRedirect(): redireciona a requisição para uma página específica, método presente no objeto response.

4. Altere o método doPost() para que fique conforme abaixo.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    // TODO Auto-generated method stub  
    doGet(request, response);  
}
```

Qual a ideia de implementarmos o doGet() e referencia-lo no doPost()?

Como vamos enviar o formulário por post, o método chamado será o doPost(), assim os dados serão mandados no corpo da página e não por URL. Porém, se for digitado na URL o nome e o valor, terá o mesmo efeito pois o doGet() foi implementado.

5. Abrir o arquivo **manterCliente.jsp**, altere o código <HTML> seguindo as marcações em amarelo para que fique igual ao formato abaixo.

```
<form class="form-horizontal" action="ManterCliente" method="post" role="form">  
  <div class="form-group">  
    <label class="col-sm-2 control-label">Nome</label>  
    <div class="col-sm-4">  
      <input type="text" name="nome" class="form-control" placeholder="Nome">  
    </div>  
    <label class="col-sm-2 control-label">Telefone</label>  
    <div class="col-sm-4">  
      <input type="text" name="telefone" class="form-control" placeholder="Telefone">  
    </div>  
  </div>  
  <div class="form-group">  
    <label class="col-sm-2 control-label">Endereço</label>  
    <div class="col-sm-10">  
      <input type="text" name="endereco" class="form-control" placeholder="Endereço">  
    </div>  
  </div>  
  <div class="form-group">  
    <label class="col-sm-2 control-label">Registro Geral</label>  
    <div class="col-sm-10">  
      <input type="text" name="rg" class="form-control" placeholder="RG">  
    </div>  
  </div>  
  <div class="row">  
    <p class="text-right" style="padding-right: 5%;">  
      <button type="reset" class="btn btn-primary">Limpar</button>  
      <button type="submit" class="btn btn-primary">Salvar</button>  
      <button type="button" disabled="disabled" class="btn btn-primary">Atualizar</button>  
      <button type="button" disabled="disabled" class="btn btn-primary">Excluir</button>  
    </p>  
  </div>  
</form>
```

action: atributo do componente <form>, deve ser mapeado o nome do servlet que tratará as requisições do formulário.

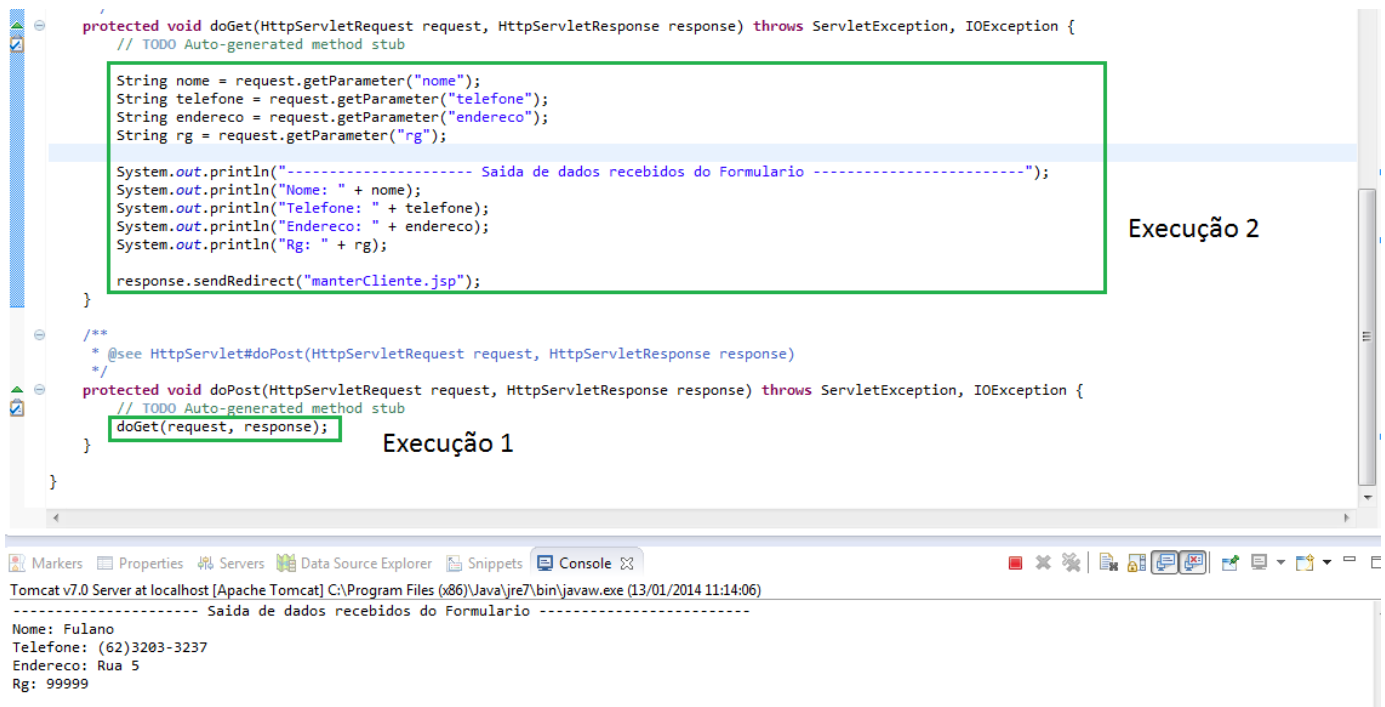
method: atributo do componente <form>, pode ser mapeado como get/post. Representa como os dados serão enviados para o servidor.

name: atributo do componente <input>, possibilita a recuperação do valor do componente pelo servlet.

type: Reset (limpa todos os dados do formulário) ou Submit (submete-dados do formulário).

6. Agora envie os dados pressionando o botão salvar.

Obs: a saída será conforme a imagem abaixo.



```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub

    String nome = request.getParameter("nome");
    String telefone = request.getParameter("telefone");
    String endereco = request.getParameter("endereco");
    String rg = request.getParameter("rg");

    System.out.println("----- Saída de dados recebidos do Formulario -----");
    System.out.println("Nome: " + nome);
    System.out.println("Telefone: " + telefone);
    System.out.println("Endereco: " + endereco);
    System.out.println("Rg: " + rg);

    response.sendRedirect("manterCliente.jsp");
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}
```

Execução 2

Execução 1

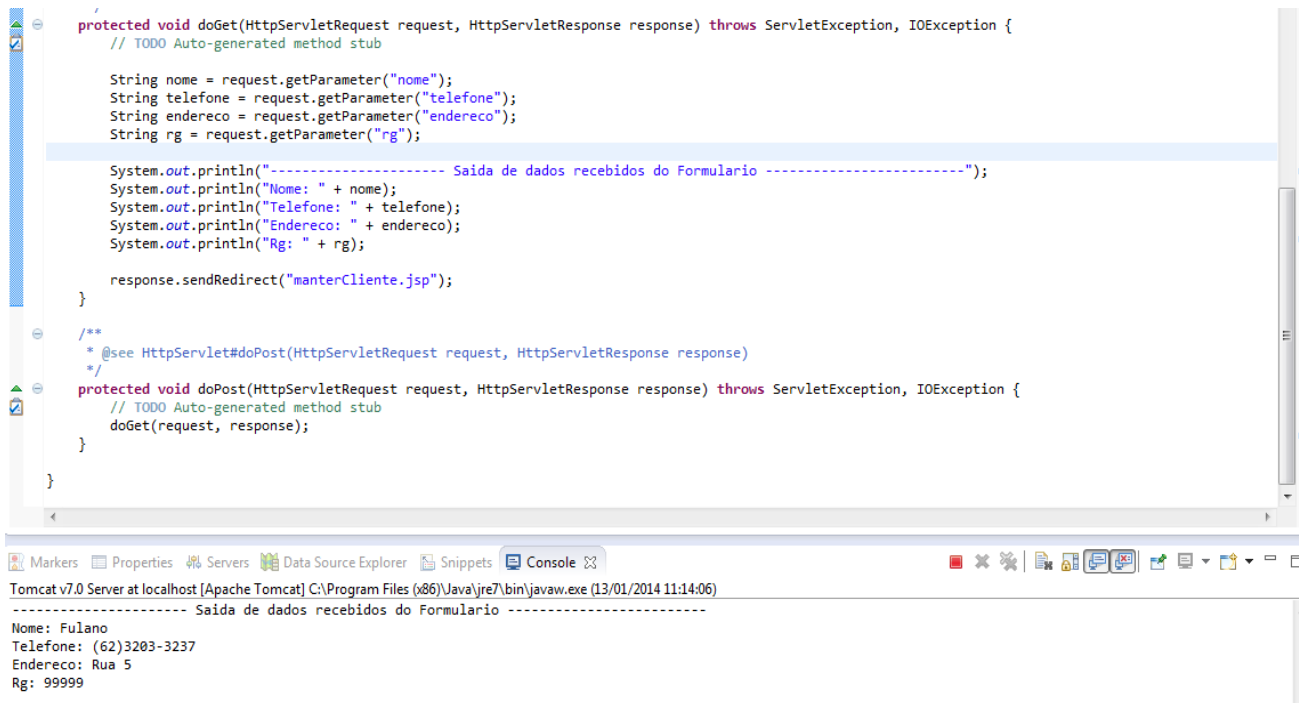
Tomcat v7.0 Server at localhost [Apache Tomcat] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (13/01/2014 11:14:06)

----- Saída de dados recebidos do Formulario -----

Nome: Fulano
 Telefone: (62)3203-3237
 Endereco: Rua 5
 Rg: 99999

7. Exemplo de envio pela URL, nesse caso o método chamado será apenas o **doGet()**, copie o código selecionado abaixo e cole na URL do navegador, a saída deve ficar igual a imagem abaixo.

- [http://localhost:8080/Banco-WebView/ManterCliente?nome=Fulano&telefone=\(62\)3203-3237&endereco=Rua+5&rg=99999](http://localhost:8080/Banco-WebView/ManterCliente?nome=Fulano&telefone=(62)3203-3237&endereco=Rua+5&rg=99999)



```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub

    String nome = request.getParameter("nome");
    String telefone = request.getParameter("telefone");
    String endereco = request.getParameter("endereco");
    String rg = request.getParameter("rg");

    System.out.println("----- Saída de dados recebidos do Formulario -----");
    System.out.println("Nome: " + nome);
    System.out.println("Telefone: " + telefone);
    System.out.println("Endereco: " + endereco);
    System.out.println("Rg: " + rg);

    response.sendRedirect("manterCliente.jsp");
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}
```

Tomcat v7.0 Server at localhost [Apache Tomcat] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (13/01/2014 11:14:06)

----- Saída de dados recebidos do Formulario -----

Nome: Fulano
 Telefone: (62)3203-3237
 Endereco: Rua 5
 Rg: 99999

Exercício 2 - Manipulando Cookies em um Servlet

1. Crie um novo **Servlet** com o nome **Login** conforme o passo um do Exercício um.
2. Realize o mapeamento no **web.xml** conforme foi feito no passo dois do Exercício um.
3. Realize o Override dos métodos doGet() e doPost(), trazendo os para a classe Login.
4. Altere o método doGet() de doPost() para que fique conforme a especificação abaixo.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

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

    final String cookieName = "usuario";
    final String cookieValue = login;

    //criando o objeto cookie
    Cookie cookieUser = new Cookie(cookieName, cookieValue);
    //vamos criar um cookie da senha apenas como exemplo para termos mais de um cookie
    Cookie cookiePsw = new Cookie("senha", senha);

    //duração de um dia
    cookieUser.setMaxAge(60*60*24);
    cookiePsw.setMaxAge(60*60*24);

    //adicionando o cookie ao navegador
    response.addCookie(cookieUser);
    response.addCookie(cookiePsw);

    //Esta funcionalidade será ensinada nas próximas aulas
    response.sendRedirect("index.jsp");

}

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

No construtor da classe Cookie é especificado o nome e valor do cookie que será criado no navegador, por enquanto não esta sendo validado login e senha apenas recuperando os dados do formulário de login e criando o cookie. Fique atento aos comentários no código.

5. No diretório WebContent, crie o arquivo **login.jsp** com o código abaixo.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="Tela de login">
<meta name="author" content="Ezequiel">
<link rel="shortcut icon" href="../../docs-assets/ico/favicon.png">

<title>:: Projeto Banco 3Way NetWorks ::</title>

<link href="resources/css/bootstrap.css" rel="stylesheet">
```

```
<link href="resources/css/geral.css" rel="stylesheet">
<script src="https://code.jquery.com/jquery-1.10.2.min.js"></script>
<script src="resources/js/bootstrap.min.js"></script>

<style type="text/css">

.form-signin {
max-width: 330px;
padding: 15px;
margin: 0 auto;
}

.form-signin .form-signin-heading,.form-signin .checkbox {
margin-bottom: 10px;
}

.form-signin .checkbox {
font-weight: normal;
}

.form-signin .form-control {
position: relative;
font-size: 16px;
height: auto;
padding: 10px;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
}

.form-signin .form-control:focus {
z-index: 2;
}

.form-signin input[type="text"] {
margin-bottom: -1px;
border-bottom-left-radius: 0;
border-bottom-right-radius: 0;
}

.form-signin input[type="password"] {
margin-bottom: 10px;
border-top-left-radius: 0;
border-top-right-radius: 0;
}

.alert{
width:350px;
margin-left: auto;
margin-right: auto;
}
</style>
</head>

<body>

<div class="container">

<div >
<div class="col-md-4 col-md-offset-4" style="margin-top: 20%;">
<div class="panel panel-default">
<div class="panel-heading">
<span class="glyphicon glyphicon-lock"></span> Login</div>
<div class="panel-body">
<form class="form-horizontal" method="post" action="Login" role="form">
<div class="form-group">
<label for="inputEmail3" class="col-sm-3 control-label">
Login:</label>
<div class="col-sm-9">
<input type="text" class="form-control" name="login" placeholder="login" required="required"
autofocus="autofocus">
</div>
</div>
</div>
<div class="form-group">
```

```
<label for="inputPassword3" class="col-sm-3 control-label">
  Password:</label>
<div class="col-sm-9">
  <input type="password" class="form-control" name="senha" placeholder="senha"
    required="required">
</div>
</div>

<div class="form-group last">
  <div class="col-sm-offset-3 col-sm-9">
    <button type="submit" class="btn btn-success btn-sm">
      Sign in</button>
    <button type="reset" class="btn btn-default btn-sm">
      Reset</button>
  </div>
</div>
</form>
</div>
<div class="panel-footer" style="text-align:center;font-size: 12px;">
  Banco S.A - Fone:(99) 999-9999
</div>
</div>
</div>
</div>
</div>

</body>
</html>
```

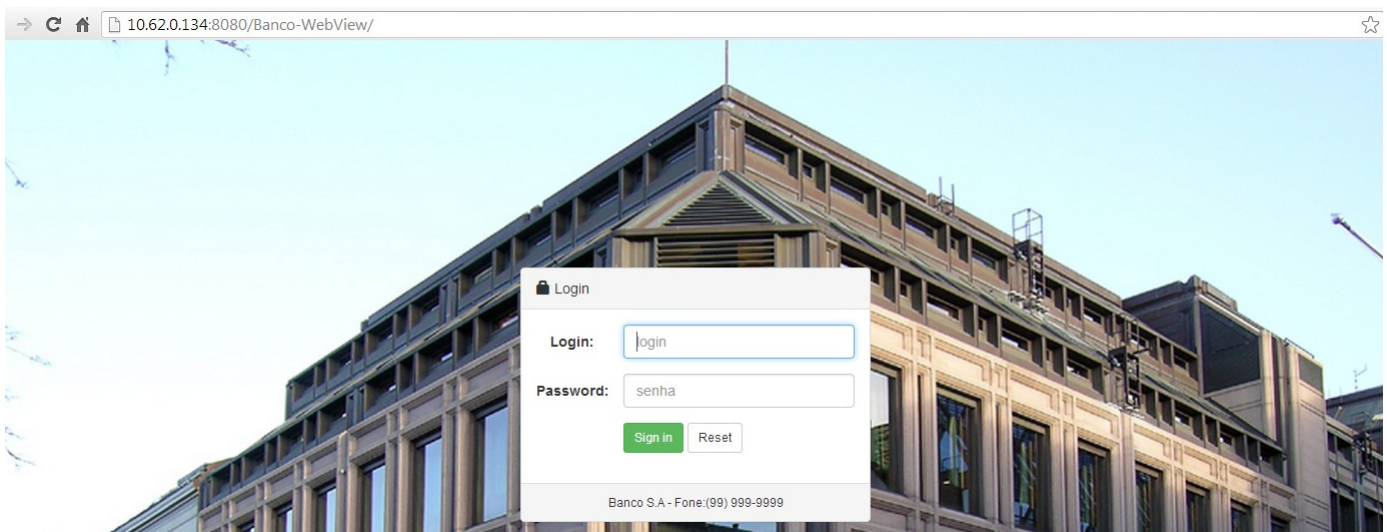
6. Abra o arquivo web.xml e altere a **tag** `<welcome-file-list>` deixando conforme abaixo.

```
<display-name>Banco-WebView</display-name>
<welcome-file-list>
  <welcome-file>login.jsp</welcome-file>
</welcome-file-list>
```

7. Agora adicione o projeto ao container apache tomcat e inicie o serviço (google chrome ou firefox).

8. Usando o navegador acesse o projeto:

- <http://localhost:8080/Banco-WebView/>
- Realize o login, por enquanto pode digitar qualquer coisa, sugestão (**login:**'admin', **senha:**'admin').

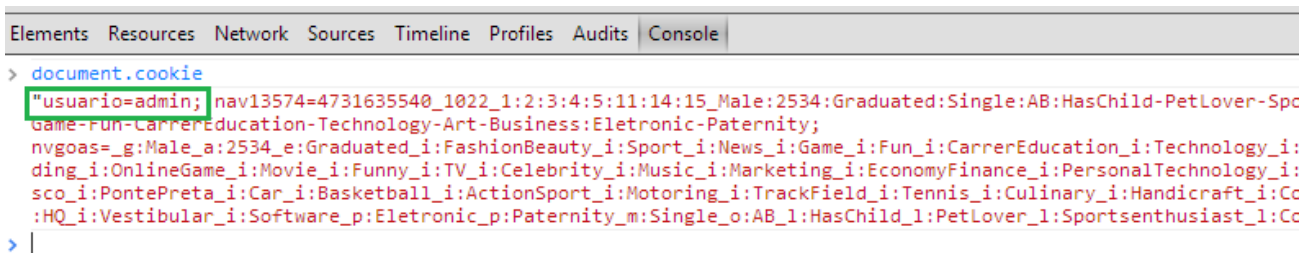


- Pressione a tecla F12, e selecione a aba console da "ferramenta para desenvolvedor" conforme a imagem abaixo, digite **document.cookie** e pressione enter.

9. Repare que ainda não adicionamos a imagem de fundo como visto acima. Crie um arquivo chamado **geral.css** na pasta `resources/css` e adicione o código abaixo. Solicite a imagem **banco.png** ao instrutor e a coloque na pasta `resources/img`.

```
body {
background: url("../img/banco.jpg") fixed;
background-size: cover;
width:100%;
height:100%;
background-repeat:no-repeat;
}

.ds-btn li{ list-style:none; float:left; padding:10px; }
.ds-btn li a span{padding-left:15px;padding-right:5px;width:100%;display:inline-block; text-align:left;}
.ds-btn li a span small{width:100%; display:inline-block; text-align:left;}
```



```
> document.cookie
"usuario=admin; nav13574=4731635540_1022_1:2:3:4:5:11:14:15_Male:2534:Graduated:Single:AB:HasChild-PetLover-SpcGame-Fun-CarrerEducation-Technology-Art-Business:Eletronic-Paternity;nvgaoas=_g:Male_a:2534_e:Graduated_i:FashionBeauty_i:Sport_i:News_i:Game_i:Fun_i:CarrerEducation_i:Technology_i:ding_i:OnlineGame_i:Movie_i:Funny_i:TV_i:Celebrity_i:Music_i:Marketing_i:EconomyFinance_i:PersonalTechnology_i:sco_i:PontePreta_i:Car_i:Basketball_i:ActionSport_i:Motoring_i:TrackField_i:Tennis_i:Culinary_i:Handicraft_i:Cc:HQ_i:Vestibular_i:Software_p:Eletronic_p:Paternity_m:Single_o:AB_l:HasChild_l:PetLover_l:Sportsenthusiast_l:Cc"
```



- Note que além do cookie criado marcado em verde, conseguimos recuperar outros cookies de outras aplicações, dessa forma aplicações maliciosas podem utilizar esses dados em cookies para outros fins, não é recomendado desenvolvermos aplicações que guardam informações ou dependem muito de cookies.
10. Vamos aplicar o uso de cookie dentro da aplicação, recuperando o cookie criado dentro do método `doGet()` do Servlet **Login** e adicionar nas paginas do sistema usando **javascript**, modifique o método **index.jsp**, adicionando o código abaixo dentro da tag **<head>** abaixo do trecho `<link href="navbar-static-top.css" rel="stylesheet">`, replique o passo para as paginas **manterCliente.jsp** e **operacoesBancarias.jsp**.

```
<script type="text/javascript">
function getCookie() {

//pega os cookies e coloca em um a array
var cookie = document.cookie;

//separa os cookies pelo ";"
var arrayCookies = cookie.split(";");

//percorre o array de cookies
for (i=0; i<arrayCookies.length; i++) {

//pega o cookie na posição atual e separa em
//um array contendo o nome do cookie e o valor
pos = arrayCookies[i].split("=");

//se a primeira posição do array criado for equivalente ao nome do cookie ("usuario")
//retornamos o valor dele, correspondente a segunda posição do array
if(pos[0] == "usuario") {
return pos[1];
}
}
return ""
```



```
}  
</script>
```

11. Localize e modifique o trecho abaixo no arquivo **index.jsp** adicionando a chamada da função **javascript** **getCookie()**.

```
<form action="Logout" id="form" method="get">
```

Bem vindo, sr(a).

```
<b>  
  <script type="text/javascript"> document.write(getCookie()); </script>  
</b>  
...
```

12. Agora nos cabeçalho das paginas terá o identificador de usuário conforme imagem abaixo.



Lembrete: Atributos de um **Cookie**.

- **getDomain/setDomain** - Permite você especificar o **domínio** o qual o cookie se aplica. O **host** deve ser parte do domínio especificado.
- **getMaxAge/setMaxAge** - Retorna/atribui o **tempo de expiração** do cookie (em segundos). Se você omitir esta atribuição, o cookie se aplica apenas para a sessão atual do browser.
- **getName** - Retorna o nome do cookie. Não existe método setName; fornecemos o nome para o construtor. Para um array de cookies, utilizamos **getName** para encontrar o cookie de interesse.
- **getPath/setPath** - Retorna/atribui o caminho no qual o cookie se aplica. Se não for especificado, o cookie se aplica a URLs que estão dentro ou abaixo do diretório contendo a página atual.
- **getSecure/setSecure** - Retorna/atribui o **flag** indica se o cookie deve ser aplicado apenas a **conexões SSL** ou para todas conexões.
- **getValue/setValue** - Retorna/atribui o valor associado com o cookie. Para novos cookies, fornecemos o valor para o construtor, não através de **setValue**. Para uma array de cookies de entrada, utilizamos **getName** para encontrar o cookie de interesse, então chamamos **getValue** no resultado. Se atribuirmos um valor ao cookie de entrada, ainda necessitamos enviá-lo de volta com **response.addCookie**.