

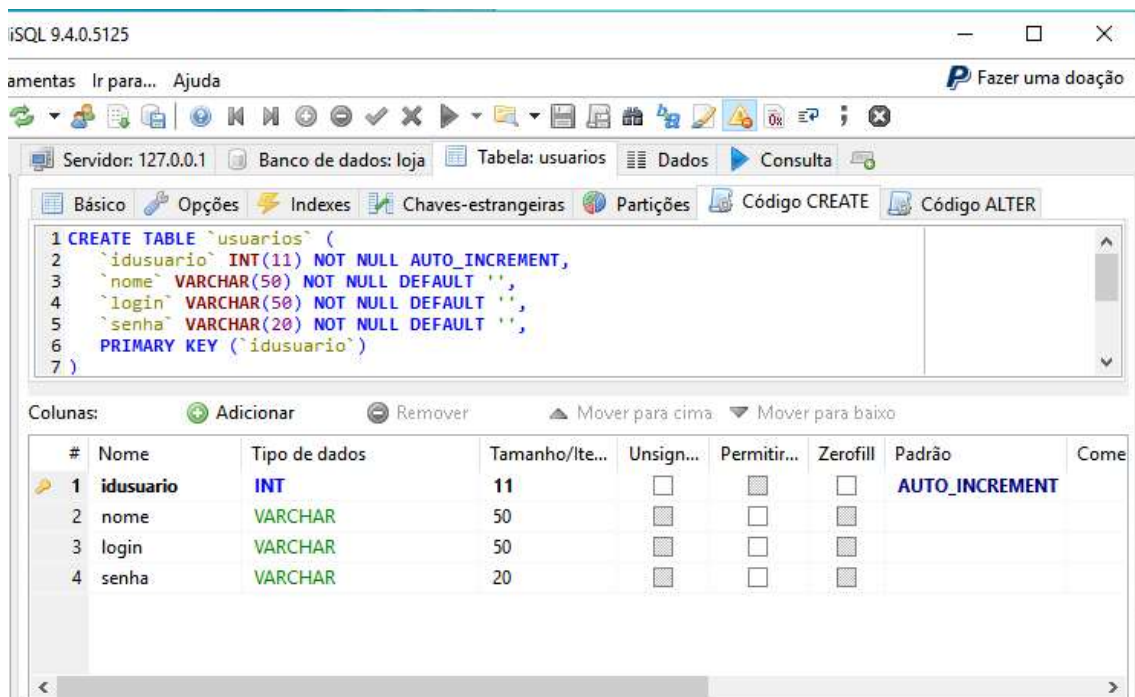
JSP Parte 2

Tópicos Abordados

- Utilização do JSP com Banco de Dados
- Trabalhar com Cookies
- Trabalhar com Sessões

Criando a Tabela de Usuários

Através do MySQL Query Browser crie a tabela de **usuários** no banco **loja**.



Instrução SQL

```
CREATE TABLE `usuarios` (  
  `idusuario` INT(11) NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(50) NOT NULL DEFAULT '',  
  `login` VARCHAR(50) NOT NULL DEFAULT '',  
  `senha` VARCHAR(20) NOT NULL DEFAULT '',  
  PRIMARY KEY (`idusuario`));
```

Inserindo alguns valores na tabela usuários

```
insert into usuarios (nome,login,senha)  
values ('Professor','prof@dai.ifmt.br','123');
```

```
insert into usuarios (nome,login,senha)  
values ('Aluno','aluno@dai.ifmt.br','123');
```

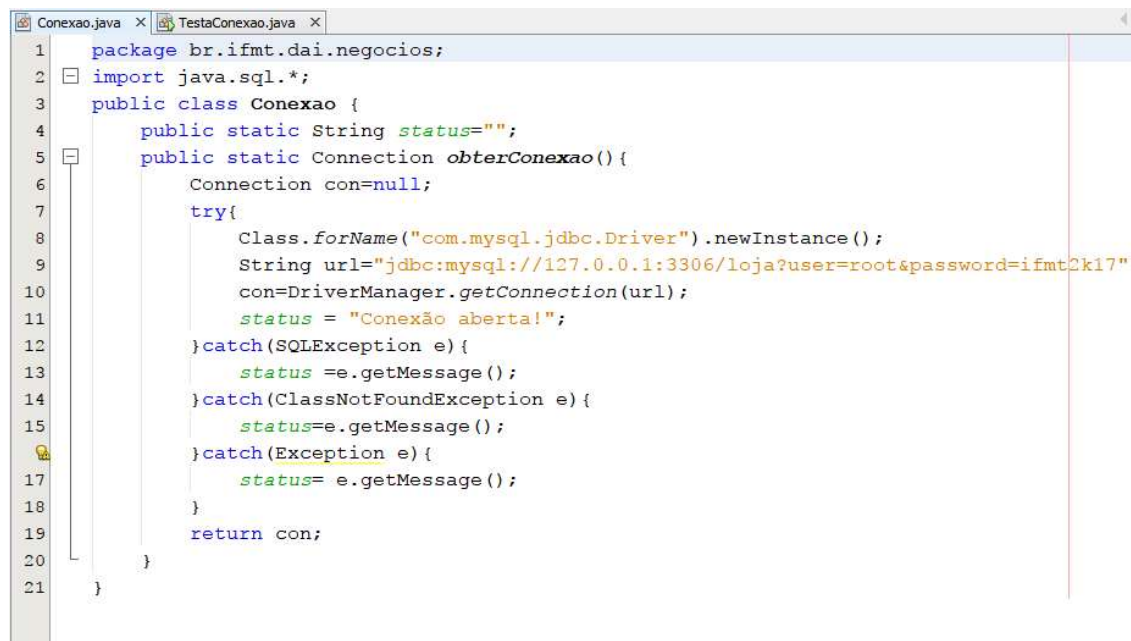
```
insert into usuarios (nome,login,senha)
values ('Monitor','monitor@dai.ifmt.br','123');
```

Verificando os valores inseridos

```
Select * from usuarios;
```

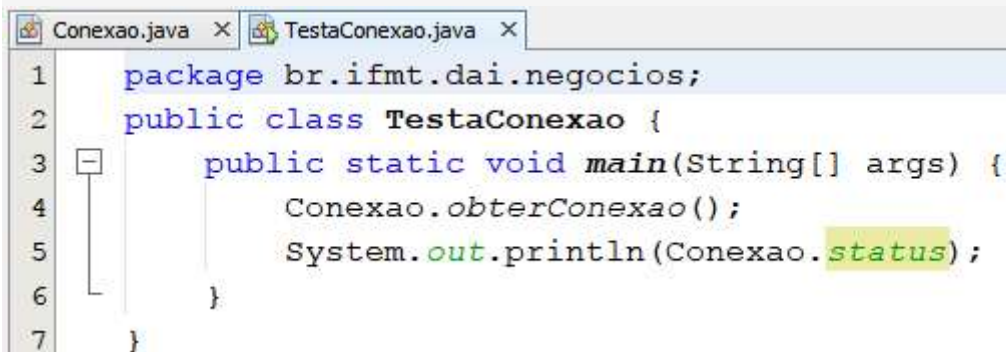
	5	Professor	prof@dai.ifmt.br	123	
	6	Aluno	aluno@dai.ifmt.br	123	
	7	Monitor	monitor@dai.ifmt.br	123	

Criando a classe Conexao



```
1 package br.ifmt.dai.negocios;
2 import java.sql.*;
3 public class Conexao {
4     public static String status="";
5     public static Connection obterConexao(){
6         Connection con=null;
7         try{
8             Class.forName("com.mysql.jdbc.Driver").newInstance();
9             String url="jdbc:mysql://127.0.0.1:3306/loja?user=root&password=ifmt2k17";
10            con=DriverManager.getConnection(url);
11            status = "Conexão aberta!";
12        }catch(SQLException e){
13            status =e.getMessage();
14        }catch(ClassNotFoundException e){
15            status=e.getMessage();
16        }catch(Exception e){
17            status= e.getMessage();
18        }
19        return con;
20    }
21 }
```

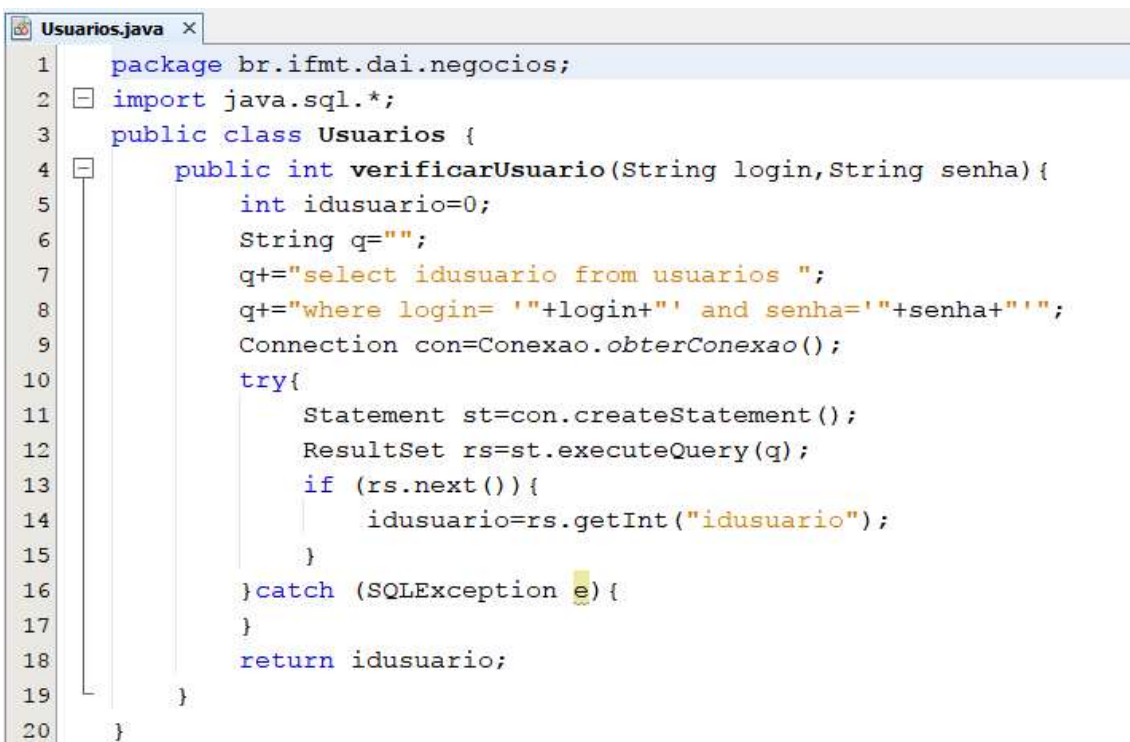
Testando a classe Conexao



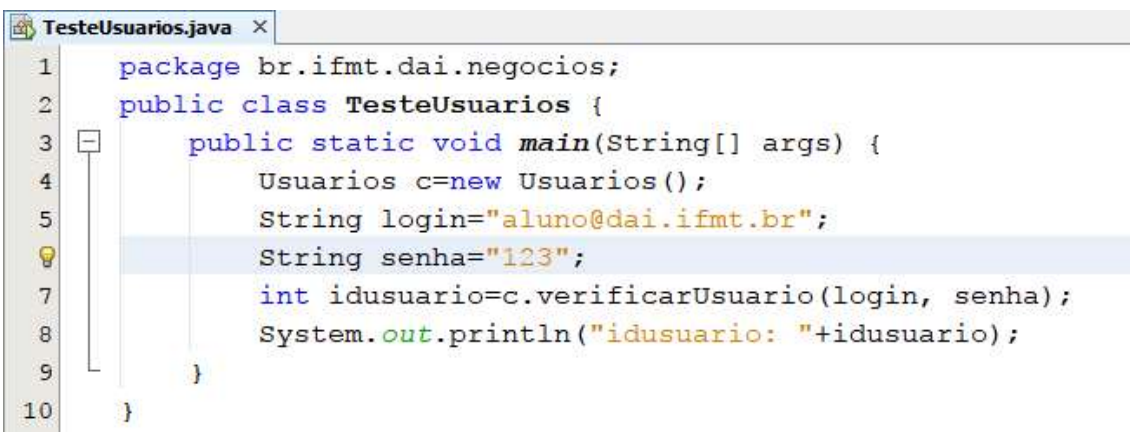
```
1 package br.ifmt.dai.negocios;
2 public class TestaConexao {
3     public static void main(String[] args) {
4         Conexao.obterConexao();
5         System.out.println(Conexao.status);
6     }
7 }
```



Criando a classe Usuarios



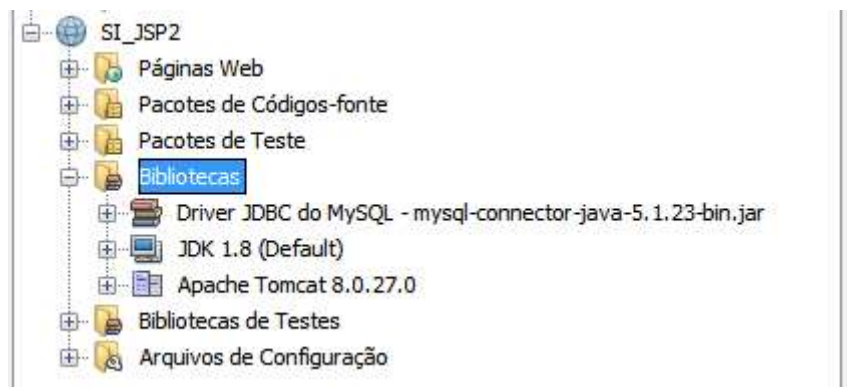
Testando a classe Usuarios





Conector Java para o MySQL

Qualquer pacote de terceiros, você deve colocar necessariamente na pasta Biblioteca. Botão direito → adicionar Biblioteca.



Criando a Tela de Login

Através desta tela o usuário entrará com o **login** e a **senha**.

Ao clicar no botão entrar será chamado o formulário **verificarlogin.jsp** (poderia ser também um Servlet)



Tela para receber os dados do Login

Esta tela receberá o login e a senha.

Através da utilização da classe Usuarios, será chamado o método verificarUsuario.

O mesmo retornará o idusuario, o qual será apresentado na tela.

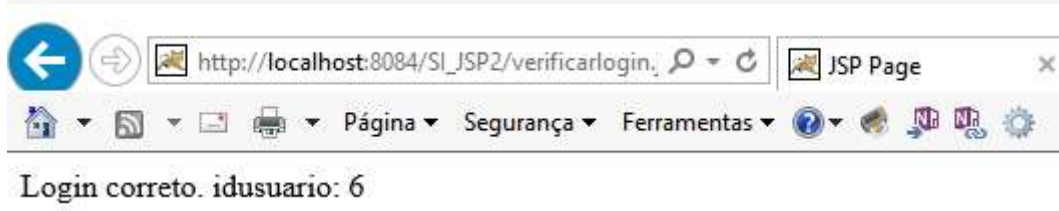
```
verificarlogin.jsp
3  <html>
4  <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
6      <title>JSP Page</title>
7  </head>
8  <body>
9      <%
10         Usuarios c= new Usuarios();
11         String login=request.getParameter("login");
12         String senha=request.getParameter("senha");
13         int idusuario=c.verificarUsuario(login, senha);
14         if(idusuario>0){
15             out.println("Login correto. idusuario: "+idusuario);
16         }else{
17             out.println("Login incorreto");
18         }
19     %>
20 </body>
21 </html>
```

Executando a página

Entre com um login e senha que você cadastrou previamente.



Resultado com a senha correta



Resultado com a senha incorreta



COOKIES com JSP

Cookie é um mecanismo padrão fornecido pelo protocolo HTTP que permite gravarmos pequenas quantidades de dados persistentes no navegador de um usuário.

Tais dados podem ser recuperados posteriormente pelo navegador.

Por padrão, os cookies expiram tão logo o usuário encerra a navegação naquele site, porém podemos configurá-los para que possam ser persistentes por vários dias.

Além dos dados que ele armazena, um *cookie* recebe um nome; um servidor pode então definir múltiplos cookies e fazer a identificação entre eles através de tais nomes.

Gerenciando Cookies

Os cookies são definidos por um servidor da web.

Quando um usuário solicita um URL cujo servidor e diretório correspondam àqueles de um ou mais de seus *cookies* armazenados, os *cookies* correspondentes são enviados de volta para o servidor.

As páginas JSP acessam os seus cookies associados através do método *getCookies()* do objeto implícito *request*.

De forma similar, as páginas JSP podem criar ou alterar *cookies* através do método *addCookie()* do objeto implícito *response*.

Esses métodos são resumidos a seguir:

request getCookies()

Retorna uma matriz de **cookies** acessíveis da página

response addCookie()

Envia um cookie para o navegador para armazenagem/modificação

Classe Cookie

Manipulamos um cookie através de instâncias da classe *javax.servlet.http.Cookie*.

Essa classe fornece apenas um tipo de construtor que recebe dois objetos do tipo String, que representam o nome e o valor do cookie:

```
Cookie cookie = new Cookie("nome", "valor");
```

Depois de construir uma nova instância, ou modificar uma instância recuperada através do método `getCookies()`, é necessário usar o método `addCookie()` do objeto `response`, com a finalidade de salvar no navegador do usuário as alterações feitas no cookie.

Para apagar um cookie utilizamos o método `setMaxAge(0)` (observe que passamos 0 como parâmetro) e depois gravamos este cookie com o método `addCookie()`.

Isso faz com que o cookie seja gravado e expirado imediatamente.

Gravando Cookie

Veja o exemplo abaixo.

Se o login estiver correto, estaremos gravando um Cookie, com o identificador login, valido por uma hora.

```
verificarlogin.jsp
1  <%@page import="br.ifmt.dai.negocios.Usuarios" contentType="text/html"
2  <!DOCTYPE html>
3  <html>
4  <...4 linhas />
5  <body>
6  <%
7      Usuarios c= new Usuarios();
8      String login=request.getParameter("login");
9      String senha=request.getParameter("senha");
10     int idusuario=c.verificarUsuario(login, senha);
11     if(idusuario>0){
12         out.println("Login correto. idusuario: "+idusuario);
13         // Gravando Cookies
14         Cookie c1=new Cookie("login",login);
15         c1.setMaxAge(60*60); // 1 Hora
16         response.addCookie(c1);
17     }else{
18         out.println("Login incorreto");
19     }
20 }
21 %>
22 </body>
23 </html>
```

Para listar todos os Cookies

```
listarcookies.jsp
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4  <...4 linhas />
5  <body>
6  <%
7      String tabela = "";
8      tabela += "<table>";
9      tabela += "<tr><td>nome</td><td>valor</td></tr>";
10     //Obtendo todos os Cookies
11     Cookie[] lista = request.getCookies();
12     for (int i=0;i<lista.length;i++){
13         String nome=lista[i].getName();
14         String valor = lista[i].getValue();
15         tabela+="<tr><td>"+nome+"</td><td>"+valor+"</td></tr>";
16     }
17     tabela+="</table>";
18     out.print(tabela);
19 }
20 %>
21 </body>
22 </html>
```

Resultado



SESSÕES COM JSP

Os atributos de uma sessão são mantidos em um objeto [HttpSession](#) referenciado pela variável `session`.

Pode-se armazenar valores em uma sessão por meio do método [setAttribute\(\)](#) e recuperá-los por meio do método [getAttribute\(\)](#).

O tempo de duração *default* de uma sessão inativa (sem o recebimento de requisições do usuário) é 30 minutos, mas esse valor pode ser alterado por meio do método [setMaxInactiveInterval\(\)](#).

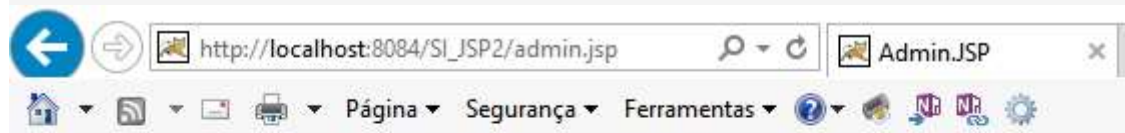
Inserindo um valor na sessão

```
verificarlogin.jsp x
3  <html>
4  <...4 linhas />
8  <body>
9  <%
10     Usuarios c= new Usuarios();
11     String login=request.getParameter("login");
12     String senha=request.getParameter("senha");
13     int idusuario=c.verificarUsuario(login, senha);
14     if(idusuario>0){
15         out.println("Login correto. idusuario: "+idusuario);
16         // Gravando Cookies
17         Cookie c1=new Cookie("login",login);
18         c1.setMaxAge(60*60); // 1 Hora
19         response.addCookie(c1);
20         // Gravando na Sessão
21         session.setAttribute("idusuario", Integer.toString(idusuario));
22         //Redirecionando para a página administrativa.
23         response.sendRedirect("admin.jsp");
24     }else{
25         out.println("Login incorreto");
26     }
27 %>
28 </body>
```


Lendo o valor da sessão

```
admin.jsp
1  <%@page import="br.ifmt.dai.negocios.Usuarios" contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4      <...4 linhas />
5  <body>
6      <h1>Admin.JSP</h1>
7      <%
8          if(session.getAttribute("idusuario")!=null){
9              int idusuario=
10                 Integer.parseInt(session.getAttribute("idusuario").toString());
11                 out.println("idusuario: "+idusuario);
12             }else{
13                 response.sendRedirect("login.jsp");
14             }
15         %>
16     </body>
17 </html>
```

Resultado



Admin.JSP

idusuario: 6

Avaliação

1- Baseado nas telas **login.jsp** e **verificarlogin.jsp**

Fazer as seguintes alterações

Exercício 1

Se houver Cookie com o login do usuario, na tela de **login.jsp** ler o Cookie e mostrar o valor na caixa de texto.

Ou seja quando o usuário for utilizar esta tela o seu login já irá aparecer.

Na tela acima o login veio de um Cookie

Exercício 2

Na tela **Admin.jsp** Mostrar os dados do usuário.

Os dados deverão ser obtidos através de um método, criado na classe Usuarios, o qual baseado no idusuario fará a busca no Banco e retornará através de metodos de acesso o nome e o email do usuario.

```
int idusuario = Integer.parseInt(session.getAttribute("idusuario").toString());
Usuarios c = new Usuarios();
c.setIdusuario(idusuario);
c.buscar();
nome = c.getNome();
email = c.getEmail();
```

Estes dados deverão ser apresentados na tela.



Admin.JSP

Nome : Aluno

Email: aluno@dai.ifmt.br

Exercício 3

Criar uma tela (**listarusuarios.jsp**) para listar todos os usuarios cadastrados.

Exercício 4

Criar uma tela (**incluirusuarios.jsp**) para incluir novos usuarios

Os dados do formulario serão enviados para uma tela chamada **salvarusuarios.jsp**.

A inclusão deverá ser feita através de um método, criado na classe Usuarios.

Observação

Validar o login e a senha (não pode ser repetido)