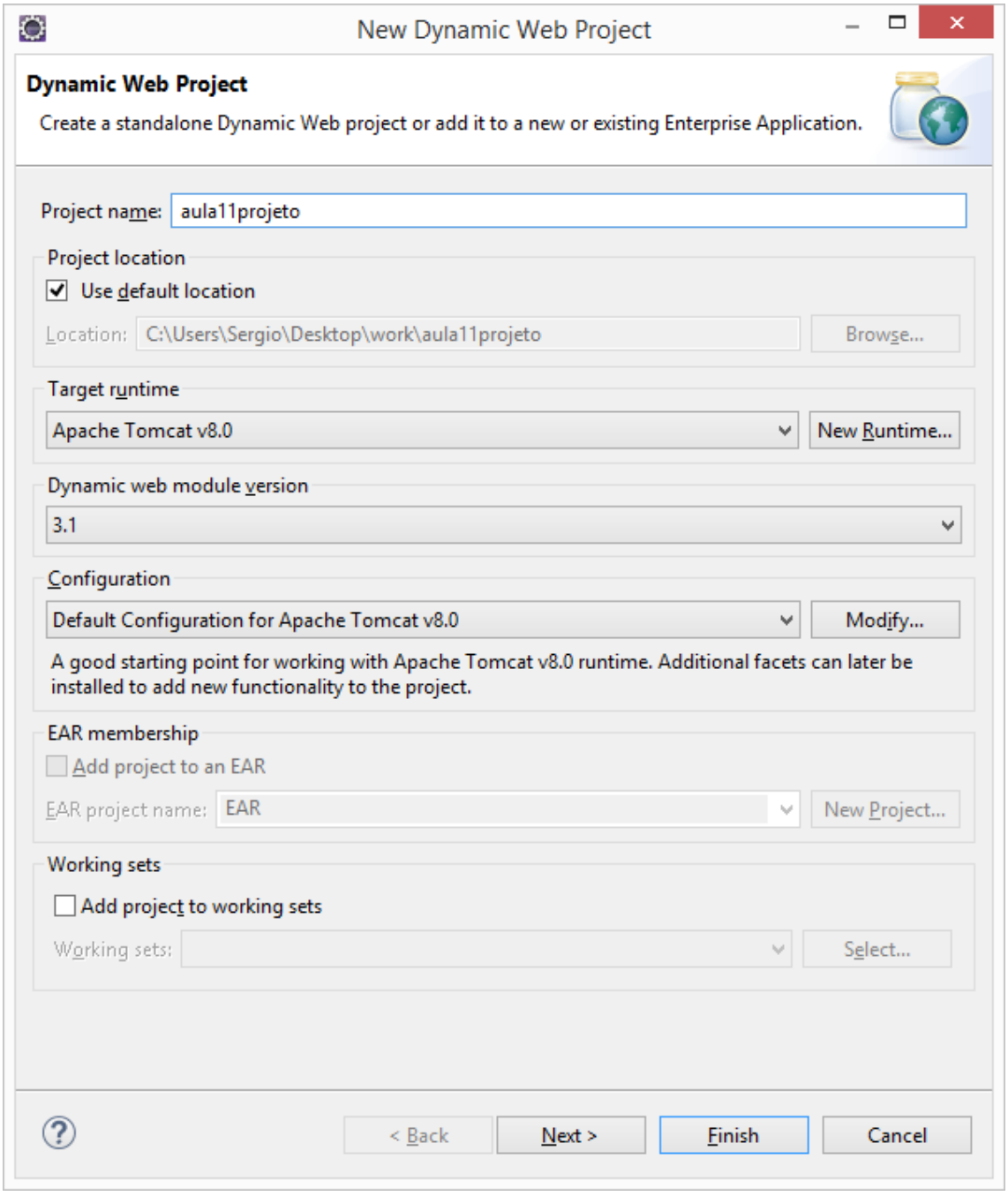
Criando o projeto

File > New > Dynamic Web Project



Criando a entidade Usuario:

```
package br.com.brq.entities;

public class Usuario {

	private Integer idUsuario;
	private String nome;
	private String login;
	private String senha;

	public Usuario() {
		// TODO Auto-generated constructor stub
	}

	public Usuario(Integer idUsuario, String nome,
			String login, String senha) {
		super();
		this.idUsuario = idUsuario;
		this.nome = nome;
		this.login = login;
		this.senha = senha;
	}

	@Override
	public String toString() {
		return "Usuario [idUsuario=" + idUsuario + ", nome="
				+ nome + ", login=" + login + ", senha=" + senha + "]";
	}

	public Integer getIdUsuario() {
		return idUsuario;
	}

	public void setIdUsuario(Integer idUsuario) {
		this.idUsuario = idUsuario;
	}

	public String getNome() {
		return nome;
	}

	public void setNome(String nome) {
		this.nome = nome;
	}

	public String getLogin() {
		return login;
	}

	public void setLogin(String login) {
		this.login = login;
	}
```

```
	public String getSenha() {
		return senha;
	}

	public void setSenha(String senha) {
		this.senha = senha;
	}
}
```

Criando a tabela na base de dados:

```
drop database if exists aula11;
create database aula11;
use aula11;

create table usuario(
	idusuario		integer			auto_increment,
	nome			varchar(50)		not null,
	login			varchar(25)		not null unique,
	senha			varchar(50)		not null,
	primary key(idusuario));

desc usuario;
```

Classe DAO

Data Access Object

```
package br.com.brq.persistence;

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class DAO {

	private static final String DRIVER = "com.mysql.jdbc.Driver";
	private static final String URL = "jdbc:mysql://localhost:3306/aula11";
	private static final String USER = "root";
	private static final String PASSWORD = "brqbrq";

	protected Connection con;
	protected PreparedStatement stmt;
	protected CallableStatement call;
	protected ResultSet rs;
```

```
        protected void openConnection() throws Exception{
                Class.forName(DRIVER);
                con = DriverManager.getConnection(URL, USER, PASSWORD);
        }

        protected void closeConnection() throws Exception{
                if(con != null){
                        con.close();
                }
        }
}
```

Criando uma classe auxiliar para realizar a criptografia da senha do usuário:

MD5 (Message Digest 5)

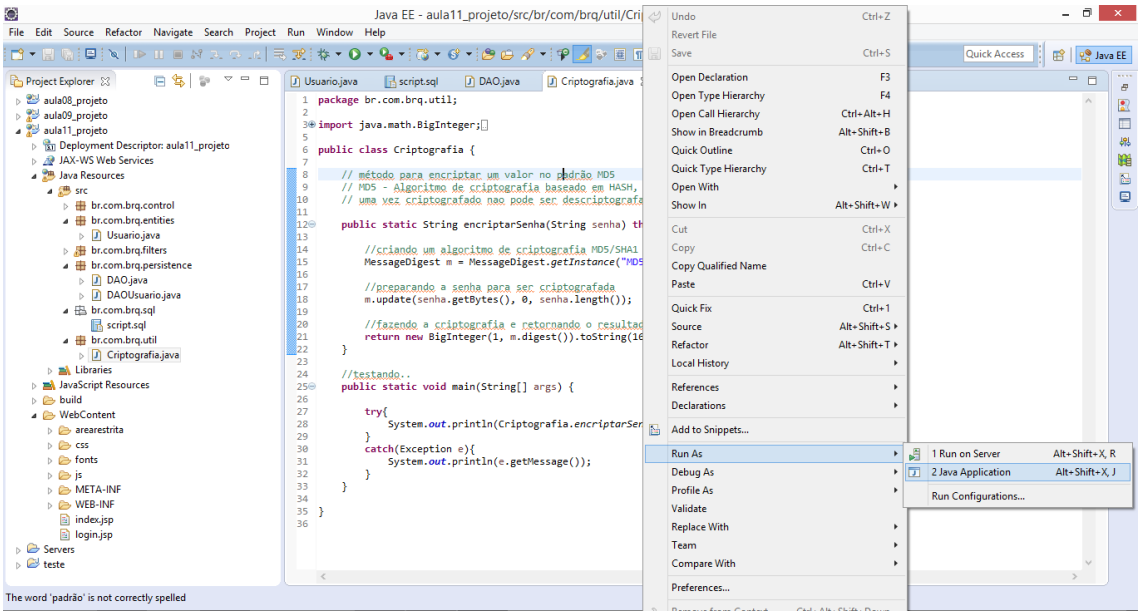
O md5 é uma **função criptográfica**, mas não deve ser confundido com **criptografia**. A encriptação é uma tarefa de mão dupla que você usa sempre que precisa armazenar com segurança uma informação, mas precisa recuperá-la mais tarde através de uma chave simétrica ou privada. Já o **hash**, é comumente utilizado quando você necessita comparar informações.

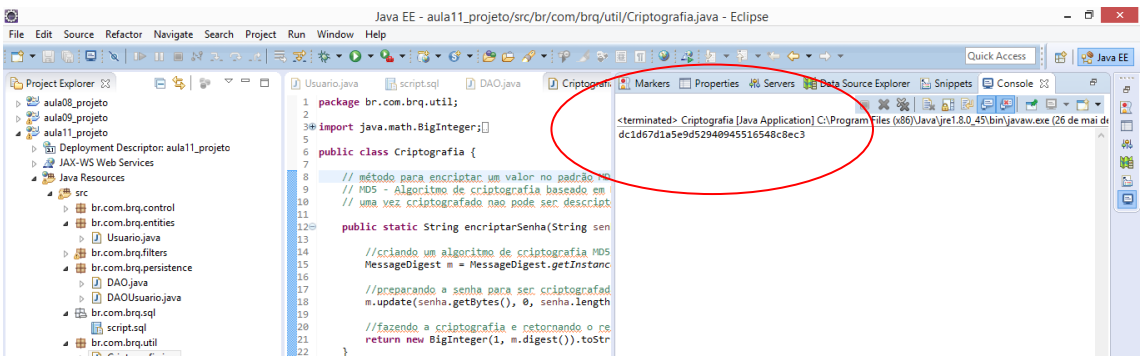
Implementando:

```
package br.com.brq.util;

import java.math.BigInteger;
import java.security.MessageDigest;

public class Criptografia {

        // método para encriptar um valor no padrão MD5
        // MD5 - Algoritmo de criptografia baseado em HASH, ou seja,
        // uma vez criptografado nao pode ser descriptografado..

        public static String encriptarSenha(String senha) throws Exception {

                //criando um algoritmo de criptografia MD5/SHA1
                MessageDigest m = MessageDigest.getInstance("MD5");

                //preparando a senha para ser criptografada
                m.update(senha.getBytes(), 0, senha.length());
```

```
                //fazendo a criptografia e retornando o resultado..
                return new BigInteger(1, m.digest()).toString(16);
        }

        //testando..
        public static void main(String[] args) {

                try{
                        System.out.println(Criptografia.encriptarSenha("Sergio"));
                }
                catch(Exception e){
                        System.out.println(e.getMessage());
                }
        }
}
```

Executando:





Criando a Classe de persistência para Usuario
/persistence/DAOUsuario.java

```
package br.com.brq.persistence;

import br.com.brq.entities.Usuario;
import br.com.brq.util.Criptografia;

public class DAOUsuario extends DAO {

	//método para inserir um usuario na base de dados..
	public void insert(Usuario u) throws Exception{

		String query = "insert into usuario(nome, login, senha) values(?, ?, ?)";

		openConnection();

		stmt = con.prepareStatement(query);
		stmt.setString(1, u.getNome());
		stmt.setString(2, u.getLogin());
		stmt.setString(3, Criptografia.encriptarSenha(u.getSenha()));
		stmt.execute();
		stmt.close();

		closeConnection();
	}


	//método para verificar se um login de usuario ja existe na base de dados..
	//true  -> login informado ja esta cadastrado na tabela de usuario..
	//false -> login informado nao esta cadastrado na tabela de usuario..
	public boolean hasLogin(String login) throws Exception{

		String query = "select count(*) as qtd from usuario "
				+ "where login = ?";

		boolean resultado = false;
		openConnection();

		stmt = con.prepareStatement(query);
		stmt.setString(1, login);
		rs = stmt.executeQuery();
```

```
            if(rs.next()){ //se resultado foi obtido..
                Integer qtd = rs.getInt("qtd");
                resultado = qtd > 0; //true ou false..
            }

            stmt.close();
            closeConnection();

            return resultado;
        }


        //método para buscar 1 usuario no banco de dados por login e senha
        public Usuario findByLoginSenha(String login, String senha) throws Exception{

            String query = "select * from usuario where login = ? and senha = ?";

            openConnection();

            stmt = con.prepareStatement(query);
            stmt.setString(1, login);
            stmt.setString(2, Criptografia.encriptarSenha(senha));
            rs = stmt.executeQuery();

            Usuario u = null; //sem espaço de memória..

            if(rs.next()){ //verificando se algum registro foi obtido..
                u = new Usuario(); //instanciando..

                u.setIdUsuario(rs.getInt("idusuario"));
                u.setNome(rs.getString("nome"));
                u.setLogin(rs.getString("login"));
                u.setSenha(rs.getString("senha"));
            }

            stmt.close();
            closeConnection();

            return u; //retornando usuario..
        }

}
```

Criando a página inicial:
index.jsp

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

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Projeto</title>

<!-- Arquivos de folha de estilo do bootstrap -->
<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css"/>
<link rel="stylesheet" type="text/css" href="css/bootstrap-theme.min.css"/>

</head>
<body class="container">

      <h2>Sistema de Controle de Usuarios</h2>
      Treinamento Java BRQ/SP - COTI Informatica
      <hr/>

      <ul>
            <li> <a href="login.jsp">Login de Usuarios</a> </li>
      </ul>

</body>
</html>
```

Resultado:
http://localhost:8082/aula11_projeto/index.jsp

Criando a página de login e cadastro de usuários:

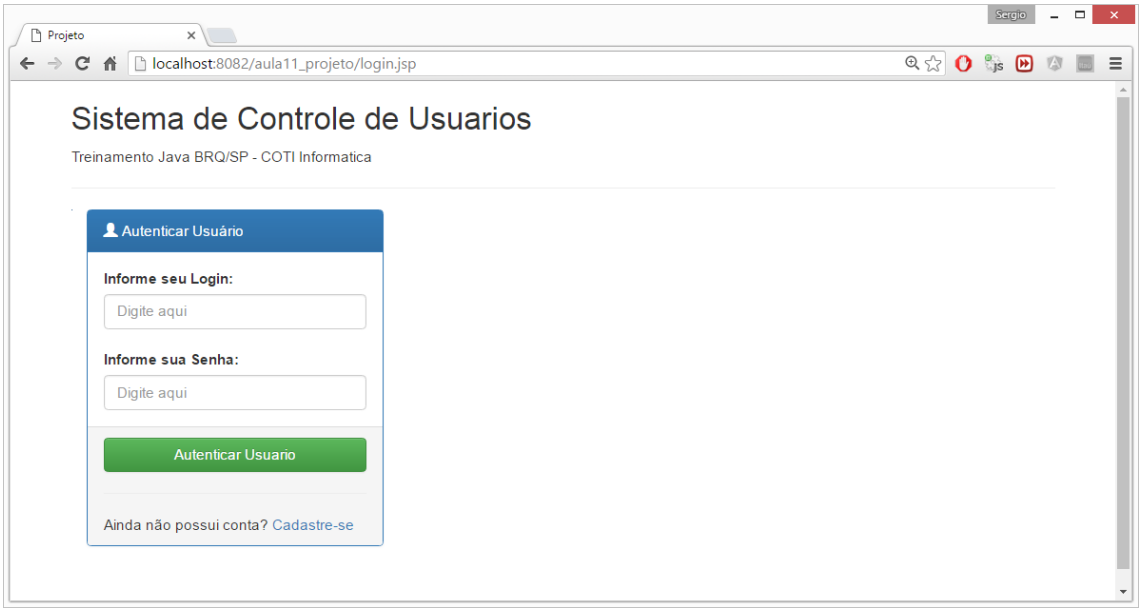
login.jsp

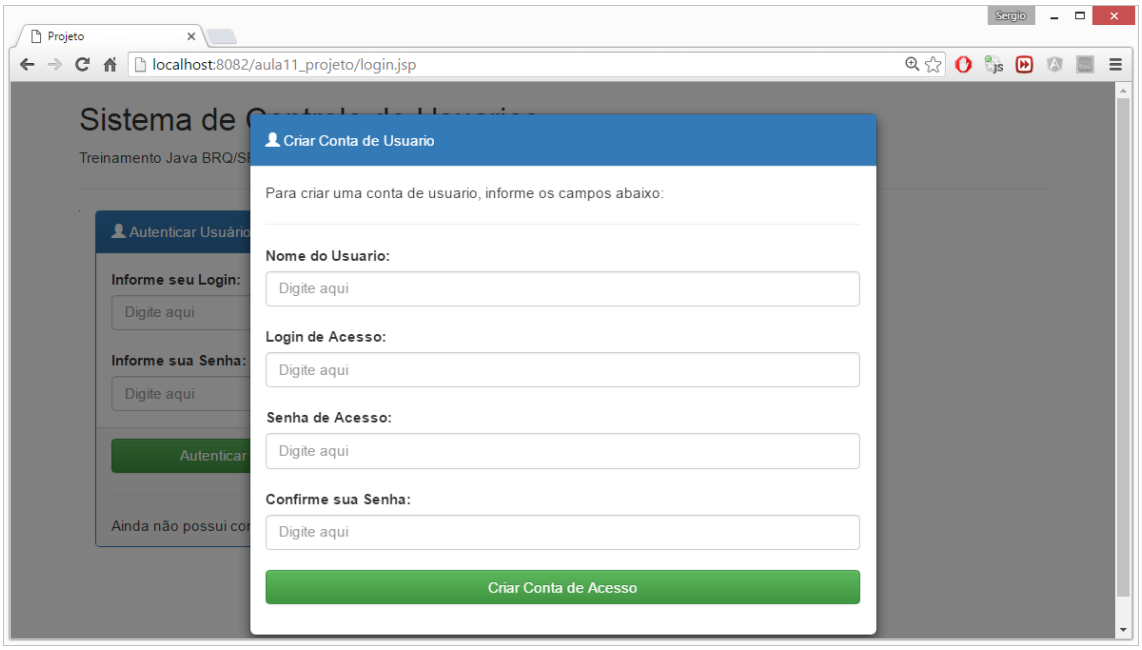
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Projeto</title>

      <!-- Arquivos de folha de estilo CSS -->
      <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css"/>
      <link rel="stylesheet" type="text/css" href="css/bootstrap-
theme.min.css"/>

      <!-- Arquivos Javascript -->
      <script type="text/javascript" src="js/jquery-1.12.4.min.js"></script>
      <script type="text/javascript" src="js/bootstrap.min.js"></script>

</head>
<body class="container">

      <h2>Sistema de Controle de Usuarios</h2>
      Treinamento Java BRQ/SP - COTI Informatica
      <hr/>

      <div class="col-md-4">

            <!-- Formulario para login de usuarios -->
            <form name="formulariologin" method="post"
                  action="ControleUsuario?cmd=autenticar">

            <div class="panel panel-primary">
                  <div class="panel-heading">
                        <span class="glyphicon glyphicon-user"></span>
                        Autenticar Usuário
                  </div>
                  <div class="panel-body">

                  <label>Informe seu Login:</label>
                  <input type="text" name="login" class="form-control"
                        placeholder="Digite aqui"/>
                  <br/>

                  <label>Informe sua Senha:</label>
                  <input type="password" name="senha" class="form-control"
                        placeholder="Digite aqui"/>
                  </div>
                  <div class="panel-footer">

                  <input type="submit" value="Autenticar Usuario"
                        class="btn btn-success btn-block"/>
                  <hr/>
```

```
        Ainda não possui conta?
        <a href="#" data-target="#janela" data-toggle="modal">
              Cadastre-se</a>

              </div>
        </div>
        </form>

        <br/><br/>
        <h4>${mensagemCadastro}</h4>

        <div class="text-danger">
              <h4>${erro}</h4>
        </div>

</div>

<!-- Janela modal para cadastro de usuarios -->
<div id="janela" class="modal fade">
      <div class="modal-dialog">
            <div class="modal-content">
            <div class="modal-header bg-primary">
                  <span class="glyphicon glyphicon-user"></span>
                        Criar Conta de Usuario
            </div>
            <div class="modal-body">

                  Para criar uma conta de usuario,
                  informe os campos abaixo:
            <hr/>

            <form name="formulariocadastro" method="post"
                    action="ControleUsuario?cmd=cadastrar">

            <label>Nome do Usuario:</label>
            <input type="text" name="nome" class="form-control"
                     value="${param.nome}"
                     placeholder="Digite aqui"/>
            <br/>

            <label>Login de Acesso:</label>
            <input type="text" name="login" class="form-control"
                     value="${param.login}"
                     placeholder="Digite aqui"/>
            <br/>

            <label>Senha de Acesso:</label>
            <input type="password" name="senha" class="form-control"
                     placeholder="Digite aqui"/>
            <br/>

            <label>Confirme sua Senha:</label>
            <input type="password" name="senhaconfirm"
                  class="form-control"
                  placeholder="Digite aqui"/>
```

```
                    <br/>

                    <input type="submit" value="Criar Conta de Acesso"
                           class="btn btn-success btn-block"/>
                </form>
            </div>
            </div>
</div>
</div>
</body>
</html>
```

Resultado:





Criando o servlet para gerenciar as operações do usuário (cadastro, autenticação e logout)

➔ **ControleUsuario.java**

```
package br.com.brq.control;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import br.com.brq.entities.Usuario;
import br.com.brq.persistence.DAOUsuario;

@WebServlet("/ControleUsuario")
public class ControleUsuario extends HttpServlet {
	private static final long serialVersionUID = 1L;

	public ControleUsuario() {
		super();
	}

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

		//resgatar a variavel que envia comandos para o servlet..
		String cmd = request.getParameter("cmd");

		if("cadastrar".equalsIgnoreCase(cmd)){
			try{
				//resgatar as senhas..
				String senha = request.getParameter("senha");
				String senhaConfirm = request.getParameter("senhaconfirm");

				if(senha.equals(senhaConfirm)){ //se as senhas estão iguais..

					//resgatar o login informado pelo usuario..
					String login = request.getParameter("login");

					DAOUsuario d = new DAOUsuario(); //persistencia..

					if( ! d.hasLogin(login)){ //se login nao existe no banco..

						Usuario u = new Usuario();
```

```
                                        //instanciando o usuario..
                                        u.setNome(request.getParameter("nome"));
                                        u.setLogin(login);
                                        u.setSenha(senha);

                                        d.insert(u); //gravando..

                                        //enviando mensagem de sucesso..
                                        request.setAttribute("mensagemCadastro",
                                                "Usuario " + u.getNome()
                                                + ", cadastrado com sucesso.");
                                }
                                else{
                                        throw new Exception("Erro: Este Login ja esta
                                                em uso. Tente novamente.");
                                }
                        }
                        else{
                                throw new Exception("Erro: Senhas não conferem.
                                        Tente novamente.");
                        }
                }
                catch(Exception e){
                        //enviar mensagem de erro para a página..
                        request.setAttribute("mensagemCadastro", e.getMessage());
                }
                finally{
                        //redirecionar de volta para a página de login..
                        request.getRequestDispatcher("login.jsp")
                                .forward(request, response);
                }
        }
        else if("autenticar".equals(cmd)){

                String destino = "login.jsp";

                try{
                        //resgatar os campos login e senha..
                        String login = request.getParameter("login");
                        String senha = request.getParameter("senha");

                        //pesquisar na base de dados o usuario pelo login e senha..
                        DAOUsuario d = new DAOUsuario();
                        Usuario u = d.findByLoginSenha(login, senha);

                        if(u != null){ //se usuario foi encontrado..

                                //criar uma sessão para armazenar
```

```
                                        o objeto do Usuario..
                                HttpSession session = request.getSession();
                                session.setAttribute("usuariologado", u);


                                destino = "arearestrita/index.jsp";
                        }
                        else{
                                throw new Exception("Acesso Negado.
                                        Tente novamente.");
                        }
                }
                catch(Exception e){
                        //exibir mensagem de erro..
                        request.setAttribute("erro", e.getMessage());
                }
                finally{
                        //redirecionar..
                        request.getRequestDispatcher(destino)
                                .forward(request, response);
                }
        }
        else if("logout".equalsIgnoreCase(cmd)){

                //remover o usuariologado da sessão
                HttpSession session = request.getSession();
                session.removeAttribute("usuariologado");
                session.invalidate(); //limpa tudo em sessão..

                //redirecionar para a página de login..
                response.sendRedirect("login.jsp");
        }
    }

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

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

Criando a página de acesso restrito:
/arearestrita/índex.jsp

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

<html>
<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Projeto</title>

<!-- Arquivos de folha de estilo do bootstrap -->
<link rel="stylesheet" type="text/css"
href="/aula11_projeto/css/bootstrap.min.css"/>

<link rel="stylesheet" type="text/css" href="/aula11_projeto/css/bootstrap-
theme.min.css"/>

</head>
<body class="container">

      <div class="well well-sm">

            <h2>Área de Acesso Restrito</h2>
            Seja bem vindo a área Administrativa do Sistema

      </div>

      <div class="col-md-12">

            <h4>Dados do Usuario Autenticado:</h4>
            <hr/>

            <label>Id do Usuario:</label> ${usuariologado.idUsuario}
            <br/>

            <label>Nome:</label> ${usuariologado.nome}
            <br/>

            <label>Login de Acesso:</label> ${usuariologado.login}
            <br/>
            <br/>

            <a href="ControleUsuario?cmd=logout"
                  class="btn btn-danger btn-sm">
                  <span class="glyphicon glyphicon-off"></span>
                  Sair do Sistema
            </a>

      </div>

</body>
</html>
```

Filters

Os filtros funcionam como intereptadores de fluxo de navegação, em outras palavras você pode evitar que o usuário consiga acessar determinado conteúdo se alguma condição não for aceita, ou você poderia criar um sistema de logs gravando o acesso do usuário a todas as páginas do sistema deixando o mesmo prosseguir com o fluxo. O filtro tem diversas utilidades e não apenas "sistemas de login".

Criando um filtro para proibir o acesso à pasta arearestrita caso não exista uma sessão identificando o usuario autenticado:

```
package br.com.brq.filters;

import java.io.IOException;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebFilter("/arearestrita/*")
public class FilterUsuario implements Filter {

	public FilterUsuario() {
	}

	public void destroy() {
	}

	//método executado quando a pasta arearestrita for acessada..
	public void doFilter(ServletRequest request, ServletResponse response,
			FilterChain chain)
			throws IOException, ServletException {

		//converter os parametros do filter para
		ficarem compativeis com os parametros
		//de um servlet (HttpServletRequest e HttpServletResponse)


		HttpServletRequest req = (HttpServletRequest) request;
		HttpServletResponse resp = (HttpServletResponse) response;
```

```
            //instanciar a session..
            HttpSession session = req.getSession();
            //verificar se existe na sessão um objeto -> usuariologado
            if(session.getAttribute("usuariologado") != null){
                chain.doFilter(request, response); //continua...
            }
            else{
                //redirecionar de volta para a página de login..
                resp.sendRedirect("/aula11_projeto/login.jsp");
            }
        }

        public void init(FilterConfig fConfig) throws ServletException {
        }
}
```

Criando um filtro para limpar o cache do navegador:

```
package br.com.brq.filters;

import java.io.IOException;
import java.util.Date;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpServletResponse;

@WebFilter(urlPatterns = { "/arearestrita/*", "/ControleUsuario" })
public class FilterCache implements Filter {

        public FilterCache() {
        }

        public void destroy() {
        }


        public void doFilter(ServletRequest request, ServletResponse response,
                        FilterChain chain)
```

```
                throws IOException, ServletException {

            HttpServletResponse resp = (HttpServletResponse) response;

            resp.setHeader("Last-Modified", new Date().toString());
            resp.setHeader("Cache-Control", "no-store, no-cache, must-revalidate, max-
                age=0, post-check=0, pre-check=0");
            resp.setHeader("Pragma", "no-cache");

            chain.doFilter(request, response);
        }

        public void init(FilterConfig fConfig) throws ServletException {
        }
}
```

Estrutura do projeto: