

Projeto de um blog em JSP	1
Introdução	1
O Banco de dados.....	1
Os Beans	1
<i>Javabeans Datas.....</i>	<i>2</i>
<i>JavaBeans Postar.....</i>	<i>2</i>
<i>JavaBeans Consultar.....</i>	<i>3</i>
Estrutura do Aplicativo	4
<i>Folha de estilos.....</i>	<i>5</i>
<i>Página de Login</i>	<i>6</i>
<i>Após logar (logged.jsp)</i>	<i>7</i>
<i>Enviar mensagens (postar.jsp).....</i>	<i>8</i>
<i>Página enviada (posted.jsp).....</i>	<i>9</i>
<i>Página principal (zlog.jsp).....</i>	<i>9</i>
As telas	9
<i>Login.....</i>	<i>10</i>
<i>Logged.....</i>	<i>10</i>
<i>Enviar.....</i>	<i>10</i>
<i>Enviado.....</i>	<i>10</i>

Projeto de um blog em JSP

Introdução

Hoje em dia, os blogs são comuns. Eu tenho a minha versão estática em minha página. Infelizmente, preciso digitar as datas e formato HTML na mão.

Usando JSP, Javabeans e banco de dados, criei um blog. Ou melhor, um Z-log. Ele deveria estar on-line. Mas o mycgiserver não me diz onde colocar meus javabeans. Aqui está implementado usando Access XP. Pela facilidade.

O Banco de dados

Crie um banco de dados chamado msg com uma tabela msg com três campos: Data (tipo texto), assunto(tipo texto) e msg (memo). O tipo de dado do campo msg muda se você utilizar o HSQL ou MySQL.

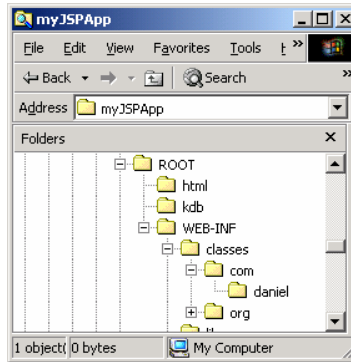
Registre esse banco no Data Sources ODBC do Windows. E está pronto.



Data Sources (ODBC).lnk

Os Beans

São precisos 3 Javabeans. Para isso, crie o package dentro do diretório apropriado com a seguinte estrutura com/daniel e inclua os javabeans.



JavaBeans Datas

Essa classe utiliza um objeto Date para retornar a hora e data. O objeto SimpleDateFormat, permite formatar a data de modo que tenhamos um formato personalizado. Note que é preciso converter a data para uma String.

```
package com.daniel;

import java.util.*;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Datas {
    private String datahj;
    public void setData(){
        Date dhj = new Date();
        SimpleDateFormat formatter =
            new SimpleDateFormat ("E yyyy.MM.dd 'at' hh:mm:ss a zzz");
        datahj = formatter.format(dhj).toString();
    }

    public String getData(){
        return datahj;
    }
}
```

JavaBeans Postar

Essa classe, envia dados para o banco de dados. Esses dados são mostrados toda vez que o usuário acessar a página principal.

```
package com.daniel;

import java.sql.*;

public class Postar{

    public boolean erro1 = false;
    public boolean erro2 = false;
    public boolean erro3 = false;

    public void setPost( String d , String a , String m ) {
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection con =
                DriverManager.getConnection("jdbc:odbc:msg","","");
            Statement s = con.createStatement();
```

```

        String query = "INSERT INTO msg " +
            " (Data , assunto, msg)" +
            " VALUES" +
            " ('" + d + "'," +
            " '" + a + "'," +
            " '" + m + "')";

        int i = s.executeUpdate(query);
        s.close();
        con.close();

    }catch (ClassNotFoundException e) {
        //
        erro1 = true;
    }catch (SQLException e) {
        //
        erro2 = true;
    }catch (Exception e) {
        //
        erro3 = true;
    }
}
} //fim de setPost()
}

```

JavaBeans Consultar

Esse bean simplesmente busca as entradas do banco e retorna os dados em três ArrayLists. É preciso retornar ArrayLists ao invés de Arrays comuns, porque sua dimensão é dinamicamente alocada.

```

package com.daniel;

import java.sql.*;
import java.util.ArrayList;

public class Consultar{

    private ArrayList datas;
    private ArrayList assunto;
    private ArrayList messages;

    protected String erro1="";
    protected String erro2="";
    protected String erro3="";

    public void setConsultar(){
        try {

            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection con =
                DriverManager.getConnection("jdbc:odbc:msg","","");
            Statement s = con.createStatement();

            String query = "SELECT * FROM msg";
            ResultSet rs = s.executeQuery(query);
            ResultSetMetaData rsmd = rs.getMetaData();
            datas = new ArrayList();
            assunto = new ArrayList();
            messages = new ArrayList();

```

```

        while ( rs.next() ) {
            datas.add(rs.getString(1));
            assunto.add(rs.getString(2));
            messages.add(rs.getString(3));
        }
        s.close();
        con.close();
        rs.close();

    }catch (ClassNotFoundException e) {
        //
        erro1 = e.toString();
    }catch (SQLException e) {
        //
        erro2 = e.toString();
    }catch (Exception e) {
        //
        erro3 = e.toString();
    }

}

public String[] getConsultar(){
    String[] ms = new String[datas.size()];
    for( int i = 0 ; i < datas.size() ; i++ ){

        ms[i] = "<div id='bloco'><h3>(Data: " +
            datas.get(i) + ")" +
            "</h3><h4>[Assunto:" + assunto.get(i) +
            "]"</h4><p>" +
            messages.get(i) + "</p></div>";

    }

    return ms;
}

public String getErrol(){
    return erro1;
}

public String getErro2(){
    return erro2;
}

public String getErro3(){
    return erro3;
}

}

```

Estrutura do Aplicativo

Duas áreas distintas:

- - página principal: É a única aberta ao usuário
- - página para envio de mensagens: É preciso logar ao sistema.

Ou:

login.html -> logged.jsp -> postar.jsp -> posted.jsp

Além disso, são precisos arquivos com os headers e footers das páginas prontos, que são enviados dinamicamente à página utilizando a diretiva include.

Folha de estilos

```
body {  
  
background-color: black;  
border: solid 5px;  
border-color: black;  
padding: 1;  
  
}  
  
p {font-family: 'Microsoft Sans Serif','Futura Lt BT','Bookman Old Style', arial, courier;  
font-size: 11pt;  
color: white;  
font-style: normal;  
text-indent: 0.5 cm;  
}  
  
#banner {font-family: 'Microsoft Sans Serif','Futura Lt BT','Bookman Old Style', arial,  
courier;  
font-size: 11pt;  
color: #ff6600;  
font-style: normal;  
text-indent: 0.5 cm;  
}  
  
#esquerda {  
  
font-family: 'Courier New','Microsoft Sans Serif','Futura Lt BT','Bookman Old Style', arial,  
courier;  
font-size: 10pt;  
color: white;  
text-align: right;  
color: #cccc99;  
}  
  
h1 {  
font-family: 'Microsoft Sans Serif','Courier New', arial, courier;  
font-size: 28pt;  
font-variant: normal;  
color:#ff9900;  
text-align: center;  
}  
  
h2 {  
font-family: 'Microsoft Sans Serif','Courier New', arial, courier;  
font-size: 20pt;  
font-style: normal;  
color:white;  
text-align: left;  
}  
  
h3 {font-family: 'Microsoft Sans Serif','Courier New', arial, courier;  
font-size: 12pt;  
font-style: normal;  
color:yellow;  
text-align: left;  
}  
  
h4 {font-family: 'Microsoft Sans Serif','Courier New', arial, courier;
```

```
font-size: 11pt;
font-style: normal;
color: #999999;
text-align: left;
}

h5 {font-family: 'Microsoft Sans Serif', 'Courier New', arial, courier;;
font-size: 10pt;
font-style: normal;
text-align: center;
color: yellow;
}

a:link {
    color: white;
    font-family: 'Courier New', Arial, Courier;
}

a:visited {
    color: white;
    font-family: 'Courier New', Arial, Courier;
}

a:hover { color: #00ffff;
    font-family: 'Courier New', Arial, Courier;
    background-color: black;
}

p.intro{
background-color: grey;
font-family: 'Courier New', Arial, Courier;
color: #ffcc00;
font-size 14pt;
}

#bloco
{
background-color: #336699;
border: solid 1px;
border-color: #999999;
padding: 2;
}

pre{
    color: white;
}
```

Página de Login

Um valor de controle é passado em um componente "hidden".

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>
<head>

<link rel="stylesheet" type="text/css" href="estilosmain.css" />

<title>Zlog</title>

<body>
```

```

<h1>Administrar Zlog</h1>

<h2>Conectar Ao Sistema</h2>
<form action="logged.jsp" method="post">
<table>
  <tr>
    <td>
      <p>
        usuário:
      </p>
    </td>
    <td>
      <input type="text" name="user" value="" />
    </td>
  </tr>
  <tr>
    <td>
      <p>
        senha:
      </p>
    </td>
    <td>
      <input type="password" name="senha" value="" />
    </td>
  </tr>
  <tr>
    <td>
      <button type="submit">Enviar</button>
    </td>
    <td>
      <input type="hidden" name="control" value="conectado" />
    </td>
  </tr>
</table>
</form>

<hr/>
<p id="esquerda">
2004 Electric Ocean All Rights Reserved. Copyright 2004 Daniel V. Gomes
</p>

</body>
</html>

```

Após logar (logged.jsp)

```

<%@ include file="html/header.html"%>

<%
String userN = request.getParameter("user");
String senhaN = request.getParameter("senha");
String controle = request.getParameter("control");

if ( controle.equals("conectado") ) {

    if ( ( userN.equals("tirimica") ) && ( senhaN.equals("thrash") ) ) {
        out.println("<h2>Bem-vindo Daniel.</h2>");
        session = request.getSession(true);
        session.setAttribute("conectado", new String("true"));
    }

}%>

```

```

        <ul>
            <li><a href="postar.jsp">Postar</a></li>
            <li><a href="apagar.jsp">Apagar</a></li>
        </ul>
    <%
    } else {
        out.println("<h3>Erro!!!</h3>");
        out.println("<h3>usuário não autorizado</h3>");
    }
    } else if ( controle.equals("null") ) {
        out.println("<h3>Erro!!!</h3>");
        out.println("<h3>usuário não autorizado</h3>");
    }
    %>

<%@ include file="html/footer.html"%>

```

Enviar mensagens (postar.jsp)

```

<%@ include file="html/header.html"%>

<%
    session = request.getSession();
    String conectado = (String)session.getAttribute("conectado");
    if (conectado.equals("true")) {
        %>

<h1>>>Postar</h1>
<form action="posted.jsp" method="post">
<table cellpadding="2">
    <tr>
        <td>
            <h2>Assunto:</h2>
        </td>
        <td>
            <input type="text" name = "assunto" size="50" />
        </td>
    </tr>
    <tr>
        <td>
            <h2>Mensagem:</h2>
        </td>
        <td>
            <textarea rows="10" cols="80" name="message"></textarea>
        </td>
    </tr>
    <tr>
        <td>
            <button type="submit">Enviar</button>
        </td>
    </tr>
</table>
</form>
<%
    } else {
        <h3>Erro</h3>
    }
    %>

<%@ include file="html/footer.html"%>

```


Página enviada (posted.jsp)

```
<%@ include file="html/header.html"%>

<jsp:useBean id="postar" class="com.daniel.Postar"/>
<jsp:useBean id="datas" class="com.daniel.Datas"/>

<%
    datas.setData();
    postar.setPost( datas.getData(), request.getParameter("assunto") ,
request.getParameter("message") );

    out.println("<p>Mensagem enviada!</p>");
%>

<a href="zlog.jsp">Ver Zlog</a>

<%@ include file="html/footer.html"%>
```

Página principal (zlog.jsp)

```
<%@ include file="html/header.html"%>
<%@ page import="java.util.ArrayList" %>

<jsp:useBean id="msgs" class="com.daniel.Consultar"/>

<h1> >>> Zlog ..... </h1>

<h2> >>> mensagens.... </h2>

<p class="intro">
Aqui está meu Z-LoG! Diário de bordo sobre tecnologia, principalmente JAVA e
eletrônica digital! Para sugestões, dicas e correções:
<a href="mailto:dansovg@ig.com.br">dansovg@ig.com.br</a>
</p>

<p class="intro">
Por favor, para maiores informações, visite minha página:
<a href="http://www.geocities.com/danielvg.geo" target="_blank">Tecnologia & Outros </a>
</p>

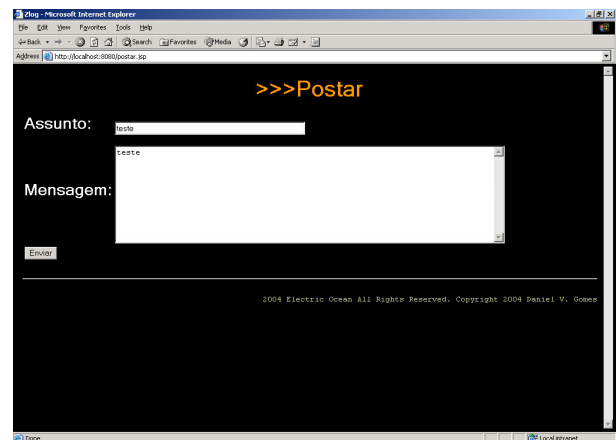
<%
    msgs.setConsultar();
    String[] m = msgs.getConsultar();
    for ( int i = 0 ; i < m.length ; i++ ){

        out.println(m[i]);
    }
%>

<%@ include file="html/footer.html"%>
```

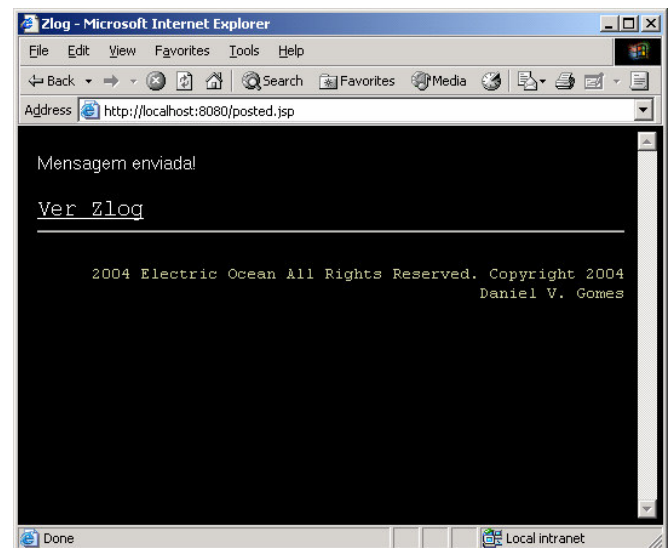
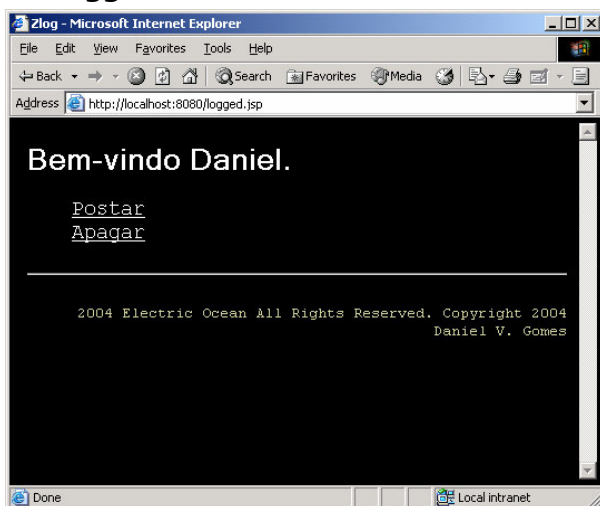
As telas

Login



Enviado

Logged



Enviar

