

# Java Web

---

AULA 02 – SERVLETS E JSP BÁSICO

## Objetivos e Conceitos

---

- Objetivos:
  - Apresentar conceitos básicos sobre Servlets e JSP. Executar uma aplicação mínima
- Conceitos:
  - Servlets, JSP, Aplicação, Deploy, Undeploy

## Tópicos

---

1. JSP Básico
2. Servlets Básico
3. Arquivo de uma Aplicação Java Web
4. Deploy/Undeploy

# 1 JSP Básico

---

## JSP Básico

Criação de páginas dinâmicas

Baseado na tecnologia Java

- Vasta gama de bibliotecas
- Independente de plataforma

Processamento no servidor (*server-side*)

Seu código é convertido em Servlet na primeira execução

Tags especiais inseridas na página HTML

- <% ... %>
- Dentro coloca-se código Java
- Pode-se ter várias tags

## JSP Básico

Comando para imprimir um HTML

```
out.println("...");
```

Exemplo

```
out.println("Oi Mundo");
out.println("Teste de String<br/>");
out.println("<h2>Oi Mundo</h2>");
```

## JSP Básico – Exemplo 1

```
<html>
    <head><title>Titulo</title>
    </head>
    <body>
        <b>Página HTML</b><br/>
        <h1><% out.println("Hello World!"); %></h1>
    </body>
</html>
```

Código Java

## JSP Básico – Exemplo 2

```
<html>
    <head><title>Titulo</title>
    </head>
    <body>
        <b>Página HTML</b><br/>
        <h1><% out.println("Hello World!"); %></h1>
        <%
            for (int i=0; i<10; i++) {
                out.println("<h2>Valor = " + i + "</h2>");
            }
        %>
    </body>
</html>
```

## JSP Básico.

Pode-se usar qualquer API do Java

Deve-se fazer as importações de pacotes necessários com:

```
<%@page import="java.util.Date" %>
<%@page import="java.time.* , java.io.*" %>
```

Exemplo 3

```
<%@page import="java.util.Date" %>
<html><head><title>Titulo</title></head>
<body>
    <b>Data/Hora Atual</b><br/>
    <h1><% out.println( new Date() ) ; %></h1>
</body>
</html>
```



## Exercícios..

1. Executar os JSPs de exemplo
  - a. Criar novo projeto no Netbeans
  - b. Adicionar um JSP para cada exemplo
  - c. Executar

# 2 Servlets Básico

## Servlets Básico

Classes Java que dão como saída um texto em HTML

São invocadas pelo Servidor (Tomcat, JBoss, Glassfish, etc)

O servidor invoca métodos específicos destas classes

Toda a saída gerada em HTML é repassado ao cliente que requisitou a página

## Servlets Básico

O servidor invoca um método da Servlet conforme a requisição HTTP:

- Se for uma requisição GET -> invoca o `doGet()`
- Se for uma requisição POST -> invoca o `doPost()`

Dentro usa-se um `PrintWriter` para gerar o HTML

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

...
out.println("<h2>minha saída em HTML</h2>");
```

## Servlets Básico – Exemplo 1.

```
1. import java.io.*;
2. import javax.servlet.*;
3. import javax.servlet.http.*;
4.
5. @WebServlet("Teste")
6. public class Teste extends HttpServlet {
7.     public void doGet(HttpServletRequest request,
8.                         HttpServletResponse response)
9.             throws ServletException, IOException {
10.
11.     PrintWriter out = response.getWriter();
12.     response.setContentType("text/html");
13.     out.println("<html><head><title>Teste</title></head>");
14.     out.println("<body>Teste de Servlet</body></html>");
15.     out.flush();
16. }
17. }
```

## Servlets Básico

O plugin do Netbeans cria um método

```
processRequest()
```

Quando ocorre uma invocação, seja via GET ou via POST, os métodos `doGet()` e `doPost()` invocam o `processRequest()`

Então, para facilitar, basta você implementar o `processRequest()`

**CUIDADO: Não apagar as implementações de `doGet()` e `doPost()` senão a Servlet NÃO FUNCIONA**

## Servlets Básico - Netbeans

```
import *;

@WebServlet(urlPatterns = {"/*TesteServlet1"})
public class TesteServlet1 extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
                                   HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet TesteServlet1</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet TesteServlet1 at " + request.getContextPath() + "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}
```

+ HttpServlet methods. Click on the + sign on the left to edit the code

Implementações de  
doGet() e  
doPost() escondidas.  
**NÃO APAGAR**

## Servlets Básico – Exemplo 2.

```
import *;  
  
@WebServlet(urlPatterns = {"/TesteServlet1"})  
public class TesteServlet1 extends HttpServlet {  
    protected void processRequest(HttpServletRequest request,  
                                  HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html; charset=UTF-8");  
        try (PrintWriter out = response.getWriter()) {  
            out.println("<!DOCTYPE html>");  
            out.println("<html>");  
            out.println("<head>");  
            out.println("<title>Teste TADS</title>");  
            out.println("</head>");  
            out.println("<body>");  
            out.println("<h1>Minha Primeira Servlet</h1>");  
            out.println("</body>");  
            out.println("</html>");  
        }  
    }  
}  
+ HttpServlet methods. Click on the + sign on the left to edit the code.  
}
```



## Exercícios..

1. Executar a primeira Servlet
  - a. Criar novo projeto no Netbeans
  - b. Adicionar uma Servlet
  - c. Alterá-lo para o Exemplo 2 anterior
  - d. Executar

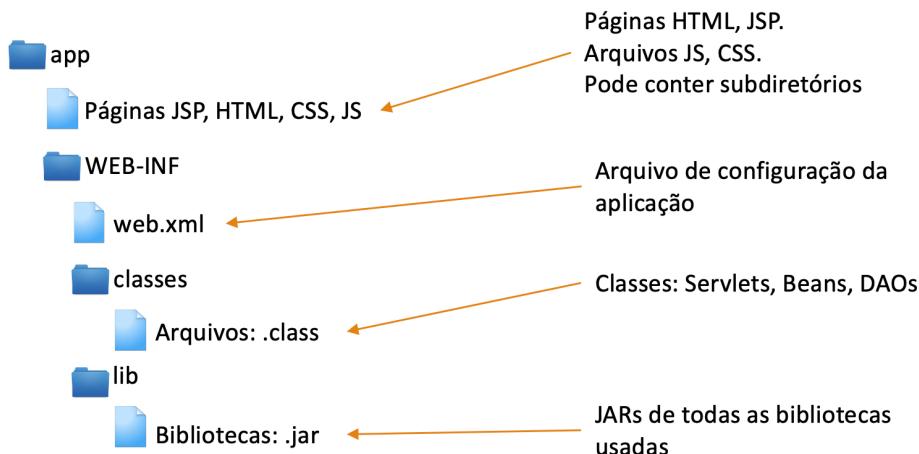
# 3 Arquivo de uma Aplicação Java Web

## Aplicação Java Web

- Aplicações Java Web estão empacotadas em arquivos **.WAR**
  - *Web Archive*
  - Zip
  - Estrutura de diretórios bem definida
  - Contém:
    - HTML, CSS, JS
    - JSPs
    - Servlets compiladas (.class)
    - Bibliotecas usadas na aplicação
    - Arquivos de configuração : web.xml

# Aplicação Java Web.

Arquivo: app.war



## Exercícios..

### 1. Verificar a hierarquia de diretórios de um .WAR

- Obter o .war de uma aplicação contendo um JSP e uma Servlet
  - Netbeans: Está no diretório DIST dentro do projeto
  - Eclipse: No projeto, escolher EXPORTAR para .WAR
- Renomear o arquivo .war para .zip
- Descompactar

# 4 Deploy / Undeploy

## Deploy / Undeploy.

- Deploy
  - Ato de instalar a aplicação no servidor de aplicação
  - Já deve-se ter um arquivo .WAR
  - Portanto a aplicação já deve estar empacotada
  - Torna a aplicação disponível para ser executada
  - Em tempo de desenvolvimento
    - Pelo Netbeans
  - Em tempo de execução
    - Via portal do servidor
- Undeploy
  - Ato de desinstalar a aplicação do servidor de aplicação
  - Torna a aplicação indisponível

# Tomcat

- Servidor de Aplicações WEB
  - Não implementa toda a especificação Java EE
  - Serve aplicações WEB : Servlets, JSP
- Responde aplicação na porta 8080 por default  
<http://localhost:8080/app>  
<http://www.servidor.com.br:8080/app>
- Sem o nome da aplicação entra na página do Tomcat  
<http://localhost:8080>  
<http://www.servidor.com.br:8080>

## Tomcat: <http://localhost:8080>

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

**Apache Tomcat/8.0.26**

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 Recommended Reading:  
[Security Considerations HOW-TO](#)  
[Manager Application HOW-TO](#)  
[Clustering/Session Replication HOW-TO](#)

[Server Status](#)  
[Manager App](#)  
[Host Manager](#)

**Developer Quick Start**

<a href="#">Tomcat Setup</a>	<a href="#">Realms &amp; AAA</a>	<a href="#">Examples</a>	<a href="#">Servlet Specifications</a>
<a href="#">First Web Application</a>	<a href="#">JDBC DataSources</a>		<a href="#">Tomcat Versions</a>

**Managing Tomcat**  
For security, access to the [manager webapp](#) is restricted. Users are defined in:  
`$CATALINA_HOME/conf/tomcat-users.xml`  
In Tomcat 8.0 access to the manager application is split between different users.  
[Read more...](#)

**Release Notes**  
[Changelog](#)  
[Migration Guide](#)  
[Security Notices](#)

**Documentation**  
[Tomcat 8.0 Documentation](#)  
[Tomcat 8.0 Configuration](#)  
[Tomcat Wiki](#)  
Find additional important configuration information in:  
`$CATALINA_HOME RUNNING.txt`  
Developers may be interested in:  
[Tomcat 8.0 Bug Database](#)  
[Tomcat 8.0 JavaDocs](#)  
[Tomcat 8.0 SVN Repository](#)

**Getting Help**  
[FAQ and Mailing Lists](#)  
The following mailing lists are available:  
[tomcat-announce](#)  
Important announcements, releases, security vulnerability notifications. (Low volume).  
[tomcat-users](#)  
User support and discussion  
[taglibs-user](#)  
User support and discussion for [Apache Taglibs](#)  
[tomcat-dev](#)  
Development mailing list, including commit messages

# Tomcat

If you're seeing this, you've successfully installed Tomcat. Congratulations!

**Developer Quick Start**

- Tomcat Setup
- First Web Application
- Realms & AAA
- JDBC DataSources
- Examples
- Servlet Specifications
- Tomcat Versions

**Managing Tomcat**

For security, access to the `manager webapp` is restricted. Users are defined in: `$CATALINA_HOME/conf/tomcat-users.xml`

In Tomcat 8.0 access to the manager application is split between different users. [Read more...](#)

**Documentation**

- [Tomcat 8.0 Documentation](#)
- [Tomcat 8.0 Configuration](#)
- [Tomcat Wiki](#)

Find additional important configuration information in:

`$CATALINA_HOME RUNNING.txt`

Developers may be interested in:

- [Tomcat 8.0 Bug Database](#)
- [Tomcat 8.0 JavaDocs](#)
- [Tomcat 8.0 SVN Repository](#)

**Getting Help**

**FAQ and Mailing Lists**

The following mailing lists are available:

- [tomcat-announce](#) Important announcements, releases, security vulnerability notifications. (Low volume).
- [tomcat-users](#) User support and discussion
- [taglibs-user](#) User support and discussion for [Apache Taglibs](#)
- [tomcat-dev](#) Development mailing list, including commit messages

# Tomcat

Inicio a sessão em localhost:8080  
A sua senha será enviada sem ser criptografada.

Nome de Usuário  
Senha  
 Memorizar esta senha

**Developer Quick Start**

- Tomcat Setup
- First Web Application
- Realms & AAA
- JDBC DataSources
- Examples
- Servlet Specifications
- Tomcat Versions

**Managing Tomcat**

For security, access to the `manager webapp` is restricted. Users are defined in: `$CATALINA_HOME/conf/tomcat-users.xml`

**Documentation**

- [Tomcat 8.0 Documentation](#)
- [Tomcat 8.0 Configuration](#)
- [Tomcat Wiki](#)

**Getting Help**

**FAQ and Mailing Lists**

The following mailing lists are available:

- [tomcat-announce](#)

# Tomcat

## SE não conseguir acessar

Usuário e Senha:

- Arquivo **tomcat-users.xml**
  - Dentro do subdiretório **CONF** da instalação do Tomcat
  - Criar um usuário/senha com as roles: manager-script, admin e manager-gui

Deve ter o seguinte conteúdo:

```
<?xml version="1.0" encoding="UTF-8"?>
<tomcat-users>
    <role rolename="manager-gui"/>
    <user password="tomcat"
        roles="manager-script,admin,manager-gui"
        username="tomcat"/>
</tomcat-users>
```

# Tomcat



✓ Página Inicial  
✓ Rolar até o final

## Tomcat Web Application Manager

Manager					
List Applications		HTML Manager Help		Manager Help	
Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions with idle ≥ 30 minutes</a>
/PrimeiroProjetoLindo	None specified		true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions with idle ≥ 30 minutes</a>
/TADS Tarde10	None specified		true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions with idle ≥ 30 minutes</a>
/TadsNoite2	None specified		true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions with idle ≥ 30 minutes</a>
/WebApplication1	None specified		true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions with idle ≥ 30 minutes</a>

# Tomcat

/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
----------	----------------	----------------------------	------	---	--

**Deploy**  
Deploy directory or WAR file located on server

Context Path (required):   
 XML Configuration file URL:   
 WAR or Directory URL:

**WAR file to deploy**

Select WAR file to upload: Seleccionar Arquivo | nenhum arquivo selecionado

**Diagnostics**  
Check to see if a web application has caused a memory leak on stop, reload or undeploy

Find leaks This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

**SSL connector configuration diagnostics**

Connector ciphers List the configured ciphers for each connector

**Server Information**

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/8.0.26	1.8.0_60-b27	Oracle Corporation	Mac OS X	10.13.2	x86_64	Neumann.local	192.168.25.100

Prof. Dr. Razer A N R Montaño

JAVA WEB

31

# Tomcat

/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
----------	----------------	----------------------------	------	---	--

**Deploy**

✓ Selecionar um arquivo .war

Context Path (required):   
 XML Configuration file URL:   
 WAR or Directory URL:

**WAR file to deploy**

Select WAR file to upload: Seleccionar Arquivo | nenhum arquivo selecionado

**Diagnostics**  
Check to see if a web application has caused a memory leak on stop, reload or undeploy

Find leaks This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

**SSL connector configuration diagnostics**

Connector ciphers List the configured ciphers for each connector

**Server Information**

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/8.0.26	1.8.0_60-b27	Oracle Corporation	Mac OS X	10.13.2	x86_64	Neumann.local	192.168.25.100

Prof. Dr. Razer A N R Montaño

JAVA WEB

32

## Tomcat.

The screenshot shows the Apache Tomcat Web Application Manager interface. At the top left, there is a logo for "The Apache Software Foundation" and a yellow cat icon. A red box highlights the message "✓ Aplicação instalada" (Application installed successfully) in the top left corner. Below the header, there is a navigation bar with tabs: "Manager", "List Applications", "HTML Manager Help", "Manager Help", and "Server Status". The main content area is titled "Applications" and contains a table with the following data:

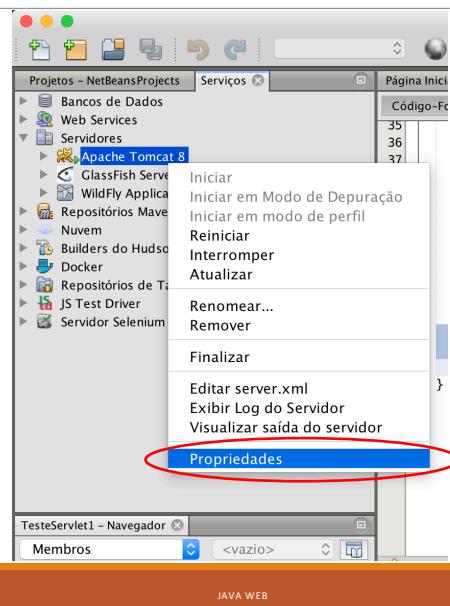
Path	Version	Display Name	Running	Sessions	Commands
!/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/PrimeiroProjetoLindo	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/TADSFazenda10	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/TadsNoite2	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/WebApplication1	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

At the bottom of the page, there is footer text: "Prof. Dr. Razer A N R Montaño", "JAVA WEB", and "33".

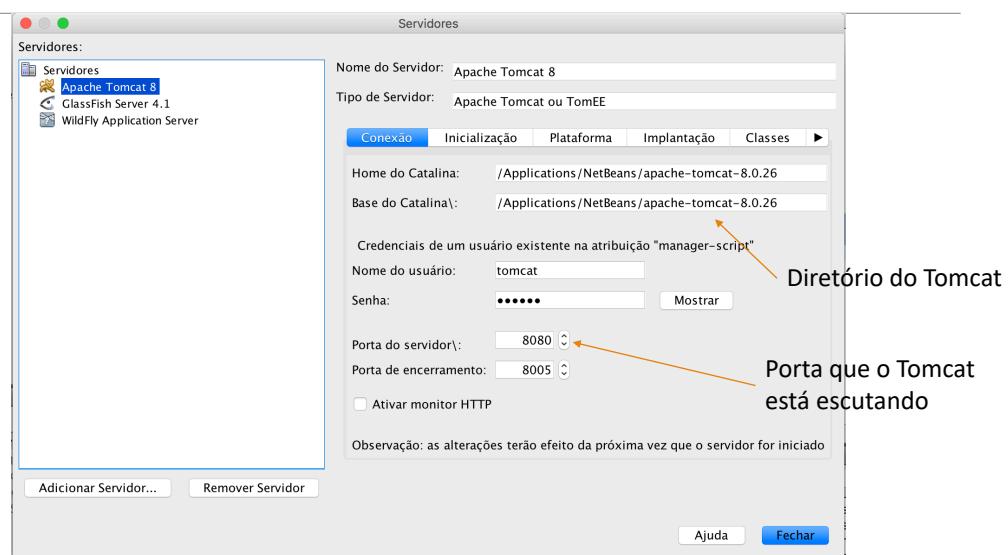
## Tomcat no Netbeans

The screenshot shows the Netbeans IDE interface. On the left, there is a "Projetos – NetBeansProjects" panel listing various services and servers, including "Apache Tomcat 8" which is highlighted with a blue box. An orange arrow points from the text "1. Aba Serviços" to this highlighted item. To the right of the projects panel is a "Serviços" tab, indicated by another orange arrow and the text "2. Botão Direito". The main workspace shows a code editor with some Java code. The footer of the screen includes the text "Prof. Dr. Razer A N R Montaño", "JAVA WEB", and "34".

## Tomcat no Netbeans



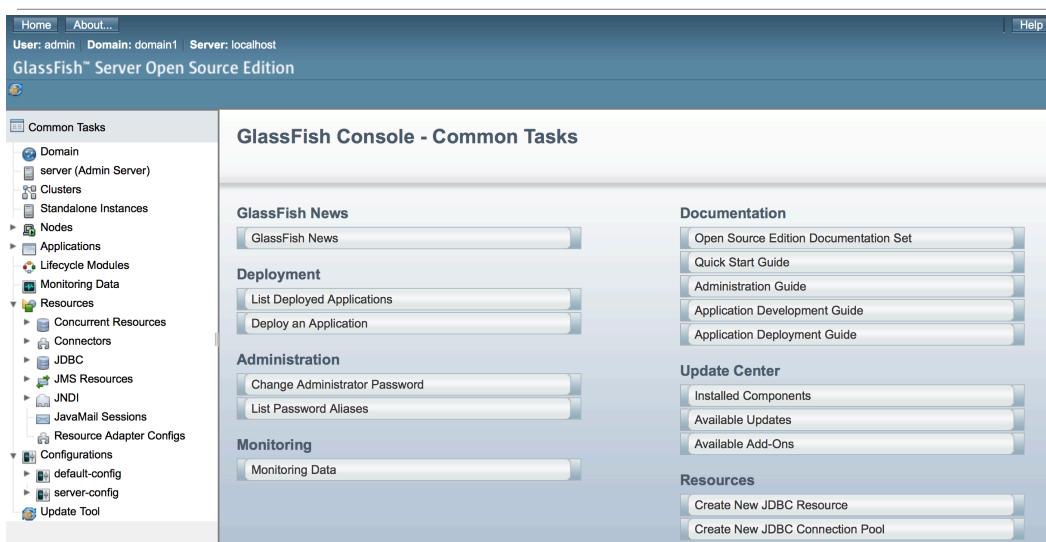
## Tomcat no Netbeans.



# Glassfish

- Servidor de Aplicações Java EE
  - Implementa toda a especificação Java EE
  - Serve aplicações Java EE completas
- Responde a aplicação na porta 8080 por default  
`http://localhost:8080/app`  
`http://www.servidor.com.br:8080/app`
- Responde a gerência na porta 4848 por default  
`http://localhost:4848`  
`http://www.servidor.com.br:4848`

## Glassfish: `http://localhost:4848`



The screenshot shows the GlassFish Server Open Source Edition console interface. At the top, there's a header bar with links for 'Home' and 'About...'. Below it, the status bar shows 'User: admin Domain: domain1 Server: localhost'. The main content area is titled 'GlassFish Console - Common Tasks'. On the left, a sidebar lists various management tasks under categories like 'Common Tasks', 'Resources', 'Configurations', and 'Deployments'. The right side contains several cards with links to 'GlassFish News', 'Documentation' (including links to the Open Source Edition Documentation Set, Quick Start Guide, Administration Guide, Application Development Guide, and Application Deployment Guide), 'Update Center' (listing Installed Components, Available Updates, and Available Add-Ons), and 'Resources' (links to Create New JDBC Resource and Create New JDBC Connection Pool).

# Glassfish

The screenshot shows the GlassFish Server Open Source Edition console. The title bar reads "GlassFish Server Open Source Edition". The left sidebar is titled "Common Tasks" and includes links for Domain, server (Admin Server), Clusters, Standalone Instances, Nodes, Applications (which is circled in red), Lifecycle Modules, Monitoring Data, Resources, Configurations, and Update Tool. The main content area is titled "GlassFish Console - Common Tasks" and contains sections for GlassFish News, Deployment (List Deployed Applications, Deploy An Application), Administration (Change Administrator Password, List Password Aliases), Monitoring (Monitoring Data), Documentation (Open Source Edition Documentation Set, Quick Start Guide, Administration Guide, Application Development Guide, Application Deployment Guide), Update Center (Installed Components, Available Updates, Available Add-Ons), and Resources (Create New JDBC Resource, Create New JDBC Connection Pool). At the bottom, the footer includes "Prof. Dr. Razer A N R Montaño", "JAVA WEB", and "39".

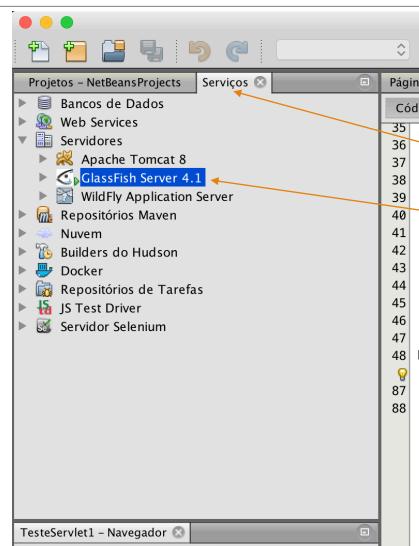
# Glassfish.

The screenshot shows the GlassFish Server Open Source Edition console. The title bar reads "GlassFish Server Open Source Edition". The left sidebar is titled "Common Tasks" and includes links for Domain, server (Admin Server), Clusters, Standalone Instances, Nodes, Applications (which is selected and highlighted in blue), Lifecycle Modules, Monitoring Data, Resources, Configurations, and Update Tool. The main content area is titled "Applications" and displays a table titled "Deployed Applications (5)". The table has columns for Select, Name, Deployment Order, Enabled, Engines, and Action. The data in the table is as follows:

Select	Name	Deployment Order	Enabled	Engines	Action
<input type="checkbox"/>	MobileWS	100	<input checked="" type="checkbox"/>	web	Launch   Redeploy   Reload
<input type="checkbox"/>	PrimeiroProjetoLindo	100	<input checked="" type="checkbox"/>	web	Launch   Redeploy   Reload
<input type="checkbox"/>	ProjetoMVCCidade	100	<input checked="" type="checkbox"/>	web	Launch   Redeploy   Reload
<input type="checkbox"/>	ProjetoMVC Pessoa	100	<input checked="" type="checkbox"/>	web	Launch   Redeploy   Reload
<input type="checkbox"/>	TesteDAO	100	<input checked="" type="checkbox"/>	web	Launch   Redeploy   Reload

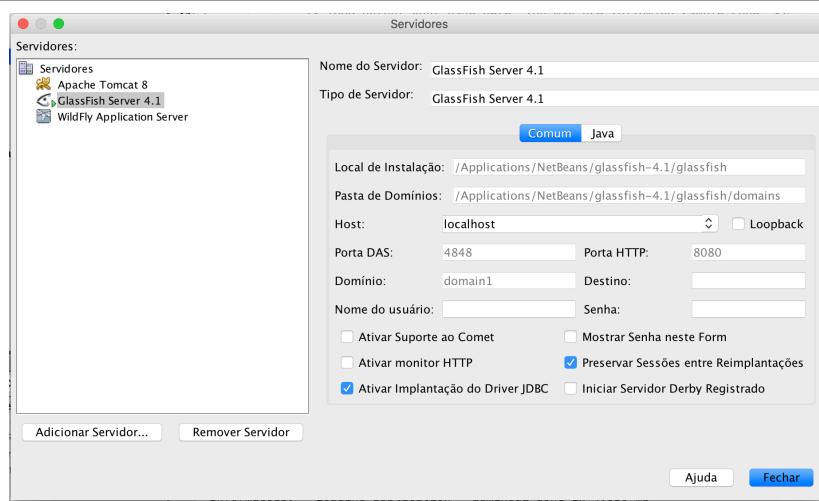
At the bottom, the footer includes "Prof. Dr. Razer A N R Montaño", "JAVA WEB", and "40".

## Glassfish no Netbeans



1. Aba Serviços
2. Botão Direito
3. Propriedades

## Glassfish no Netbeans.





## Exercícios..

1. Escreva uma JSP que mostra seu nome 100 vezes (usando laço)
2. Escreva uma JSP que mostra a tabuada do 1 ao 9, usando laços e tabela HTML
3. Escreva uma Servlet que mostra a data e hora atuais
4. Escreva uma Servlet que mostra a tabuada do 1 ao 9, usando laços e tabela HTML