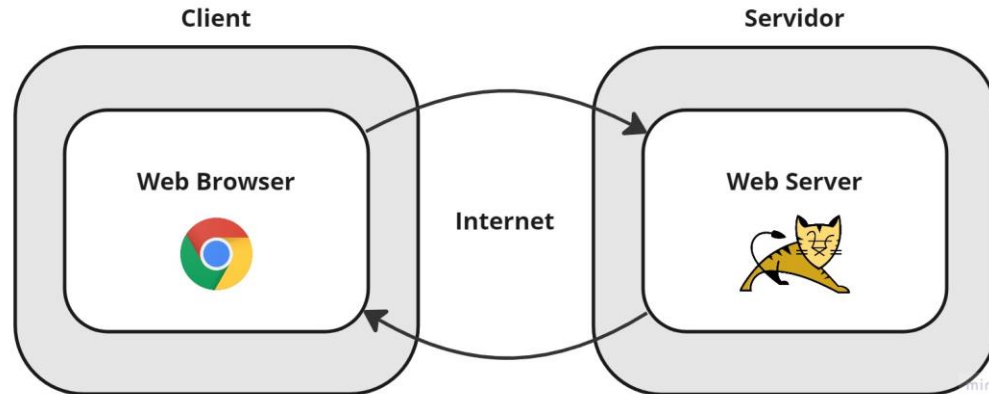


Conceitos sistemas web e JSP

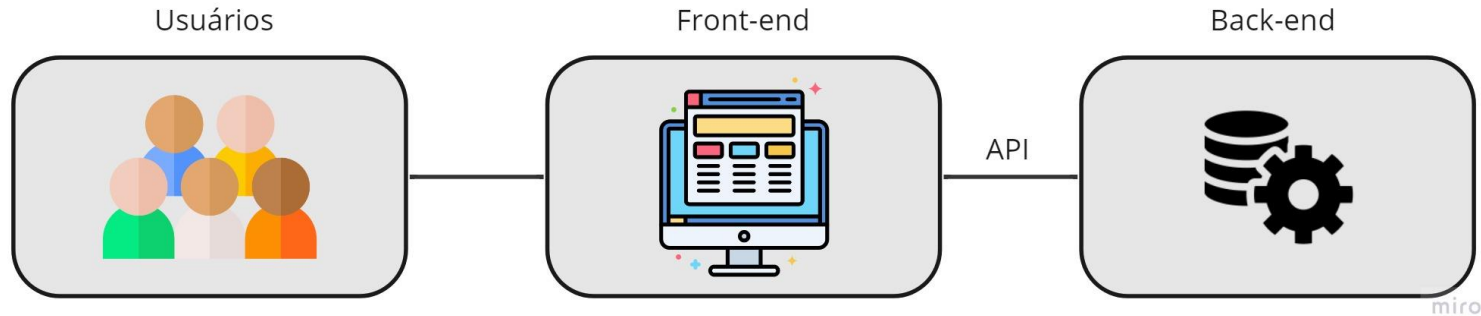
Mateus Elias Gündel | mateus8923@gmail.com

- É um sistema hospedado em um servidor web.
- Funciona recebendo solicitações e fornecendo uma resposta adequada a solicitação que foi recebida.
- O protocolo principal é o Protocolo HTTP (HyperText Transfer Protocol), o qual determina o padrão de comunicação entre os clientes e servidores.
- HTTP é a mesma coisa que HTTPS?
- SSL/TLS - Secure Sockets Layer/Transport Layer Security.
- Outros protocolos também utilizados em sistemas web são: FTP (arquivos); SMTP, POP e IMAP(mensagens de e-mail); DHCP (distribuição de Ip); SSH (conexão remota).

- Funciona em qualquer dispositivo com um browser e conexão com a internet.
- A maioria das pessoas já utiliza um navegador
- Todas as atualizações são realizadas em um servidor somente, sem precisar de reinstalação.
- Melhorias de infraestrutura não são percebidas pelos clientes.



- **Usuários:** Quem vai usar e visualizar
- **Front-end:** Toda a parte visual do site, onde o usuário irá interagir e navegar na interface construída, normalmente é construída em HTML e CSS com o uso do Javascript.
- **Back-end:** Onde toda a lógica e as regras estão principalmente aplicadas, também é responsável por realizar toda a comunicação com outras aplicações e com os bancos de dados. Podem ser usadas diversas linguagens de programação aqui, como Java, Python, C#, PHP, Node.js dentre outras.
- **API:** Implementação de normas, para a comunicação entre as plataformas.



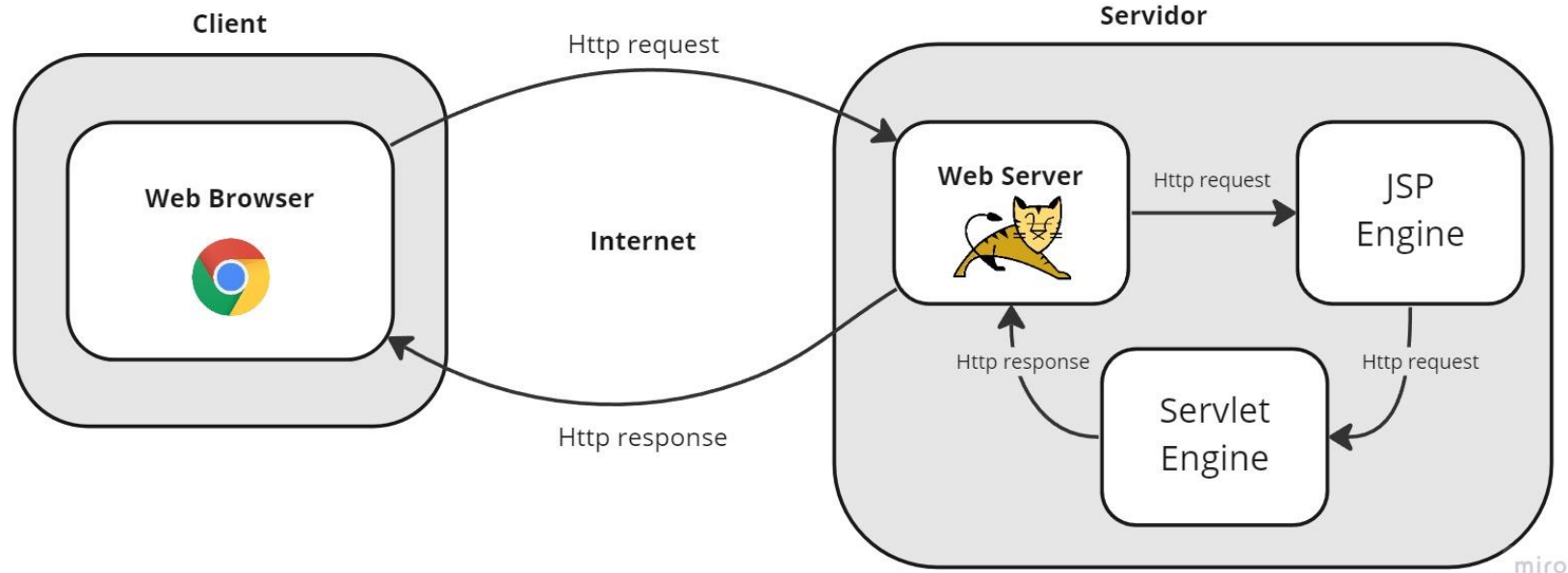
- JSP
- Servlets
- JSF
- Struts
- Spring
- JavaBeans
- Applets
- Click
- Wicket

- A linguagem JSP (Java Server Pages), é utilizada para construir páginas dinâmicas com o Java.
- A página não é dinâmica no browser em si, mas sim, é montada dinamicamente, no momento em que é solicitada.
- Assim, o JSP é a junção do HTML com o JAVA.

Exemplo JSP:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <%
      out.println("<br>A página acessada é " + request.getServerName()
        + ":8080" + request.getContextPath());
    %>
  </body>
</html>
```

- A imagem abaixo detalha como é realizada a montagem da página JSP quando solicitada pelo usuário.



O que é preciso para executar um JSP:

- Java
- Servidor Web (Tomcat)
- Navegador

Para criar o html junto com o java, o JSP tem algumas sintaxes específicas.

Começamos com a tag principal, que define onde é script java `<% e %>`, dentro desta tag, pode ser criado todo tipo de código java.

`<% seu código aqui %>`

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <%
      out.println("<br>A página acessada é " + request.getServerName()
        + ":8080" + request.getContextPath());
    %>
  </body>
</html>
```

Temos também a tag de declarações `<%! e %>`, que são utilizadas para declarar as variáveis ou métodos que serão usados no JSP.

`<%! Sua declaração aqui %>`

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <%! int idade = 26;%>
3  <!DOCTYPE html>
4  <html>
5      <head>
6          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7          <title>JSP Page</title>
8      </head>
9      <body>
10         <%
11             out.println("<br>Sua idade é: " + idade);
12         %>
13     </body>
14 </html>
```

Temos também a tag de expressões `<%=` e `%>`, que são utilizadas para inserir uma string diretamente no HTML, sem precisar fazer um print. Importante lembrar que não se usa “;” na tag de expressão.

`<%=` Sua expressão aqui `%>`

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <%! public int getIdade() {return 26;}%>
3  <!DOCTYPE html>
4  <html>
5      <head>
6          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7          <title>JSP Page</title>
8      </head>
9      <body>
10         <br>Sua idade é: <%=getIdade()%>
11     </body>
12 </html>
```

Temos também a tag de comentário `<%-- e --%>`, que é utilizada para inserir qualquer comentário que julgue necessário ou comentar qualquer coisa. O que estiver entre a tag, será ignorado pela engine na hora de gerar o html.

`<%-- Seu comentário aqui --%>`

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <%! public int getIdade() {return 26;}%>
3  <!DOCTYPE html>
4  <html>
5      <head>
6          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7          <title>JSP Page</title>
8      </head>
9      <body>
10         <br>Sua idade é: <%=getIdade()%>
11     </body>
12 </html>
```

Temos 3 tipos de diretivas, que funcionam de forma diferente e afetam diretamente a estrutura do nosso JSP.

<%@ page ... %>	Define as instruções para a criação da página, pode ser incluída em qualquer lugar do JSP.
<%@ include ... %>	É utilizado para adicionar conteúdo no nosso JSP, como por exemplo outros arquivos JSP.
<%@ taglib ... %>	É utilizada para criar tags customizadas para usar no Html

A diretiva page, possui alguns parâmetros específicos:

contentType	Define o tipo de conteúdo da página.
import	Especifica a lista de package java que deve ser importado.
pageEncoding	Define o encoding da página.
errorPage	Define a url de uma página de erro.
extends	Especifica uma classe que deve ser estendida.
session	Especifica se a página JSP tem acesso ao http.
Info	Define a informação disponível em <code>getServletInfo()</code> .

Exemplo de uso do page, para configurar algumas informações e realizar o import do Date, para exibir a data atual da página.

E também o uso do include, para adicionar o header criado em outra página jsp a esta página, assim podendo modularizar os jsp.

```
1  <%@page contentType="text/html" pageEncoding="UTF-8" import="java.util.Date"%>
2
3  <!DOCTYPE html>
4  <html>
5      <head><title>Exemplo de while</title></head>
6      <body>
7          <%@include file="header.jsp"%>
8          <h1>A data atual é:
9              <%= new Date () %>
10         </h1>
11     </body>
12 </html>
13
```

As actions são outra forma de adicionar algumas funcionalidades a página.

jsp:include	Inclui um arquivo quando a página é requisitada.
jsp:useBean	Instancia um JavaBean.
jsp:setProperty	Seta um propriedade do JavaBean.
jsp:getProperty	Busca uma propriedade do JavaBean para ser usada na página.
jsp:forward	Faz o forward da requisição para outra página.
jsp:plugin	Usado para definir e executar um plugin java.
jsp:element	Define elementos xml dinâmicos na página.
jsp:attribute	Define atributos xml dinâmicos na página.
jsp:body	Define o body xml dinâmico na página.
jsp:text	Utilizado para escrever templates de texto nas paginas.

Existem alguns objetos, que não precisam ser declarados, que já estão disponíveis para o uso.

request	Objeto HttpServletRequest , contendo os dados da request.
response	Objeto HttpServletResponse , contendo os dados da response.
out	Objeto PrintWriter , contendo informações definidas no servlet.
session	Objeto HttpSession , que é vinculado a request.
application	Objeto ServletContext , correspondente ao contexto da aplicação.
config	Objeto ServletConfig , contendo as configurações da página.
pageContext	Objeto utilizado para atualizar atributos da página.
page	É um sinônimo ao this nas classes.
Exception	Utilizado para controlar os dados de Exceção pelo JSP.

Todos os controles de fluxo e tomada de decisão do java, podem ser usados no JSP.

Utilizando o if else para verificar se já é maior de idade:

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <%! public int getIdade(){return 26;}%>
3  <!DOCTYPE html>
4  <html>
5      <body>
6          <% if (getIdade() > 18){%>
7              <p>Você é maior de idade</p>
8              <%}else{%>
9                  <p>Você é menor de idade</p>
10             <%}>
11         %>
12     </body>
13 </html>
```

Utilizando o switch para exibir o dia da semana:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%! public int getDiaSemana() {return 5;}%>
<!DOCTYPE html>
<html>
<body>
    <% switch(getDiaSemana()) {
        case 0:
            out.println("É domingo");
            break;
        case 1:
            out.println("É segunda");
            break;
        case 2:
            out.println("É terça");
            break;
        case 3:
            out.println("É quarta");
            break;
        case 4:
            out.println("É quinta");
            break;
        case 5:
            out.println("É sexta");
            break;
        default:
            out.println("É sábado");
        }
    %>
</body>
</html>
```

Utilizando o for para exibir a mensagem “Tutorial de JSP!” várias vezes e alterar o tamanho da fonte de exibição:

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <%! int fontSize; %>
4  <html>
5      <head><title>Exemplo de for</title></head>
6      <body>
7          <%for ( fontSize = 1; fontSize <= 3; fontSize++){ %>
8              <font color = "green" size = "<%= fontSize %>">
9                  Tutorial de JSP!
10             </font><br />
11             <%= %>
12         </body>
13     </html>
14
```

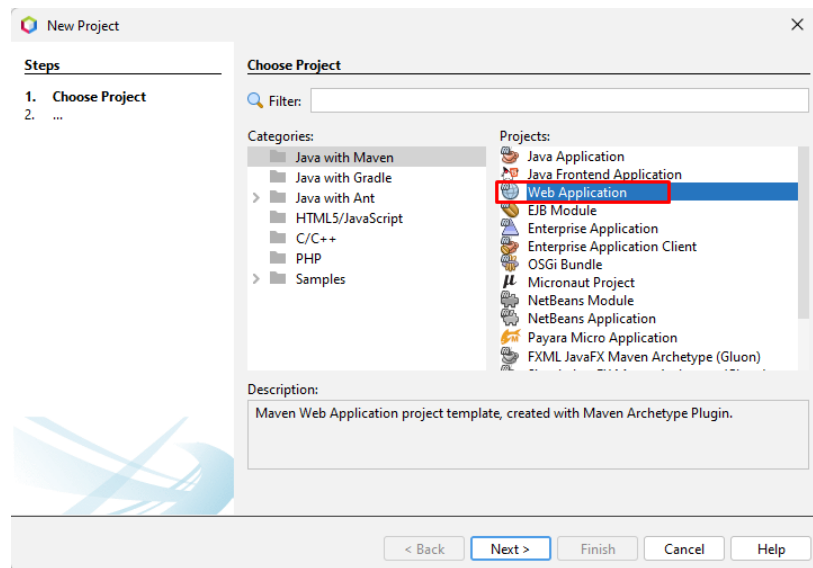
Utilizando agora o while para exibir a mensagem “Tutorial de JSP!” várias vezes e alterar o tamanho da fonte de exibição:

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <%! int fontSize; %>
4  <html>
5      <head><title>Exemplo de while</title></head>
6      <body>
7          <%while ( fontSize <= 3){ %>
8              <font color = "green" size = "<%= fontSize %>">
9                  Tutorial de JSP!
10             </font><br />
11             <%fontSize++;%>
12         <%}%>
13     </body>
14 </html>
```

Criação de um projeto Web

E agora, mãos à obra, vamos criar uma página JSP e implementar as nossas primeiras páginas!

Primeiro, abra o NetBeans, crie um novo projeto, selecione Java com Maven e selecione aplicação web, então vá em next.



Criação de um projeto Web

Após selecionar o projeto web, será necessário configurar algumas informações, como nome do projeto, a pasta de destino do projeto e o group_id, que é como o package do java será estruturado.

Particularmente, eu só adiciono um nome ao projeto e altero o group_id para ficar mais legível com à organização.

New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Settings

Name and Location

Project Name: projetoWeb

Project Location: C:\Users\mateu\OneDrive\Documentos\NetBeansProjects

Project Folder: teu\OneDrive\Documentos\NetBeansProjects\projetoWeb

Artifact Id: projetoWeb

Group Id: com.unisc

Version: 1.0-SNAPSHOT

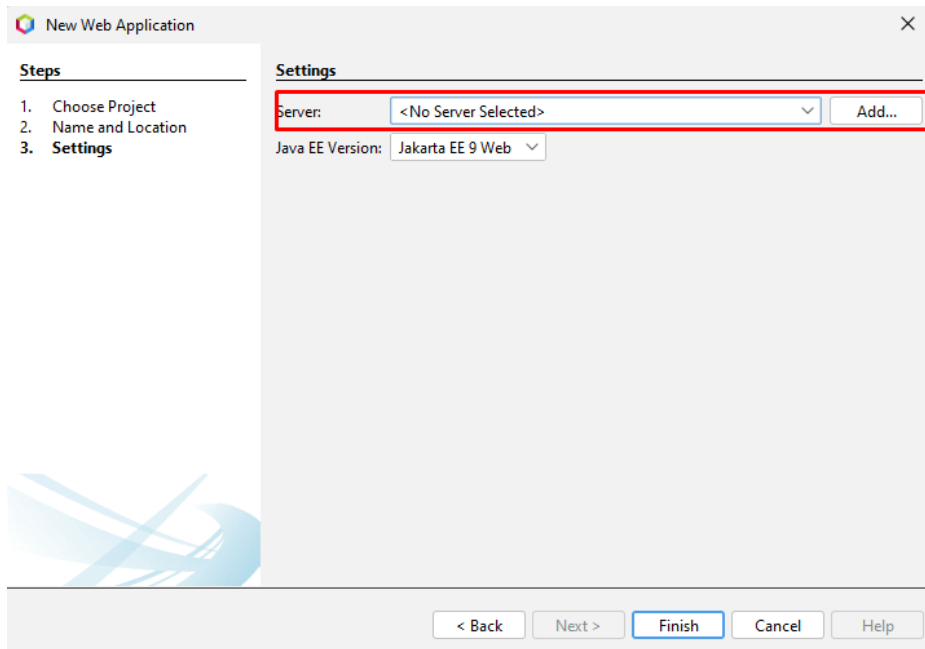
Package: com.unisc.projetoWeb (Optional)

< Back Next > Finish Cancel Help

Criação de um projeto Web

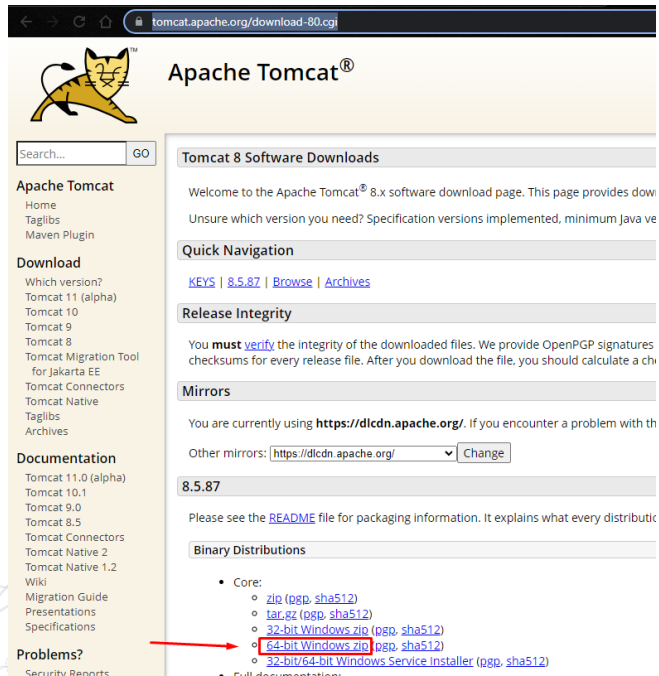
Após configurar as informações básicas do projeto, como este é um projeto web, precisaremos definir o nosso servidor web. Usaremos o tomcat.

Se tiver à opção do Tomcat 8, podem escolher esta opção e pular para o próximo slide de Criação de um projeto Web. Se não tiver, será necessário adicionar o Tomcat ao nosso netbeans.

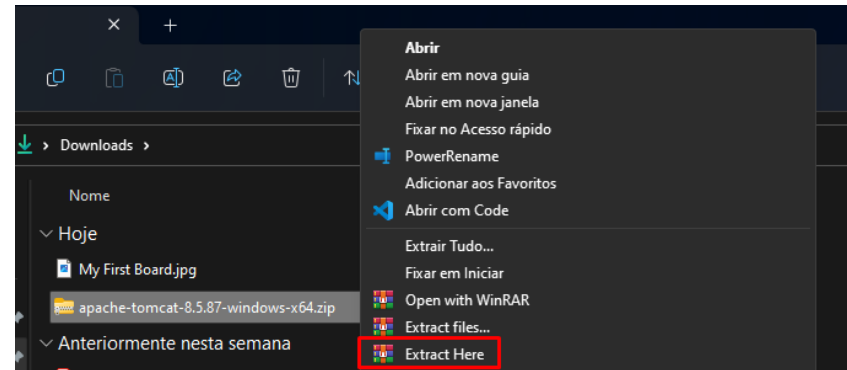


Se já fez o download do Tomcat 8, pode passar ao próximo slide. Primeiro passo, é acessar a página de Download do Tomcat, fazer o download do arquivo zipado e então, tirar ele do zip para podermos ler no Netbeans.

Link de download: <https://tomcat.apache.org/download-80.cgi>

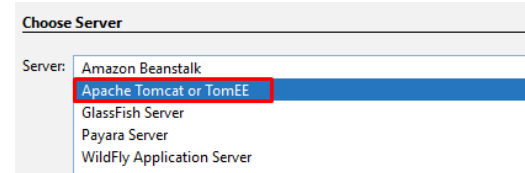
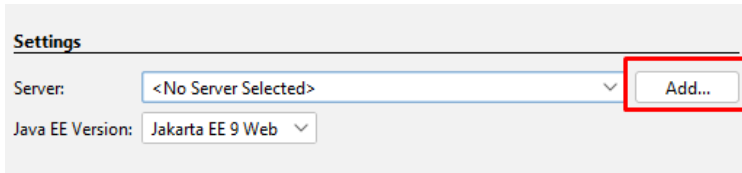


Após baixar, é necessário extrair, isso pode ser realizado, clicando com o botão direito no arquivo, e então em extrair aqui ou extrair tudo, dependendo do seu computador.

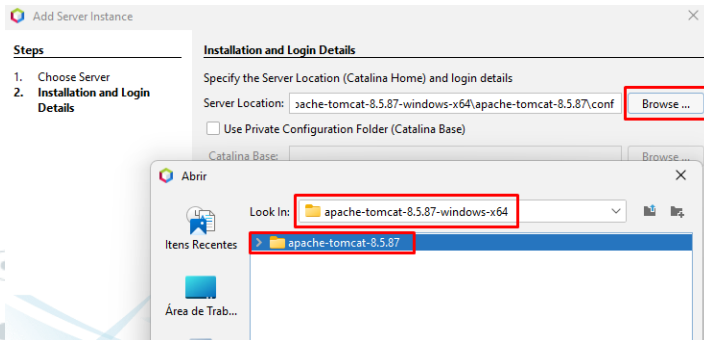


Com o Tomcat baixado, podemos dizer ao Netbeans onde encontrá-lo para rodar a nossa aplicação web.

Nas configurações, selecione Add, e então selecione Apache Tomcat or TomEE



Após selecionar, vá em próximo, e selecione o local do arquivo que extraiu o tomcat.



Cuidado ao selecionar a pasta, selecione a pasta que extraiu, e dentro terá a pasta apache-tomcat-8.5.87, selecione ela e então vá em abrir.

Como última configuração do Tomcat, será necessário adicionar um usuário e uma senha, pode ser qualquer uma, pois o tomcat irá perguntar essas credenciais ao executar a aplicação. Por exemplo, utilizei usuário e senha, “admin”.

Add Server Instance

Steps

1. Choose Server
2. **Installation and Login Details**

Installation and Login Details

Specify the Server Location (Catalina Home) and login details

Server Location:

☐ Use Private Configuration Folder (Catalina Base)

Catalina Base:

Enter the credentials of an existing user in the manager or manager-script role

Username:

Password:

☒ Create user if it does not exist

< Back Next > **Finish** Cancel Help

Como última configuração do Tomcat, será necessário adicionar um usuário e uma senha, pode ser qualquer uma, pois o tomcat irá perguntar essas credenciais ao executar a aplicação. Por exemplo, utilizei usuário e senha, “admin”. Ao final, basta ir em Finalizar.

Add Server Instance

Steps

1. Choose Server
2. **Installation and Login Details**

Installation and Login Details

Specify the Server Location (Catalina Home) and login details

Server Location:

☐ Use Private Configuration Folder (Catalina Base)

Catalina Base:

Enter the credentials of an existing user in the manager or manager-script role

Username:

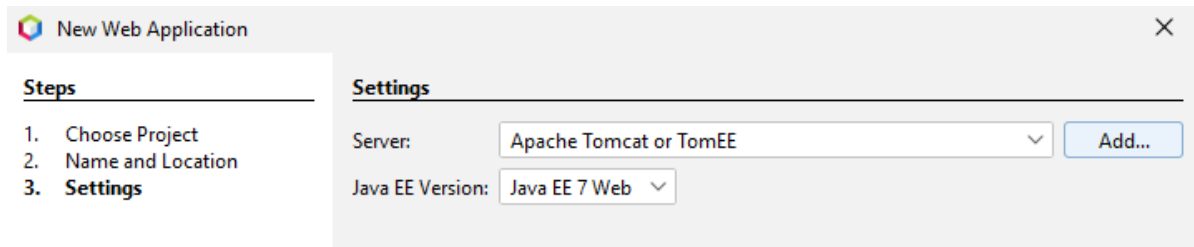
Password:

☒ Create user if it does not exist

< Back Next > **Finish** Cancel Help

Criação de um projeto Web

Com o servidor web configurado, podemos selecionar o nosso servidor e também a versão que iremos usar do Java EE Web.



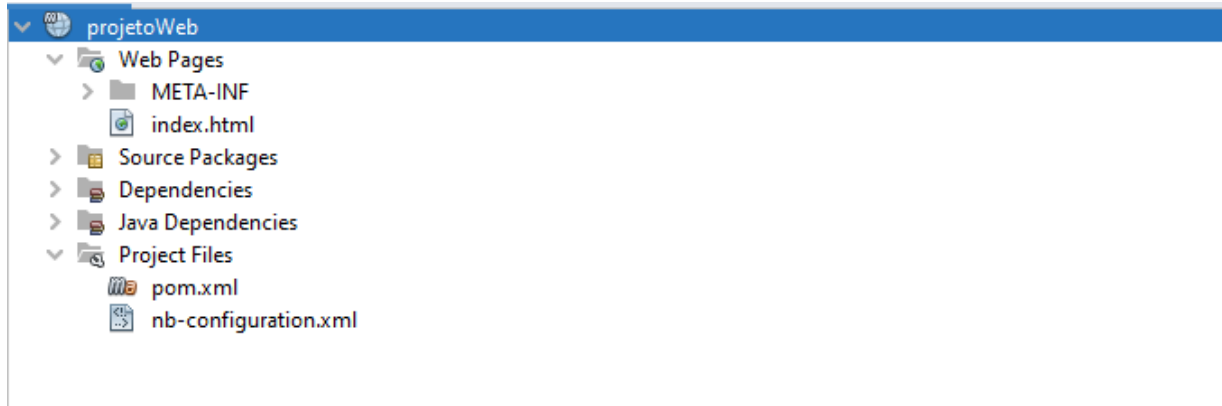
Devem estar estas opções disponíveis, selecionem elas e então, o botão de finalizar.

Pronto! O seu projeto web foi criado!

PS: Toda vez que precisar criar um projeto, e não tiver configurado o Tomcat ainda, será necessária realizar as configurações como demonstrado anteriormente.

Criação de um projeto Web

Ao ver o projeto a primeira vez, temos a seguinte estrutura.



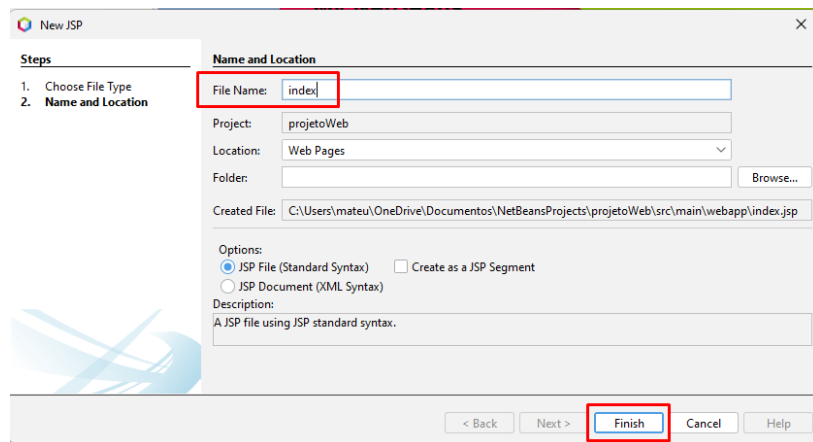
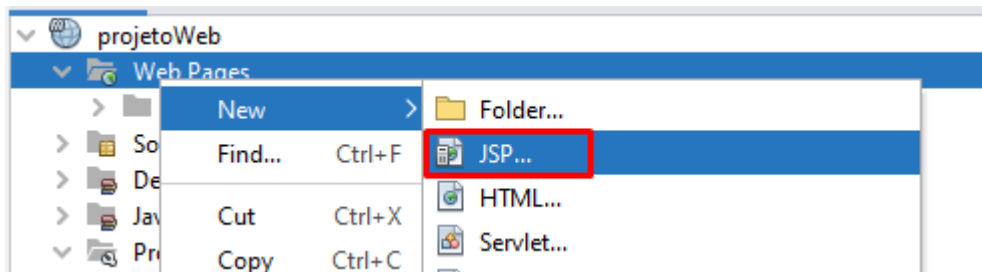
Como vamos somente usar JSP, então vamos utilizar somente a pasta Web Pages, pasta essa que armazena todos os arquivos que serão renderizados para os usuários como HTML. Assim o seguindo essa estrutura e criando nosso arquivos JSP ali, mantemos o projeto organizado.

Criação de um projeto Web

A primeira coisa que podemos fazer, é excluir o arquivo index.html e criar o arquivo index.jsp para usar o jsp.

Remova então o arquivo index.html, então use o botão direito na Pasta Web Pages, então vá em new e então em JSP...

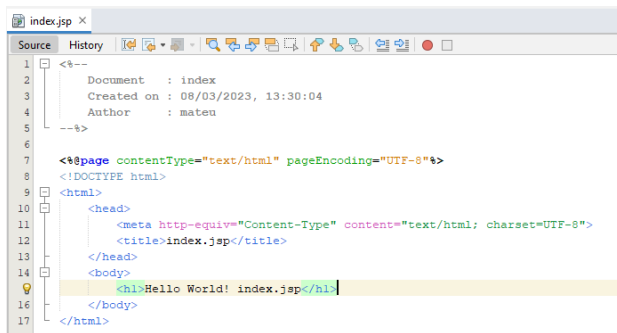
Ao abrir a página de criação do arquivo, altere o nome para “index” e então Finalize.



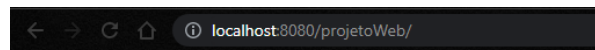
Criação de um projeto Web

Agora temos um projeto Web em java rodando com a página web em JSP.

Ao acessar a url <http://localhost:8080/projetoWeb/> (projetoWeb é o nome que deu para o seu projeto) sempre vamos ser redirecionados para index.jsp, essa funcionalidade é implícita na aplicação. Abaixo temos o exemplo do arquivo index.jsp criado e o resultado ao acessar.

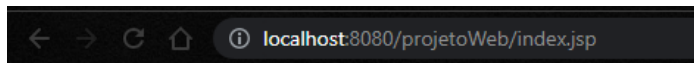


```
1 <!--
2 Document : index
3 Created on : 08/03/2023, 13:30:04
4 Author : mateu
5 -->
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>index.jsp</title>
13 </head>
14 <body>
15 <h1>Hello World! index.jsp</h1>
16 </body>
17 </html>
```



Hello World! index.jsp

Também podemos acessar <http://localhost:8080/projetoWeb/index.jsp> que aí vamos estar acessando diretamente o arquivo index.jsp. Abaixo, acessamos ele pela url diferente, mas o resultado continua igual.

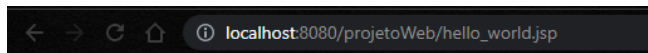
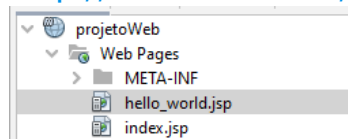


Hello World! index.jsp

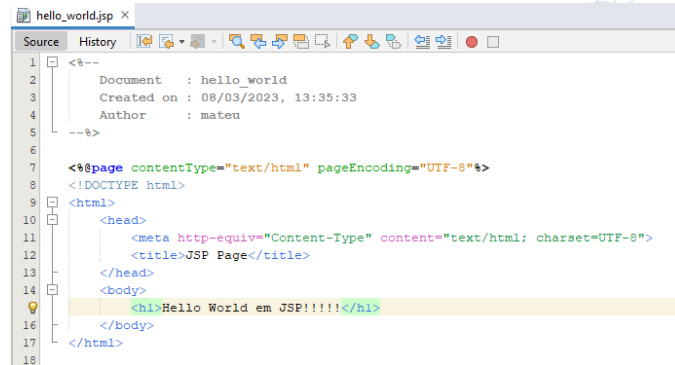
Criação de um projeto Web

Podemos criar um novo arquivo JSP e para acessá-lo, vamos utilizar o nome deste arquivo. Por exemplo, criando o arquivo `hello_world.jsp`, podemos acessá-lo através da url

http://localhost:8080/projetoWeb/hello_world.jsp.

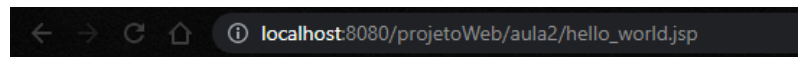
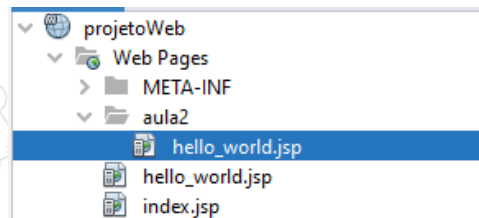


Hello World em JSP!!!!



Se criarmos uma pasta chamada `aula2`, dentro de `Web Pages`, para separar nossos arquivos JSP, e dentro dela, criar o mesmo arquivo `hello_world.jsp`, para acessar esse arquivo no browser, precisaremos utilizar a estrutura que criamos na url, adicionando a pasta `aula2` na url

http://localhost:8080/projetoWeb/aula2/hello_world.jsp.



Hello World em JSP!!!!

1. Crie uma página que exiba os números de 1 a 20 em uma lista.
2. Exiba agora somente os números pares de 1 a 20, em uma lista.
3. Crie uma página com uma tabela que exiba informações de 3 pessoas (crie essas 3 pessoas), a tabela deve conter 4 colunas, nome, peso, altura e IMC. Para calcular o IMC, deve ser criada uma função que realize este cálculo. $(\text{peso}/(\text{altura}*\text{altura}))$
4. Crie uma página dividida em 3 partes, header deve conter um cabeçalho com um botão para acessar a página de lista e a página com o cálculo de IMC criados anteriormente, o body precisa exibir a informação de data e hora atual e no footer deve estar o seu e-mail.