

Transcrição

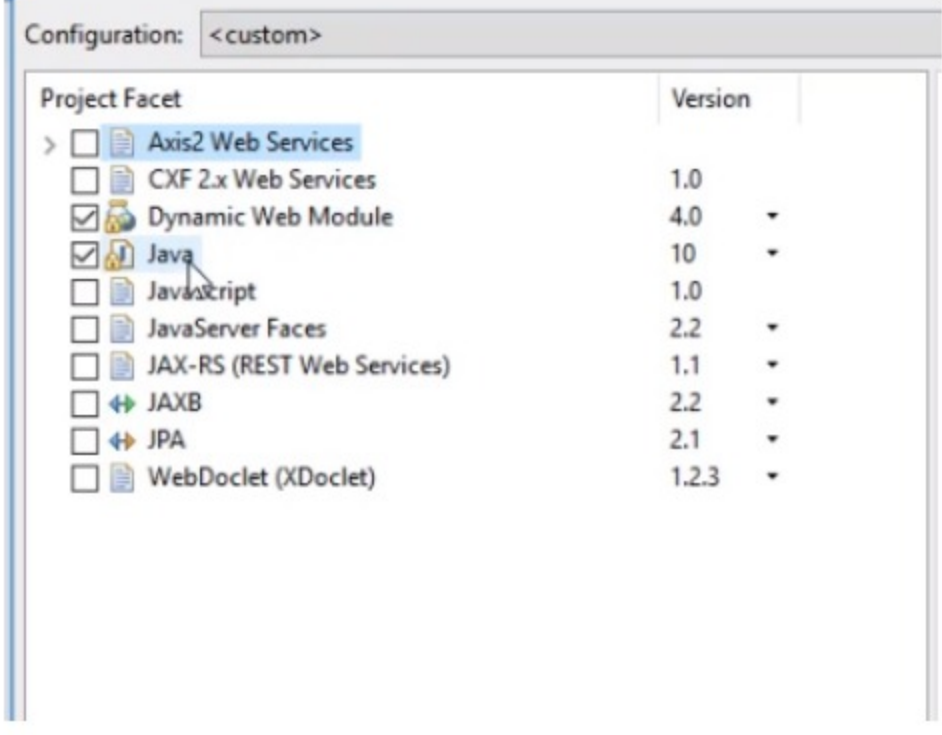
Nosso ambiente de trabalho está configurado com os elementos Java, Tomcat e Eclipse. Já sabemos como executar o servidor Tomcat e acessá-lo, ainda que por enquanto só conseguimos visualizar uma mensagem de erro emitida por este mesmo servidor, alegando que não há conteúdo para ser exibido.

Criaremos um projeto com Java e o associaremos com este servidor para que tenhamos algo a ser exibido no navegador. No Eclipse, clicaremos sobre o "File" no cabeçalho de ferramentas, em seguida, selecionaremos as opções "New > Dynamic Web Project". Repare que temos uma gama vasta de opções para criar esse projeto, como "Maven Project", "Enterprise Application Project" e tantos outros. Não estamos lidando apenas com um simples projeto Java, afinal precisamos trabalhar com arquivos web, HTML e talvez alguma imagem. Exatamente por essa complexidade, escolhemos a opção "Dynamic Web Project".

Na nova caixa de diálogo "Dynamic Web Project", daremos um nome para este novo projeto ("Projec name"), que será gerenciador . É importante que você utilize o mesmo nome, pois assim os links que criarmos estarão sincronizados com a documentação do curso e os exercícios que faremos.

Em seguida, definiremos o local do projeto ("Project location") e na opção "Target runtime" manteremos a configuração "Apache Tomcat v9.0". Na opção "Dynamic web module version" também manteremos a configuração "4.0".

Temos a opção "Configuration" que também não alteraremos, aliás, é muito raro que tenhamos de realizar qualquer tipo de alteração neste campo, mas clicaremos em "Modify" apenas para propósitos expositivos. Na nova janela teremos a opção de, por exemplo, especificar que o nosso projeto é executado com `JavaServerFaces` .



Voltaremos a janela "Dynamic Web Project" Poderíamos finalizar e de fato começar o projeto clicando sobre o botão "Finish", porém avançaremos("Next") um pouco mais para que você conheça as outras configurações possíveis.

Na nova fase da caixa de diálogo, veremos na área "Source folders on build path" um arquivo `src` . Como criaremos um projeto Java, teremos uma pasta `src` que abrigará nossas classes. Por padrão, o Eclipse compila para uma pasta "build\classes".

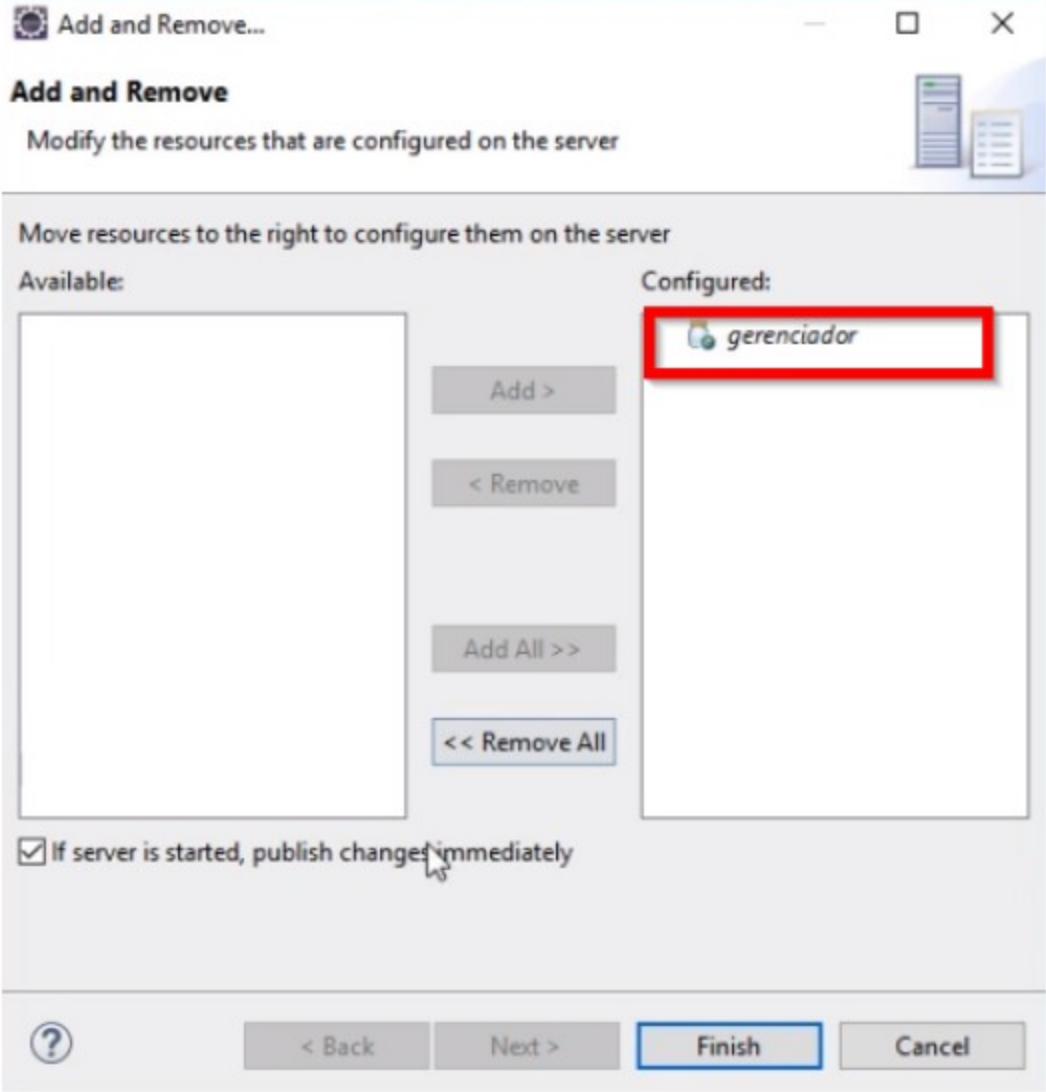
Prosseguindo nas configurações, encontraremos o campo "Context root" que possui o nome `gerenciador` , isto é, o título do projeto. O context root é como o projeto é chamado na url. Em seguida, teremos o campo "Content directory", configurado como `WebContent` , isto é, os arquivos relacionados ao mundo web, como páginas HTML, serão armazenadas em uma pasta `WebContent` .

Mais abaixo, assinalaremos a opção "Generate web.xml deployment descriptor", a criação de um arquivo de configuração. Servlet é uma especificação Java, embora ainda não saibamos o que é isso exatamente, sabemos que cada especificação Java possui seu `.xml`. No mundo Servlet, esse arquivo é o `web.xml`. No mundo JPA - outra especificação -, por exemplo, esse arquivo é chamado `persist.xml`.

Atualmente, esses arquivos `.xml` são menos importantes e podemos criar nosso projeto sem ele. Contudo, geraremos o arquivo para posteriormente realizarmos algumas configurações. Finalmente, pressionaremos o botão "Finish". O Eclipse fará o processamento das configurações, o que pode levar alguns minutos.

Terminado o processamento, teremos na "Projec Explorer" o projeto `gerenciador` , e ao clicarmos no arquivo teremos alguns recursos criados automaticamente. O mais importante é te tempos a pasta `src` armazenada em `Java Resources` . Teremos, também, a pasta `build` e `WebContent` , que armazena alguns arquivos padrão, incluindo o `web.xml` . Não nos preocuparemos em detalhes com isso neste momento, cada ponto será explicado ao longo do curso.

A próxima etapa é associar o projeto `gerenciador` ao Tomcat. Podemos simplesmente arrastar a pasta `gerenciador` para a aba "Servers" ou clicar sobre ela com o botão direito e selecionar a opção "Add and Remove..." dessa forma será exibida uma caixa de diálogo com todos os projetos disponíveis e compatíveis com o Tomcat. Selecionaremos o projeto e clicaremos sobre o botão "Add".



Ao executarmos o Tomcat no Eclipse, veremos que o projeto `gerenciador` já está associado ao servidor. Contudo, ainda não temos nada a ser exibido. Clicaremos com o botão direito sobre a pasta `WebContent` e selecionaremos "New > File" para criar um novo arquivo chamado `bem-vindo.html` , o formato entendido pelo navegador.

Criaremos um código bem simples com elementos básicos, o elemento raiz `<html>` e o corpo da mensagem envolvido por `<body>` .

```
<html>

<body>
  Bem vindo no curso Servlets da Alura
</body>

</html>
```

Queremos acessar essa página a partir do navegador. Acessaremos o endereço `localhost:8080` , mas o que veremos é a mensagem de erro 404, como se ainda faltasse conteúdo a ser exibido. O problema é que precisamos especificar o acesso ao projeto `gerenciador` , logo, o endereço que digitaremos no navegador será `localhost:8080/gerenciador/` .

Ao pressionarmos "Enter" veremos a mesma mensagem de erro. Não estamos sendo específicos o suficiente, precisamos explicitar a página `bem-vindo.html` , logo `localhost:8080/gerenciador/bem-vindo.html` . Assim feito, teremos a mensagem `Bem vindo no curso Servlets da Alura!` .

O Tomcat recebeu a chamada do navegador, isto é, uma requisição, e foi passado o conteúdo que criamos no projeto. Ao analisarmos o código fonte da página, veremos exatamente o código escrito no arquivo `bem-vindo.xml` , armazenado na pasta `WebContent` .

Para acessar esse conteúdo no navegador, não precisamos escrever `localhost:8080/gerenciador/WebContent/bem-vindo.html` , e caso escrevêssemos dessa maneira, o conteúdo não seria exibido. O Tomcat realiza esse percurso de forma automática. O mesmo conteúdo pode ser exibido em outras máquinas, desde que saibamos seu IP.

Ainda não realizamos nenhum trabalho com Servlet, mas estamos avançando!