

Desenvolvimento Web I

Aula 01 - Ambiente de desenvolvimento para Web em Java

Apresentação

Nesta aula, você aprenderá a instalar e configurar o nosso ambiente de desenvolvimento e depuração (debug). Para isso, vamos aprender a organizar e instalar três ferramentas muito importantes para o nosso estudo. São elas: Eclipse, Apache Tomcat e o JDK. Então, vamos colocar a mão na massa?!



Vídeo 01 - Apresentação

Objetivos

- Instalar os softwares necessários para construir, executar e depurar (debugar) suas aplicações em um ambiente Java para Web.

Baixando e instalando os softwares necessários

Dos três softwares mencionados na apresentação, o que você realmente precisa para executar um sistema Java web é do *Apache Tomcat*, que é uma implementação de código aberto para os ambientes de *Servlets* e JSP, e o do JDK, que traz consigo as classes padrão Java e a máquina virtual para compilar e executar código Java. O *Eclipse* é apenas uma ferramenta que facilita o nosso desenvolvimento, mas nada nos impediria de construirmos nossas aplicações sem ele (por exemplo, usando um editor de texto e um terminal de comandos). Porém, o esforço e o tempo de construção de programas Web são bem menores usando ambientes de desenvolvimento como o *Eclipse*.

Pois bem, comecemos por fazer o download e instalar o JDK versão 8 na seguinte URL: <http://www.oracle.com/technetwork/pt/java/javase/downloads/index.html>. Na opção “JDK” clique em Download e em seguida, na lista de Downloads na sessão “Java SE Development Kit 8u152”, marque a opção “Accept License Agreement” para que os arquivos fiquem disponíveis para serem baixados. Selecione a versão adequada para o seu sistema operacional (No caso do Windows deve ter o nome similar a “jdk-8u152-windows-x64.exe”) e uma vez concluído o download, vamos agora instalar o JDK. Para isso, use a instalação padrão, sem maiores intervenções.

Depois de instalar, vamos baixar da Internet e descompactar o Apache Tomcat, um dos servidores web mais conhecidos no mercado. Para isso acesse a seguinte URL: <https://tomcat.apache.org/download-80.cgi>. Apesar de ter uma versão para instalação automática (um executável), é interessante instalar o Apache Tomcat através do arquivo compactado (.zip), já que iremos ativar e desativar o Tomcat apenas através do Eclipse. Basta então baixar o arquivo zip indicado para o seu sistema operacional e descompactá-lo no diretório de sua preferência. Para efeito de padronização, crie uma pasta chamada “desen” dentro sua pasta “Documentos”. Essa pasta “desen” será utilizada para guardar as ferramentas que iremos utilizar. Assim, ao término deste passo, você terá uma pasta chamada apache-tomcat-X.X.XX (onde X.X.XX é o número da versão do tomcat que você baixou, utilizamos neste curso a versão apache-tomcat-8.5.24) dentro da pasta “desen”. Tenha cuidado para

não baixar a versão 8.0.X do tomcat pois essa já está com seu fim de vida anunciado e a versão 9.0.X ainda está na versão beta (versão de testes), por isso estaremos utilizando a versão mais estável mais recente que é a 8.5.X do Tomcat.

Pronto! Viu como foi fácil ter o ambiente mínimo necessário para executar suas futuras aulas? Porém, o nosso objetivo é fazer com que você use um ambiente de desenvolvimento integrado e que facilite a construção e testes das suas aplicações; então, para isso, vamos instalar e configurar o Eclipse. A primeira coisa a fazer é baixá-lo. O arquivo para download poderá ser encontrado aqui: <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/oxygen2>.

Depois de baixá-lo, você terá que descompactar o arquivo e, mais uma vez, sugerimos que a pasta destino seja "Documentos\desen". Quando o processo de descompactação terminar, você terá o diretório "eclipse" dentro dessa pasta (assim como o diretório "apache-tomcat-8.5.24 criado anteriormente).



Vídeo 02 - Obtendo Aplicativos

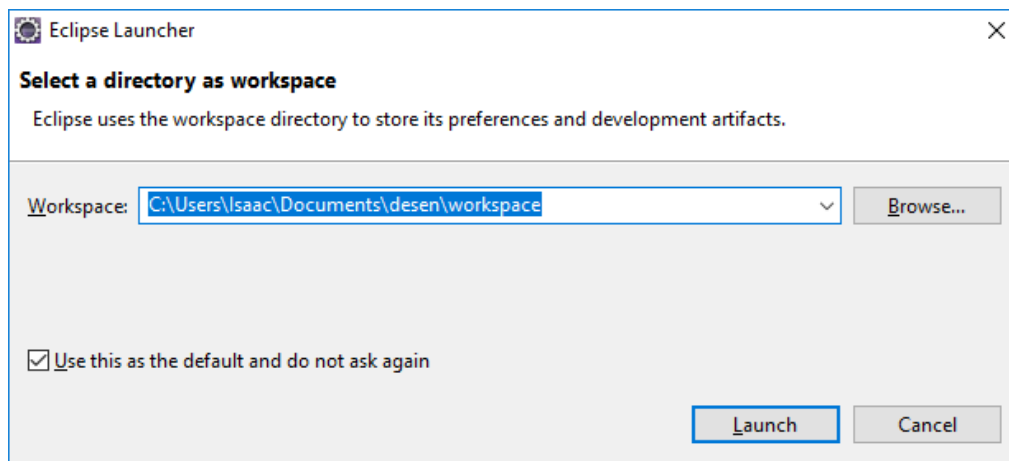
Essa é a versão do Eclipse que é utilizada como base para o curso, mas já existem versões mais recentes que, provavelmente, também funcionam para nossas práticas. Várias versões de Eclipse para Desenvolvimento Web (JEE) podem ser encontradas em: <http://www.eclipse.org/downloads/>.

Configurando nosso ambiente

Bom, agora tudo foi instalado, falta apenas configurar nosso ambiente. Para isso, execute o arquivo eclipse.exe que está em Documentos\eclipse. É interessante criar um atalho na área de trabalho para este arquivo, pois este ambiente será utilizado durante todo o curso. Assim que você executar esse arquivo, uma janela como a mostrada na Figura 1 será exibida pedindo que você escolha o seu workspace (pasta de trabalho padrão para os projetos Java). Neste caso, vamos criar

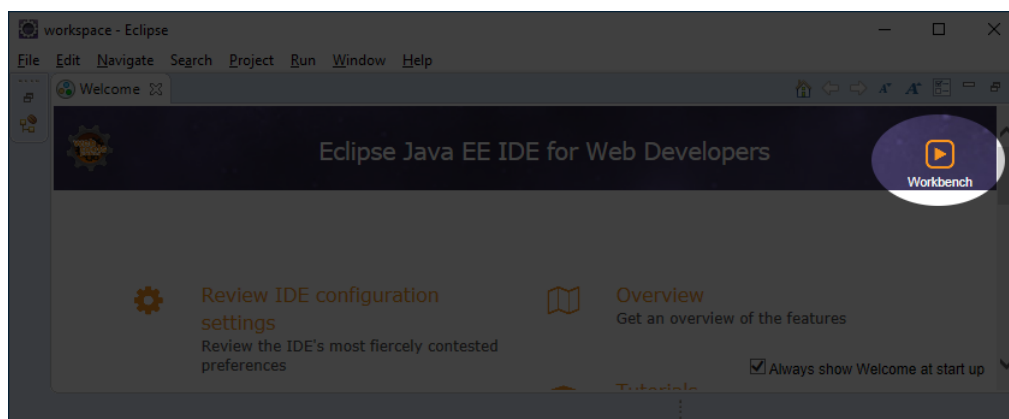
um diretório chamado workspace em Documentos\desen e escolhê-lo. Se quiser, marque a opção “Use this as the default and do not ask again” para que essa informação não seja perguntada novamente, toda vez que o Eclipse for iniciado.

Figura 01 - Selecionando a pasta de trabalho para o Eclipse



Feito isso, a próxima janela a ser exibida é mostrada na Figura 2. Nela você encontrará algumas opções para quem é iniciante, porém vamos direto para a opção *workbench* (opção indicada figura abaixo), pois essas opções podem ser exibidas a qualquer momento.

Figura 02 - Tela inicial do *Eclipse*



Vídeo 03 - Configurando

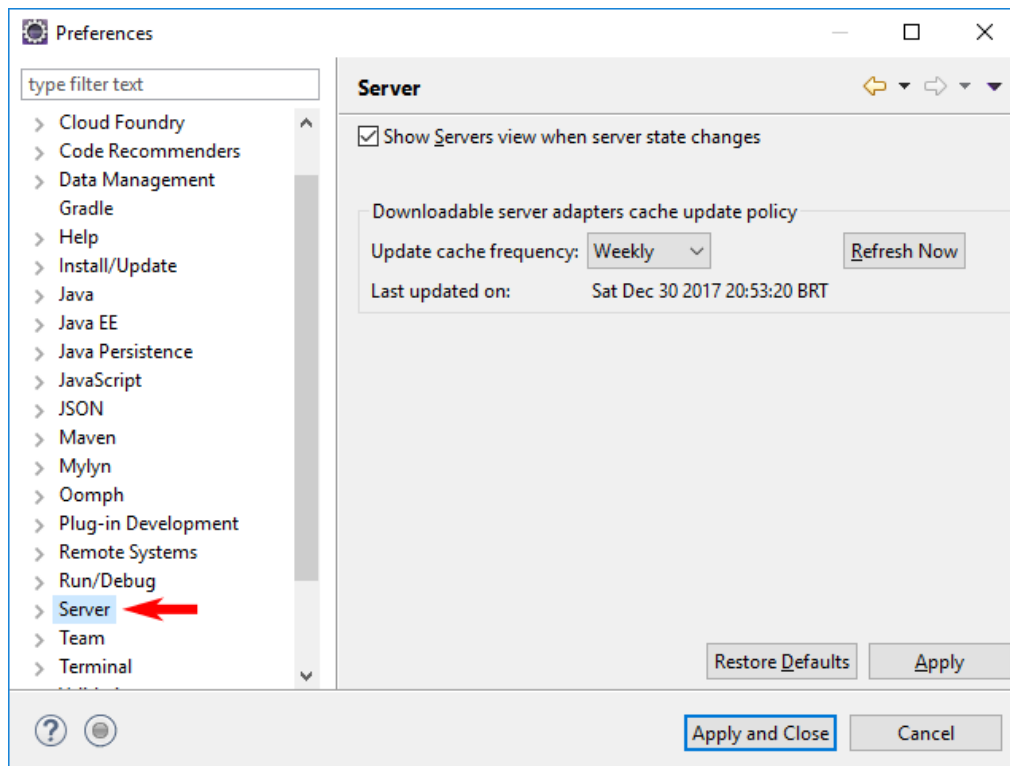
Pronto, agora estamos vendo o nosso ambiente de desenvolvimento que, de agora em diante, nos acompanhará na execução das nossas atividades.

Bom, antes de instalarmos o *Eclipse*, nós instalamos o JDK e o Apache Tomcat. E agora, como integrar o JDK e o Apache Tomcat com o *Eclipse*? Com relação ao JDK, não precisamos nos preocupar, pois o *Eclipse*, ao ser executado, descobre através do sistema operacional onde o JDK foi instalado, então nenhuma configuração é necessária.

E o *Apache Tomcat*? Bom, A versão do *Eclipse* que nós baixamos e instalamos é uma versão para desenvolvedores JEE - desenvolvimento web, entre outras coisas - e nesta versão ela vem com um plugin chamado Web Tools Platform (WTP). Esse plugin permite a integração do *Eclipse* com a maioria dos servidores JEE do mercado, entre eles o *Apache Tomcat*. Para que essa integração funcione, temos que informar ao plugin onde o Apache Tomcat foi instalado, então vamos lá:

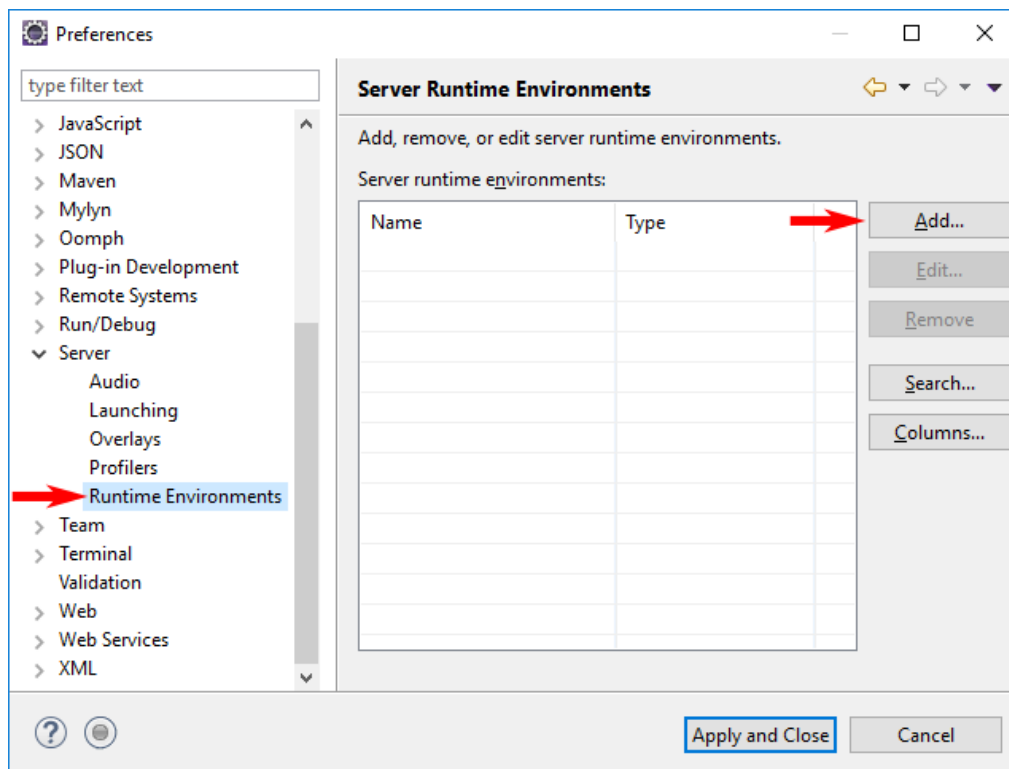
No menu do *Eclipse* selecione a opção **Window / Preferences**. Uma janela como a da Figura 3 será exibida. Clique em **Server**, como indicado pela seta na figura.

Figura 03 - Selecionar opção *Server*



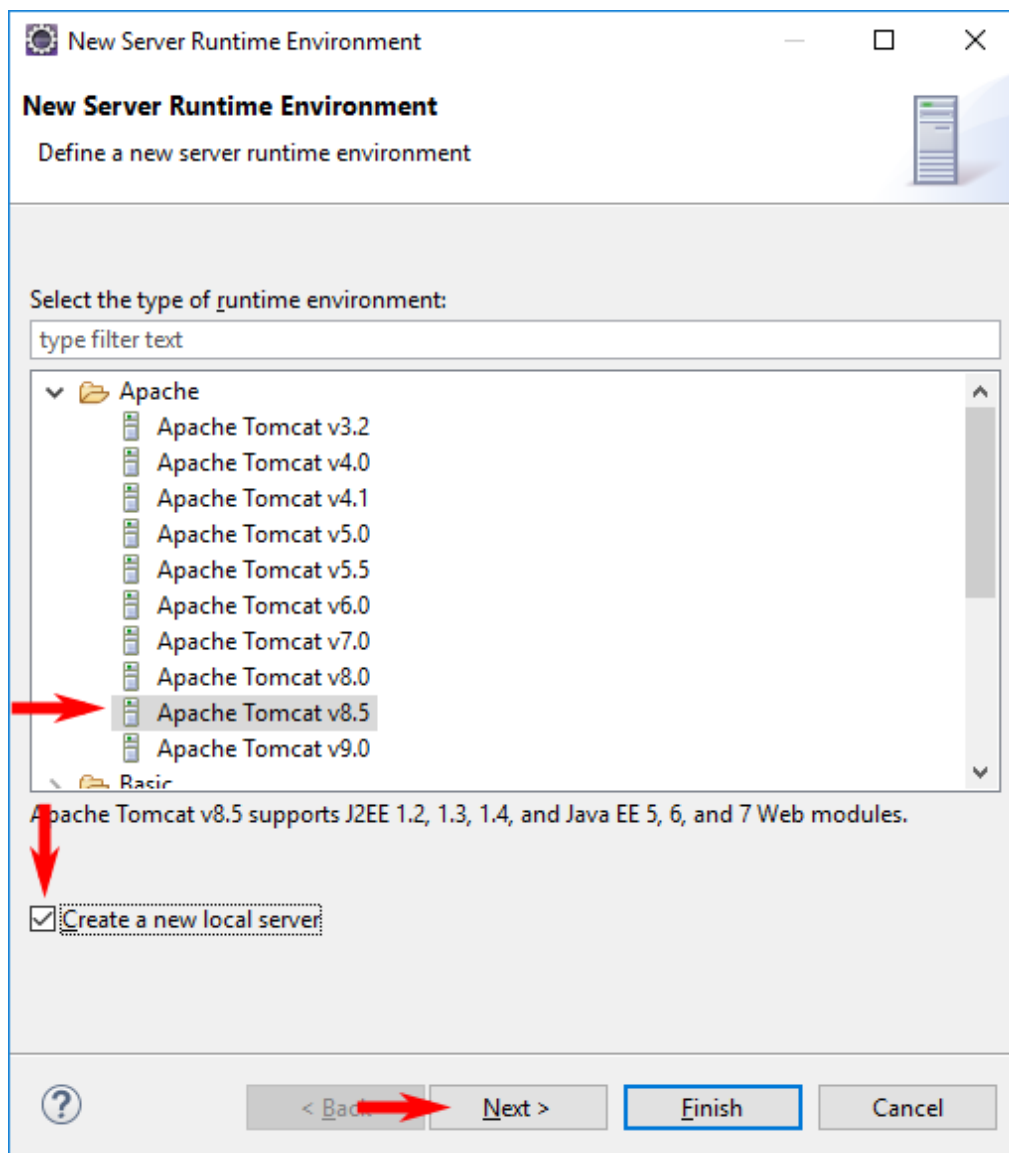
Ao fazer isso, as opções de servidores serão exibidas. Clique em *Runtime Environment* e em seguida no botão "**Add...**", como mostrado pelas setas na Figura 4.

Figura 04 - Selecionar opção *Runtime Environment* e depois clicar no botão "**Add...**".



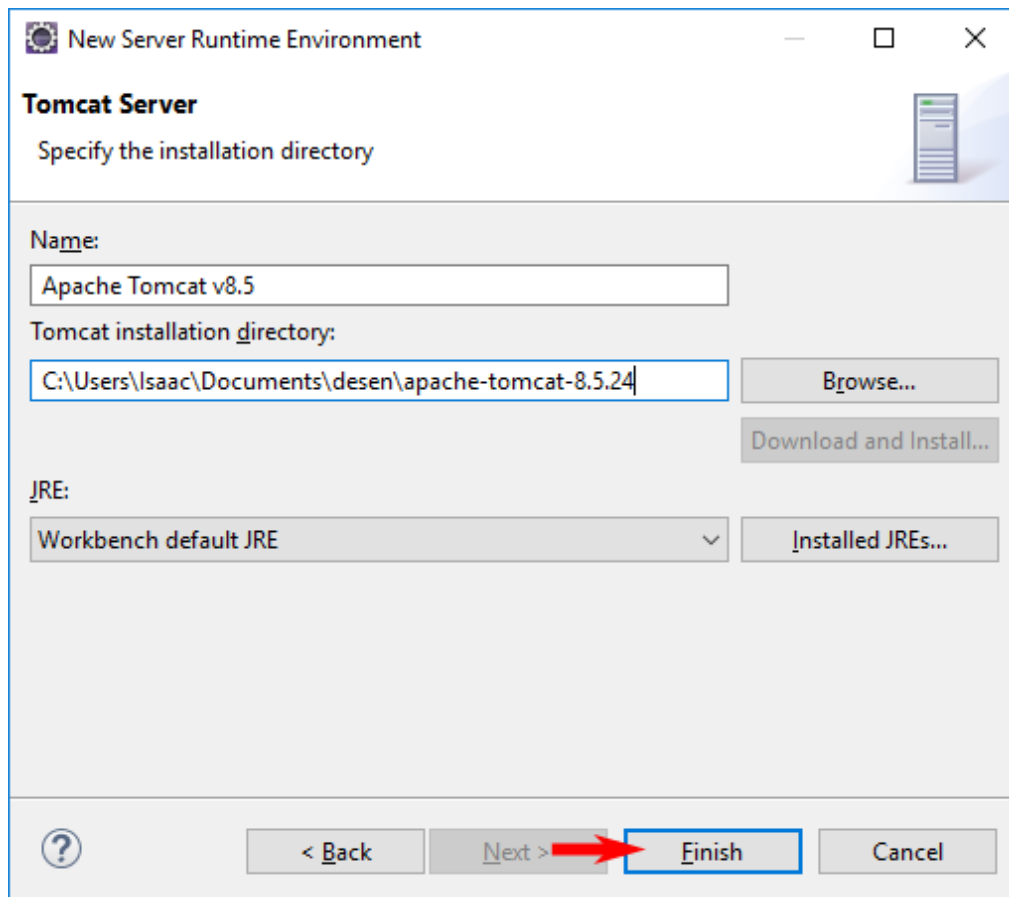
Uma nova janela será exibida, como mostrado na Figura 5. Selecione a opção Apache, e em seguida a opção **Apache Tomcat v8.5**. Marque o checkbox **Create a new local server** para termos acesso ao servidor mais facilmente via o Eclipse. Por fim, clique na opção Next, como indicado pelas setas na Figura 5.

Figura 05 - Selecionar a versão do *Tomcat* utilizada e a opção “*Create a new local server*”



Na próxima janela a ser aberta (Figura 6), clique no botão “Browse...”, selecione o diretório Documentos\desen\apache-tomcat-X.X.XX e depois clique no botão Finish, como indicado pelas setas da Figura 6.

Figura 06 - Indicando a localização da instalação do *Apache Tomcat*



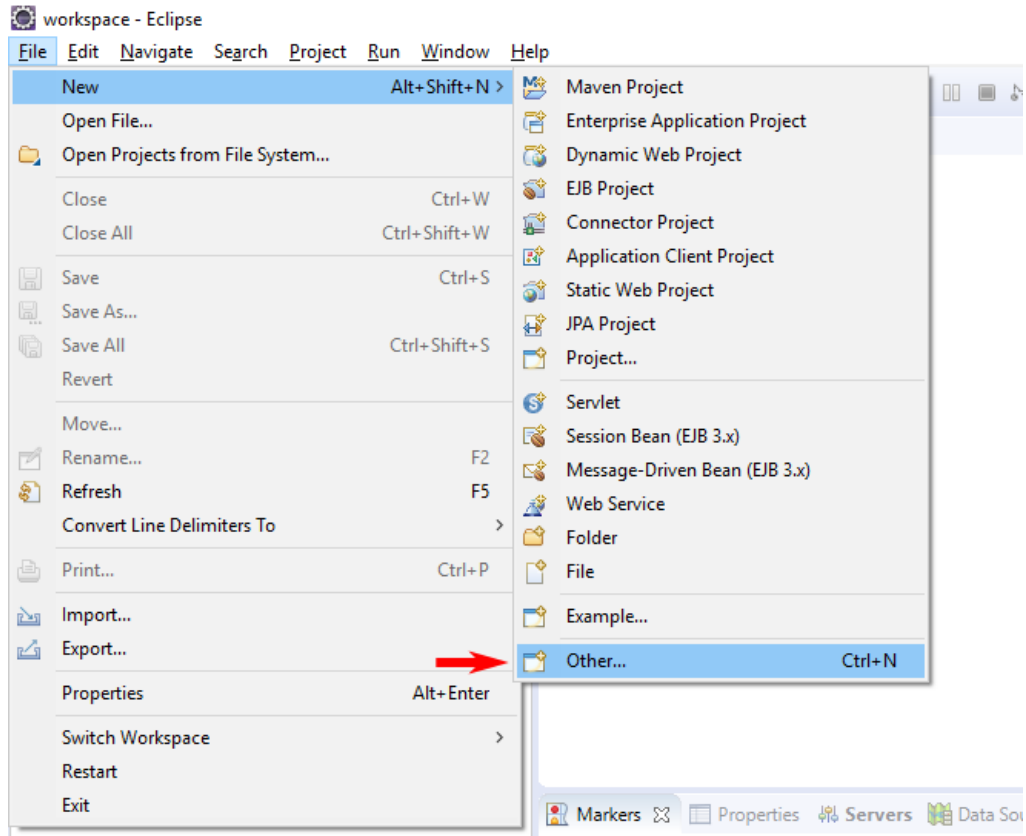
Para concluir a integração, confirme tudo clicando no botão OK.

Agora temos o Eclipse integrado com o *Apache Tomcat* e, a partir de agora, quando criarmos nossos projetos web já poderemos informar ao Eclipse que estes serão executados no *Apache Tomcat* previamente instalado. Além disso, como veremos mais a frente, o plugin do WTP nos permite executar o *Apache Tomcat* de dentro do Eclipse em dois modos de execução diferentes. O primeiro é o convencional, mais rápido, e o segundo permite fazer a depuração (execução comando por comando) do código do programa. Não é legal?

Criando projetos de sistemas Java para Web

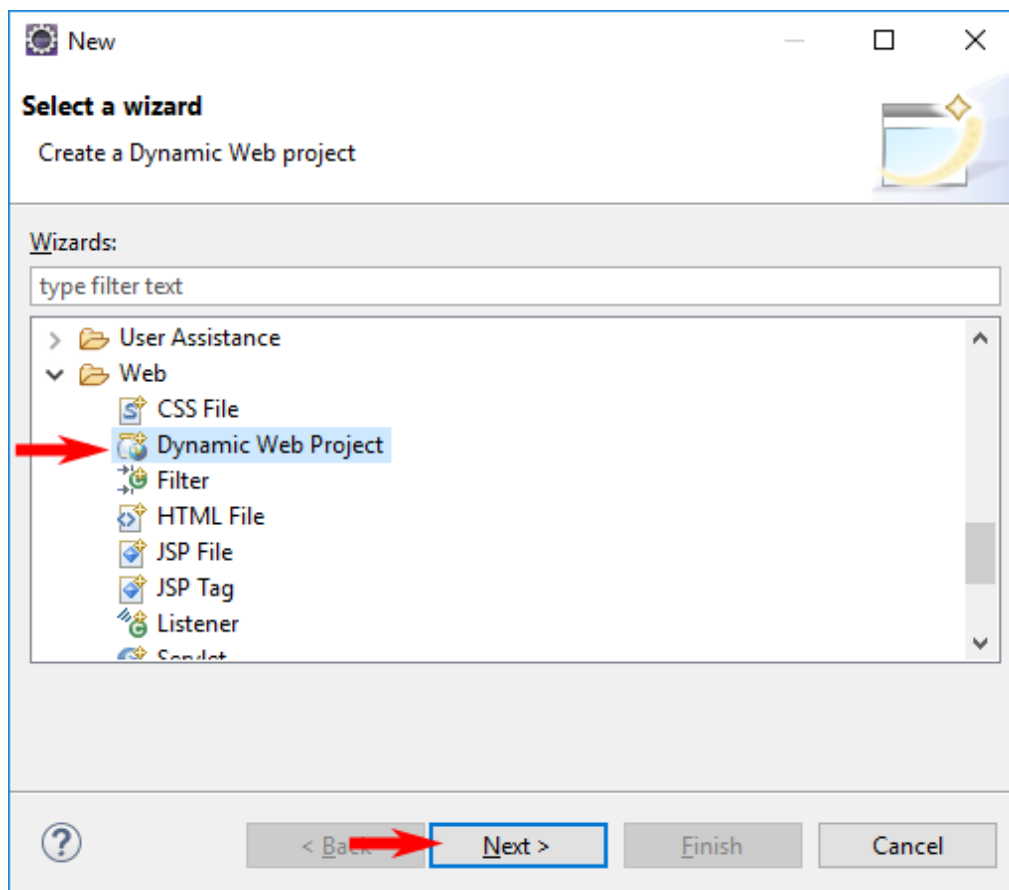
Bom, agora vamos criar o nosso primeiro projeto web usando o *Eclipse*. Esse projeto será criado e integrado com o *Apache Tomcat* que nós configuramos na seção anterior. Usando o menu do *Eclipse* selecione **File / New / Other...**, como mostrado na Figura 7.

Figura 07 - Acionando a opção de menu *File/New/Others...*



Na tela seguinte, selecione dentro da pasta **Web** a opção *Dynamic Web Project*, como mostrado na Figura 8. Em seguida clique no botão *Next*.

Figura 08 - Criando um novo projeto de sistema web



Na tela seguinte (Figura 9), defina o nome do projeto como "aula" e em seguida clique em *Next*.

Figura 09 - Nomeando o projeto

New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project location

☒ Use default location

Location:

Target runtime

Dynamic web module version

Configuration

A good starting point for working with Apache Tomcat v8.5 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

EAR project name:

Working sets

☐ Add project to working sets

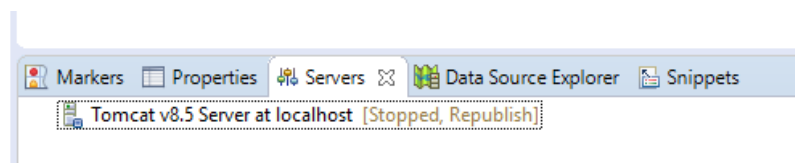
Working sets:

Nas telas seguintes, observe que existem várias opções de configuração, porém as mesmas podem ficar com seus valores padrão, com exceção da última tela onde o campo 'Generate web.xml deployment descriptor' que deve ser marcado. Caso não esteja, marque-o antes de finalizar com o botão **Finish**.

Agora vamos dizer que a implantação (*deploy*) e execução dessa aplicação será feita no nosso servidor Apache Tomcat já configurado. Para isso, execute os seguintes passos:

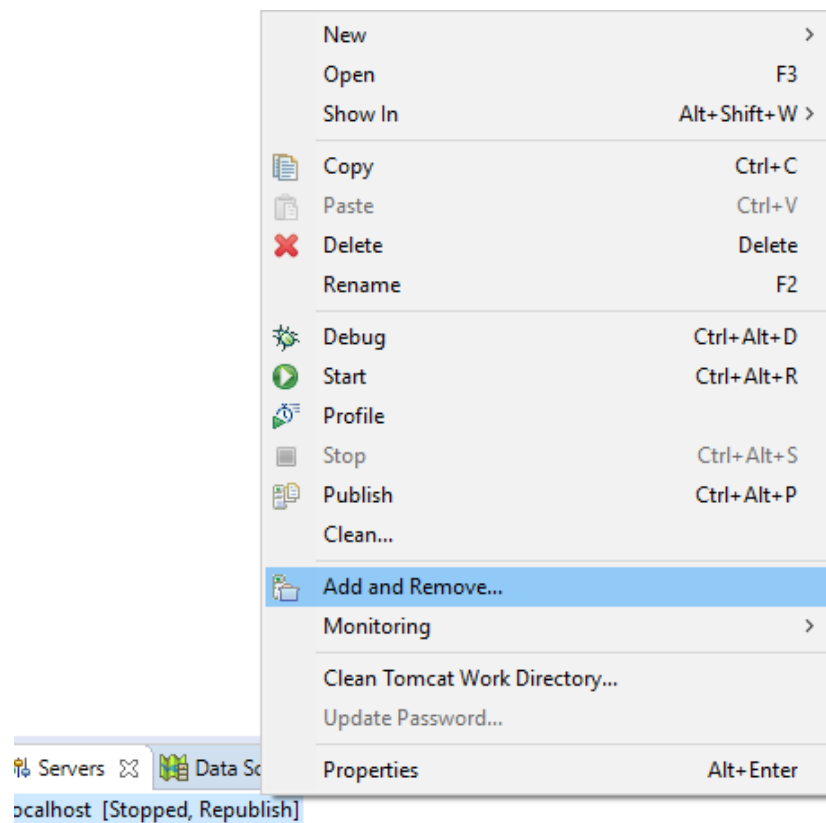
- Clique na visão *Servers*, como destacado na Figura 10. As visões são áreas do *Eclipse* que mostram informações específicas relacionadas a um determinado assunto. No caso da visão *Servers*, ela apresenta informações sobre os servidores web instalados localmente no *Eclipse*.
- Em seguida, com o botão direito do mouse, clique em Tomcat v8.5 Server at localhost [Stopped], como mostrado pela seta da Figura 10.

Figura 10 - Visão *Servers* indica situação dos servidores web instalados no Eclipse.



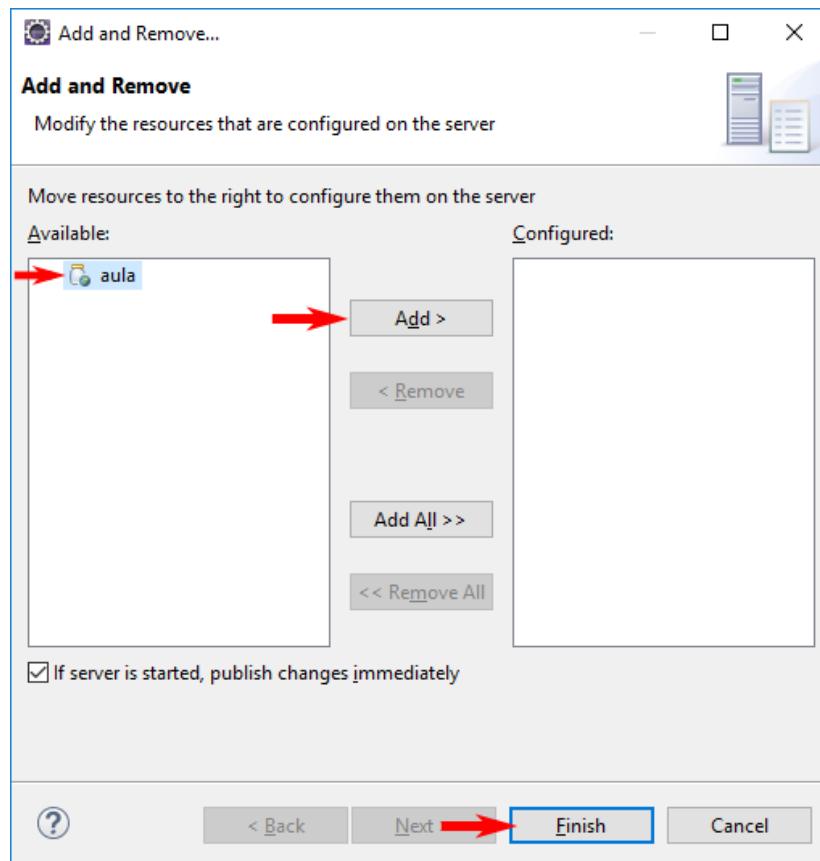
- Na janela que se abrirá, selecione a opção *Add and Remove...*, como mostrado pela seta da Figura 11.

Figura 11 - Selecionando a opção para gerenciamento de projetos associados ao servidor web



- Na janela seguinte (Figura 12), iremos selecionar quais projetos queremos instalar no servidor web. Selecione o nosso projeto “aula” e clique no botão “Add >”. Isso fará com que o projeto passe do lado esquerdo (*available*) para o lado direito (*configured*), indicando que ele ficará associado ao servidor web do *Tomcat*. Por fim, clique no botão *Finish* para finalizar a operação.

Figura 12 - Associando o projeto ao servidor web



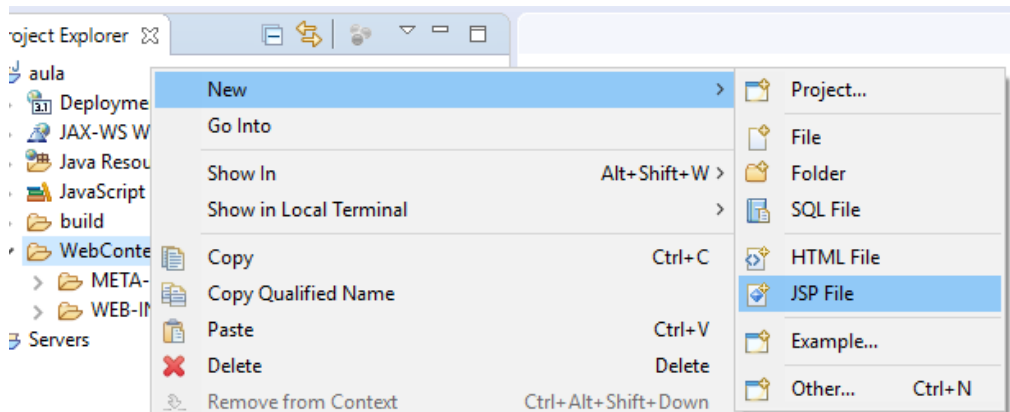
E finalmente temos nossa aplicação configurada para rodar no *Tomcat*. Na verdade precisamos criar nossa aplicação, certo?

Escrevendo o código de uma aplicação Web simples

Vamos, então, criar uma aplicação simples para testar nosso projeto funcionando no *Apache Tomcat*. Essa aplicação simples consistirá de um arquivo de nome `index.jsp` a ser criado na pasta Web do projeto, a qual é mostrada no Eclipse com o nome *WebContent*. Essa pasta representa a raiz do nosso site (aplicação web).

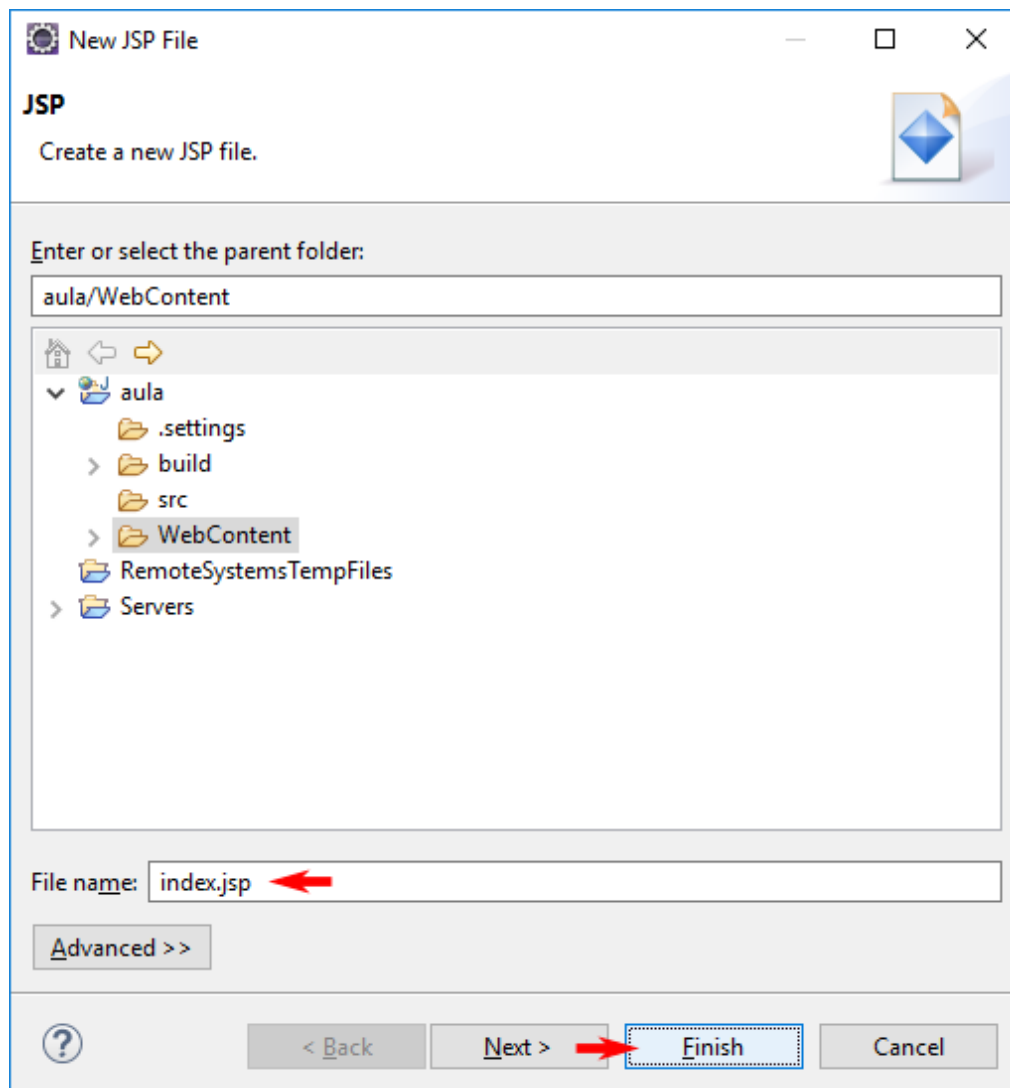
Para isso, clique com o botão direito na pasta *WebContent* do projeto “aula” e escolha a opção **New / JSP File**, como mostrado na Figura 13. Você estudará em uma aula mais a frente sobre arquivos JSP, mas por enquanto basta saber que ele pode ser visto como um arquivo HTML especial.

Figura 13 - Criando um arquivo JSP no projeto



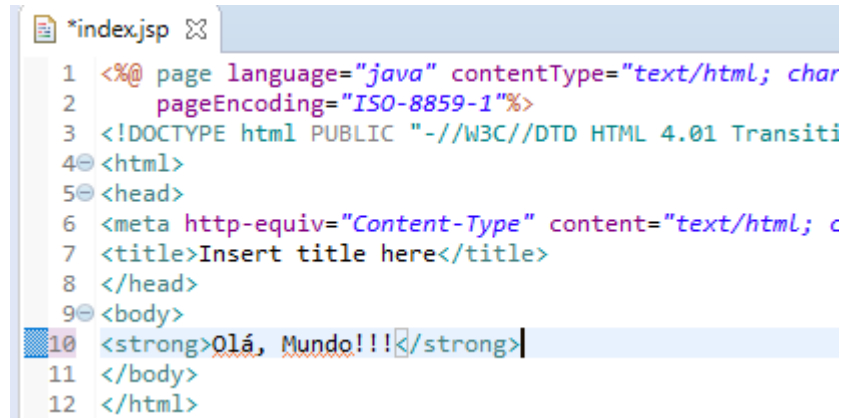
Na janela que será exibida digite no campo **File name** o nome index.jsp e em seguida clique no botão **Finish**, como mostrado pelas setas da Figura 14.

Figura 14 - Indicando o nome e localização do arquivo JSP



Será criado e aberto para edição um novo arquivo JSP. Nesse arquivo digite dentro das tags o texto Olá, Mundo!!!, como mostrado na Figura 15 – Note que a maior parte do texto do arquivo index.jsp é HTML puro, que você já tem conhecimento.

Figura 15 - Conteúdo do arquivo index.jsp

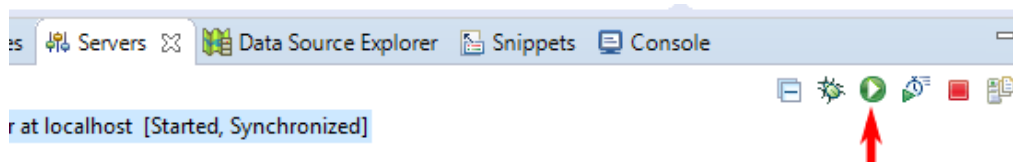


```
*index.jsp
1 <%@ page language="java" contentType="text/html; char
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transiti
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; c
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <strong>Olá, Mundo!!!</strong>
11 </body>
12 </html>
```

Iniciando o servidor Web e rodando a aplicação

Agora vamos colocar nosso servidor para executar. Novamente clique na visão **Servers** e em seguida no botão **start** indicado por uma seta na Figura 16.

Figura 16 - Inicializando o servidor web

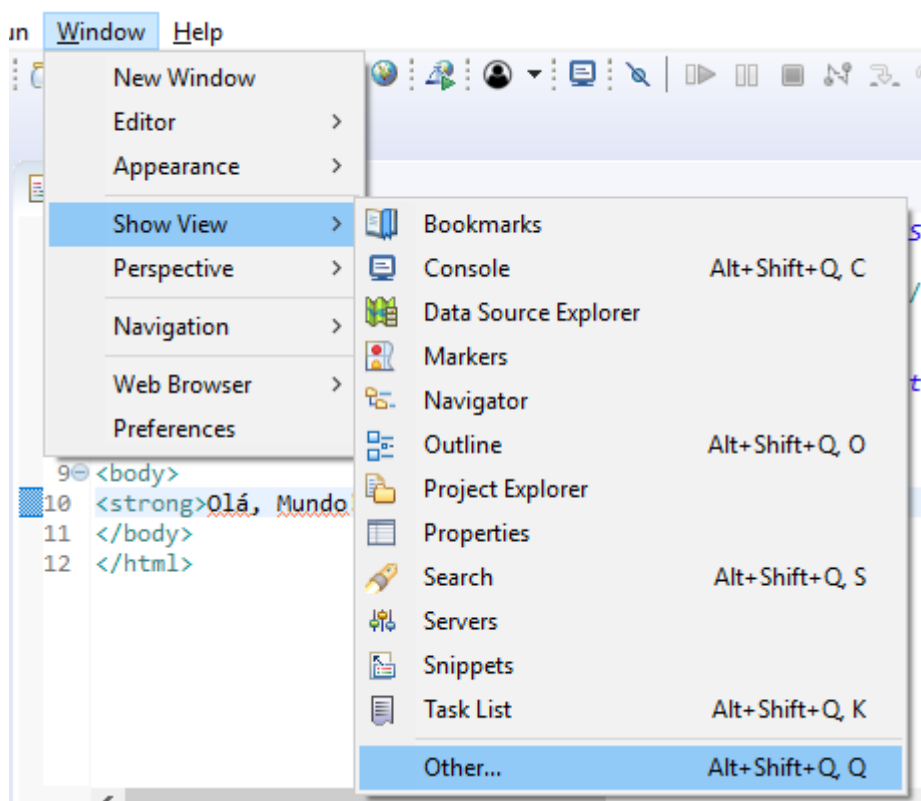


Na visão **Servers** da Figura 16 existe uma segunda opção para iniciar o servidor, que é a execução no modo de depuração (*debug*), a qual nos permite depurar (executar comando por comando) nossas classes e arquivos JSPs. Não é legal? Entretanto, essa opção só deve ser usada quando realmente queremos depurar o código para encontrar algum erro ou para entender o que está acontecendo durante a execução, pois ela deixa o Eclipse um pouco mais lento. Para iniciar o *Tomcat* no modo de depuração você deve clicar no botão ao lado do apontado pela seta da Figura 16 (aquele que parece um inseto – *bug*).

Ao iniciar a execução do servidor, seja no modo normal de execução ou no modo de depuração, irá aparecer na visão Servers um botão vermelho utilizado para você parar a execução do *Tomcat*.

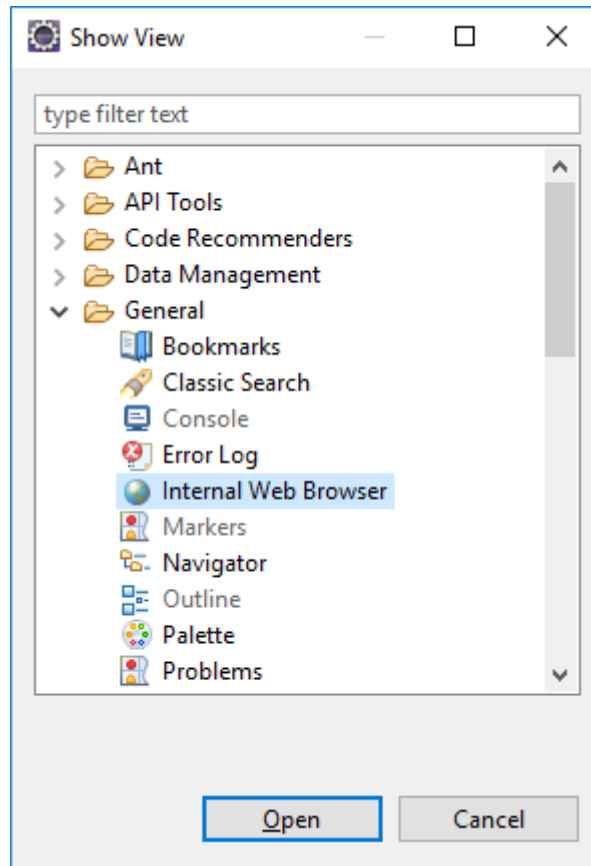
Agora vamos ver como ficou a página criada para nosso sistema web. Para isso, selecione **Window / Show view / Other**, conforme mostrado na Figura 17.

Figura 17 - Acessando o nosso sistema web



Na janela a ser aberta (ver Figura 18), selecione a pasta **General**, opção **Internal Web Browser**.

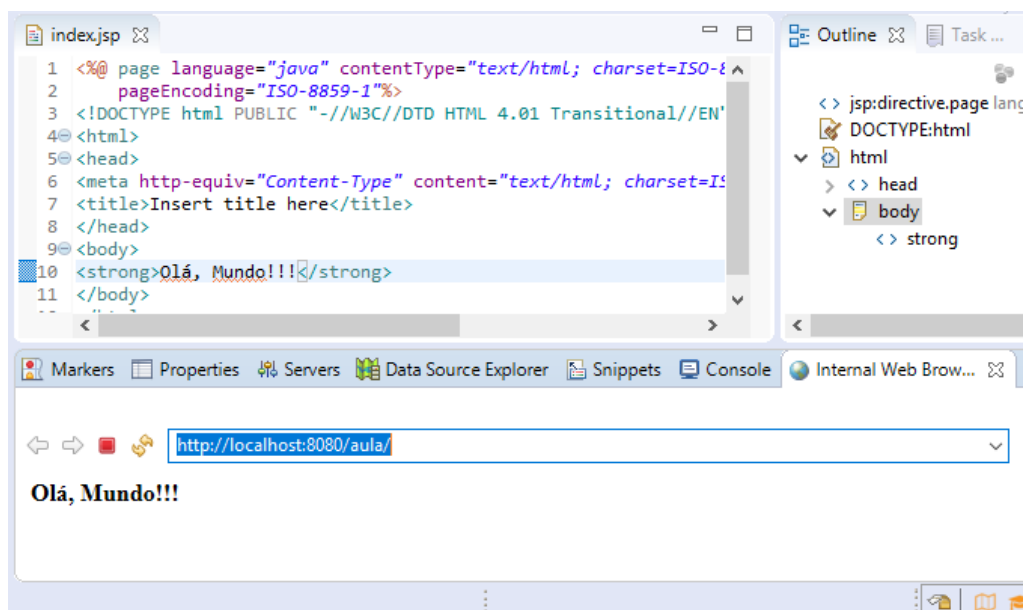
Figura 18 - Abrindo o navegador web do Eclipse



Na visão **Internal Web Browser** que será aberta (parte inferior da Figura 19), digite a URL **<http://localhost:8080/aula>** e pressione a tecla enter. Isso fará com que a página web representada pelo arquivo JSP seja carregada e apresentada, como mostrado na Figura 19. Isso porque ao colocar a URL **<http://localhost:8080/aula>**, estamos solicitando que o navegador web acesse o próprio computador (*localhost*), usando a porta 8080 do computador (porta padrão de instalação do *Tomcat*), acessando o conteúdo de /aula (pasta WebContent do projeto “aula” criado no *Eclipse*). O conteúdo do arquivo index.jsp será mostrado porque, quando não indicado um nome do arquivo na URL, o servidor por padrão busca por arquivos como index.html e index.jsp. Da mesma forma, você pode copiar a URL (**http://localhost:8080/aula**) e colocar no seu navegador preferido (Google Chrome, Firefox, etc) que a página será exibida.

Você pode acessar a página web criada através de qualquer navegador, fora do Eclipse. Para isso, você pode copiar a URL (<http://localhost:8080/aula>) e colocar no seu navegador preferido (Google Chrome, Firefox, etc) que a página será exibida.

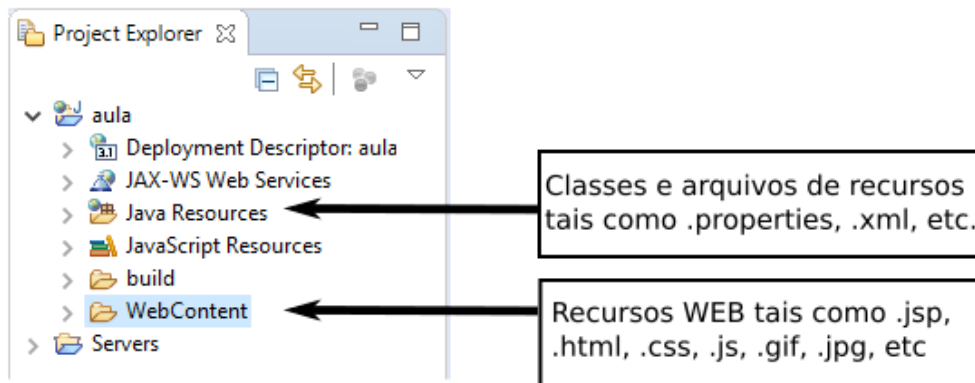
Figura 19 - Acessando o sistema web criado pelo projeto “aula”



Viu como foi fácil?!

Por último, vamos dar uma olhada na estrutura do projeto que foi criado (ver Figura 20). Perceba que toda a estrutura da aplicação web foi criada pelo plugin do WTP, inclusive o descritor de implantação (deploy) web.xml, cuja função você estudará em uma aula mais à frente. Para seguirmos incrementando nossa aplicação, você verá, na próxima aula, que criaremos os chamados Servlets na pasta Java Resources / src. Os nossos arquivos JSPs, imagens, css, js, continuarão sendo criados dentro da pasta WebContent, que é a pasta raiz da nossa aplicação web. Perceba também que existe um item no nosso projeto chamado Deployment Descriptor. Ao clicar nele automaticamente será aberto o arquivo web.xml.

Figura 20 - Estrutura de pastas do projeto criado no Eclipse



Pronto, concluímos a configuração do Eclipse, Apache Tomcat, JDK e do nosso ambiente de desenvolvimento. Muito fácil, não é?

Atividade 01

1. Execute os passos de instalação e configuração das ferramentas *Eclipse* e *Apache Tomcat*, além do JDK, se necessário, conforme mostrado em aula. Reporte se você encontrou algum problema.
2. Crie o projeto aula, execute o servidor e acesse-o, conforme mostrado em aula. Reporte se você encontrou algum problema.
3. No projeto aula crie um novo arquivo JSP chamado de novo.jsp, de forma similar ao arquivo criado index.jsp, colocando agora uma mensagem pessoal. Acesse o novo arquivo via browser (ex: <http://localhost:8080/aula/novo.jsp>) e verifique se a sua mensagem pessoal está sendo mostrada corretamente.
4. Repita o passo anterior e confirme que o mesmo procedimento da questão anterior pode ser feito tanto no navegador do Eclipse como em navegadores web externos, como o Internet Explorer ou o Mozilla Firefox.

Leitura Complementar

Acesse os sites das ferramentas Eclipse <<http://www.eclipse.org>> e Tomcat <<http://tomcat.apache.org/>> e procure passar a vista em sua documentação (visão geral, tutoriais introdutórios etc.), de forma a complementar seu entendimento sobre essas ferramentas.

Resumo

A aula de hoje foi introdutória ao ambiente de trabalho que você irá utilizar durante o curso. Você aprendeu que para montar o nosso ambiente de desenvolvimento web é necessário instalar uma IDE (Eclipse), um servidor web (Apache Tomcat), além do próprio JDK. Além disso, é necessário configurar o Eclipse indicando o caminho de instalação do servidor web a ser utilizado. Com isso, projetos web dinâmicos podem ser criados e configurados para serem executados através do servidor instalado. Muito bem, uma vez dominada a instalação e configuração do ambiente de desenvolvimento web, já podemos avançar para as nossas próximas atividades, construindo servlets e arquivos JSPs dentro de projetos criados no Eclipse. Esse será o assunto das nossas próximas aulas.

Autoavaliação

1. Descreva os passos necessários para estar com o ambiente Eclipse + Apache Tomcat preparado para iniciar a programação web.
2. Descreva os passos para se criar e executar um projeto web simples usando essas ferramentas.

Referências

AHMED, K. Z.; UMRYSH, C. E. **Desenvolvendo aplicações comerciais em Java com Java J2EE e UML**. Rio de Janeiro: Editora: Ciência Moderna, 2003. 324 p.

BASHAM, Bryan; SIERRA, Kathy; BATES, Bert. **Head First Servlets and JSP: passing the Sun Certified Web Component Developer Exam (SCWCD)**. 2nd ed. O'Reilly Media, 2008.

CATTELL, Rick; INSCORE, Jim. **J2EE: criando aplicações comerciais**. Rio de Janeiro: Editora Campus, 2001.

HALL, Marty. **More Servlets and JavaServer Pages (JSP)**. New Jersey: Prentice Hall PTR (InformIT), Sun Microsystems Core Series, 2001. 752p. Disponível em: <<http://pdf.moreservlets.com/>>. Acesso em: 11 maio 2012.

HALL, Marty; BROWN, Larry. **Core Servlets and JavaServer Pages (JSP)**. 2nd ed. New Jersey: Prentice Hall PTR (InformIT), Sun Microsystems Core Series, 2003. 736 p. (Core Technologies, 1). Disponível em: <<http://pdf.coreservlets.com/>>. Acesso em: 11 maio 2012. HYPERTEXT Transfer Protocol -- HTTP/1.1: methods definition. Disponível em: <<http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>>. Acesso em: 11 maio 2012

J2EE Tutorial. Disponível em: <<http://java.sun.com/javaee/reference/tutorials/>>. Acesso em: 11 maio 2012.