

Transcrição

Nesta primeira aula aprenderemos como preparar o ambiente: o que precisamos instalar e configurar para as aulas serem acompanhadas. Deveremos ter na máquina três itens: o Java propriamente, JRI ou JDK, a segunda opção possui mais documentação, mas para o nosso objetivo neste curso o JRI é o suficiente.

Usaremos a versão Java 10, mas a 8, 9 também é o suficiente. Talvez quando você esteja realizando esse curso já existam versões mais recentes, como 11 ou 12, não há problema algum em utilizá-las pois o Java trabalha com servidor e as versões recentes sempre funcionam.

Observe a versão instalada na linha de comando:

```
C:\Users\Alura>java -version
java version "10.0.2" 2018-07-17
Java(TM) SE Runtime Environment 18.3 (build 10.0.2+13)
Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.2+12,mixed mode)
```

Não usaremos o Java na linha de comando, então não há necessidade de utilizar quaisquer comandos por essa via, nosso trabalho se dará no Eclipse, portanto é necessário que ele esteja instalado.

A [versão do Eclipse](#) necessária para esse curso mais avançado é a mais recente, pois ela apresenta alguns recursos a mais e que diferente da versão inicial. Atualmente, quando realizamos o download o Eclipse versão Photon, ele vem com um instalador a ser executado. A diferença dessa versão para as anteriores, é precisamos do recurso Java EE, um pacote de ferramentas como EE,JPA,JSF,Mylyn e outras para criação de aplicações web.

No momento de abrir o Eclipse teremos a opção "Eclipse IDE for Java EE Developers", é essa que iremos escolher. Em seguida, iremos configurar o workspace, isto é, a pasta em que armazenaremos nossos projetos.

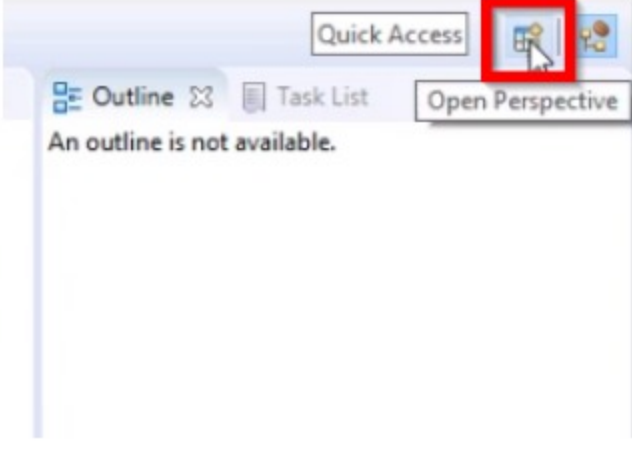
O último item que precisamos instalar é o o servidor, em nosso caso usaremos o [Tomcat](#), de longe o mais famoso no mundo Java. As versões do Tomcat são mais estáveis, ainda que ele seja atualizado com alguma frequência. Usaremos a versão 9.0.11, que até o momento é a mais recente, mas podemos ainda utilizar a versão 7 ou 8 sem problemas. O que de fato importa é que a versão possa dar suporte para o Servlet 3.0 em diante.

Lembrando que o ideal é que você utilize as mesmas versões apresentadas no curso, para que tenhamos o máximo de sincronia e possamos desenvolver o projeto sem problemas.

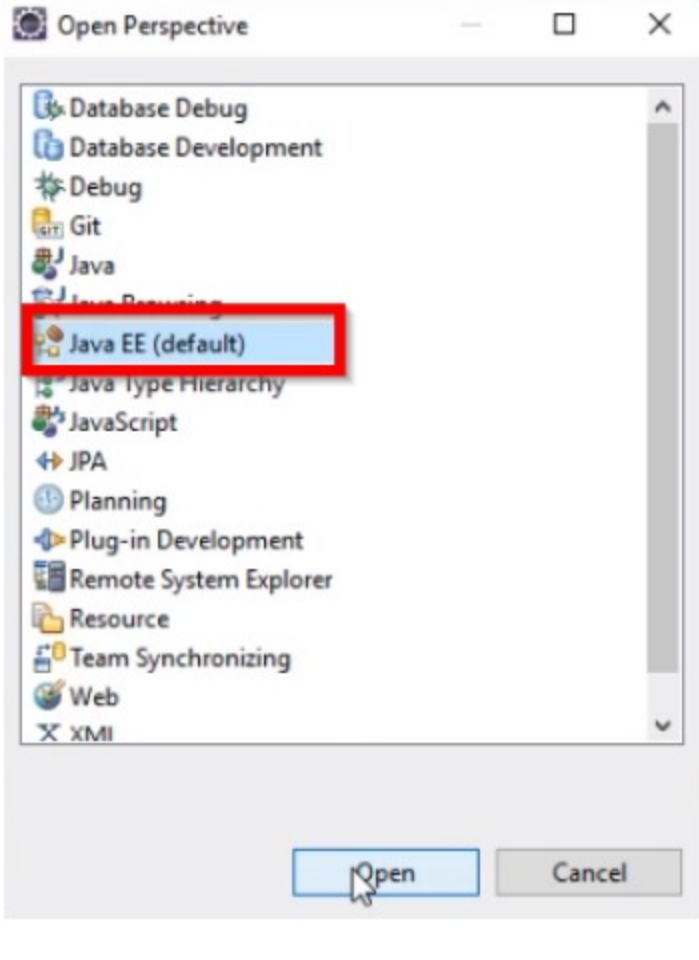
No site do [Tomcat](#) você encontrará a versão para ser baixada em zip, e para os usuários de Linuxs haverá disponível o arquivo tar.gz. Trata-se de um arquivo pequeno, cerca de 10.268 KB. Depois de realizar o download do arquivo, basta extrai-lo.

Em seguida, precisaremos associar o Eclipse ao Tomcat, afinal não queremos inicializá-lo via linha de comando, pois no dia a dia utilizamos o Eclipse, isto é, seu ambiente de desenvolvimento.

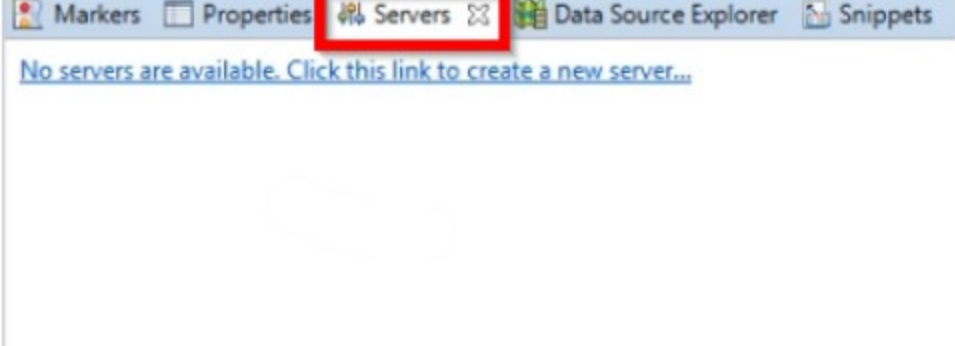
Com o Eclipse aberto, selecionaremos o ícone "Open Perspective" na parte superior direita da tela.



Na nova janela, selecionaremos a opção "Java EE (default)".



A próxima etapa é configurar o Tomcat com o Eclipse. Por meio do "Quick Access" - buscador na parte superior direita da tela - no Eclipse, digitaremos a "Servers". Essa opção também pode ser acessada via aba "Servers".

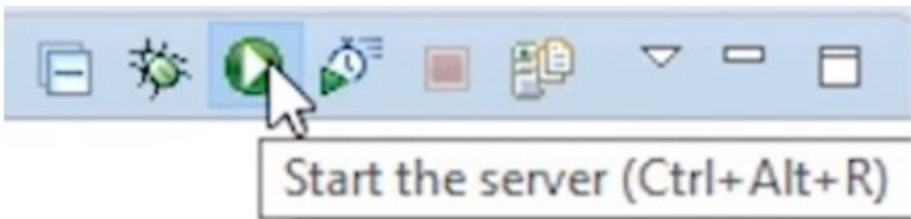


Clicaremos no link disponível na área "Servers" e começaremos a realizar a configuração. O Tomcat vem do Apache, uma organização *open source* de códigos. Na nova janela de diálogo "Define a New Server", clicaremos sobre "Apache > Tomcat v9.0 Server". Existem algumas opções editáveis, como "Server's host name", e as manteremos com o modelo padrão. Em seguida, pressionaremos o botão "Next". Teremos a opção "Tomcat installation directory", em que colocaremos o diretório em que o servidor foi salvo na máquina, neste ponto colocaremos o endereço correto da pasta raiz.

Na opção "JRE", selecionaremos "jre-10.0.2", isto é, o Eclipse já detectou que estamos utilizando o Java 10. Feito isso, pressionaremos o botão "Finish".

Desse modo, na área "Servlet" será exibida a informação `Tomcat v9.0 Server at localhost [Stopped, Republish]` . O Eclipse automaticamente criará um novo projeto para guardar as configurações do servidor, como podemos ver na área "Project Explorer", "Servers > Tomcat v9.0 Server at localhost-config". Não podemos apagar ou fechar o projeto, caso isso seja feito o Eclipse não conseguirá mais acessar as configurações.

Clicaremos sobre o ícone "Start the server" ou usaremos o atalho "Ctrl + Alt + R" para iniciar o servidor.



Ao clicarmos na aba "Console", poderemos ver a saída em vermelho. Não devemos associar essa cor com alguma mensagem de erro, podemos verificar que na última linha teremos a confirmação de que o servidor foi inicializado com sucesso:

```
INFORMAÇÕES: Server startup in 520 ms
```

Agora podemos acessar o Tomcat. Porque escolhemos trabalhar com este servidor? Não sabemos ainda o que são os tais "Servlets", mas vamos escalharecer alguns pontos: estamos programando em Java e queremos trabalhar na web, e no que diz respeito ao Tomcat, o mai importante é utilizar protocolo HTTP e HTML. Acessamos a página da [Alura](#), por exemplo, por meio do navegador e utilizando o protocolo HTTP. O que o navegador exibe ao usuário é o resultado de uma página HTML, portanto precisamos aprender mais a fundo sobre como esses sistema operam. O Tomcat sabe trabalhar muito bem com protocolo HTTP e com HTML. Vamos com calma para entender detalhadamente cada uma das ideias apresentadas.

De volta ao Eclipse, sabemos que o nosso servidor está no ar, portanto já temos um processo Java sendo executado. Dessa forma, podemos utilizar o protocolo HTTP para acessar o servidor.

Abriremos uma nova aba no navegador Chrome, mas você pode utilizar o Firefox ou qualquer outro de sua preferência. Em algum momento na configuração inicial foi explicitado o endereço do servidor, que é `http://localhost` , esse é padrão de endereço para qualquer máquina, indiferentemente do sistema operacional. Porém, se escrevermos somente essa informação, não poderemos acessar o servidor e somos levados a uma página de erro **alegando um problema de conexão**. Isso ocorreu, pois faltou explicitarmos a **porta específica**.

Teremos a porta padrão `80` , mas (`http:\\localhost:80`), mas o Tomcat não utiliza esse padrão `80` , mas sim, `8080` e precisamos explicitar essa porta neste caso.

Ao acessarmos o endereço `http:\\localhost:8080` no navegador, teremos uma nova mensagem de erro, mas com um formato diferente: não há mais um erro de conexão, pois essa conexão com o servidor foi de fato estabelecida, observemos a mensagem:

```
HTTP Status 404 - Not Found

type Status Report
Message/
Description The origin server did not find a current representation for the target resource.
Apache Tomcat/9.0.11
```

O Tomcat recebeu a chamada do navegador, e respondeu que não há nada a ser exibido. No mundo do protocolo HTTP o status 404 significa que nenhum conteúdo foi encontrado. Mesmo que acessássemos essa página por outra máquina, como por exemplo um celular, e teríamos a mesma mensagem de erro.

Como o Tomcat é executado, várias máquinas podem acessá-lo desde que ele saiba o IP dessa máquina. Conseguimos instalar e executar o Tomcat, nas próximas aulas começaremos a discutir de fato sobre Servlet e criaremos nosso projeto Java.