

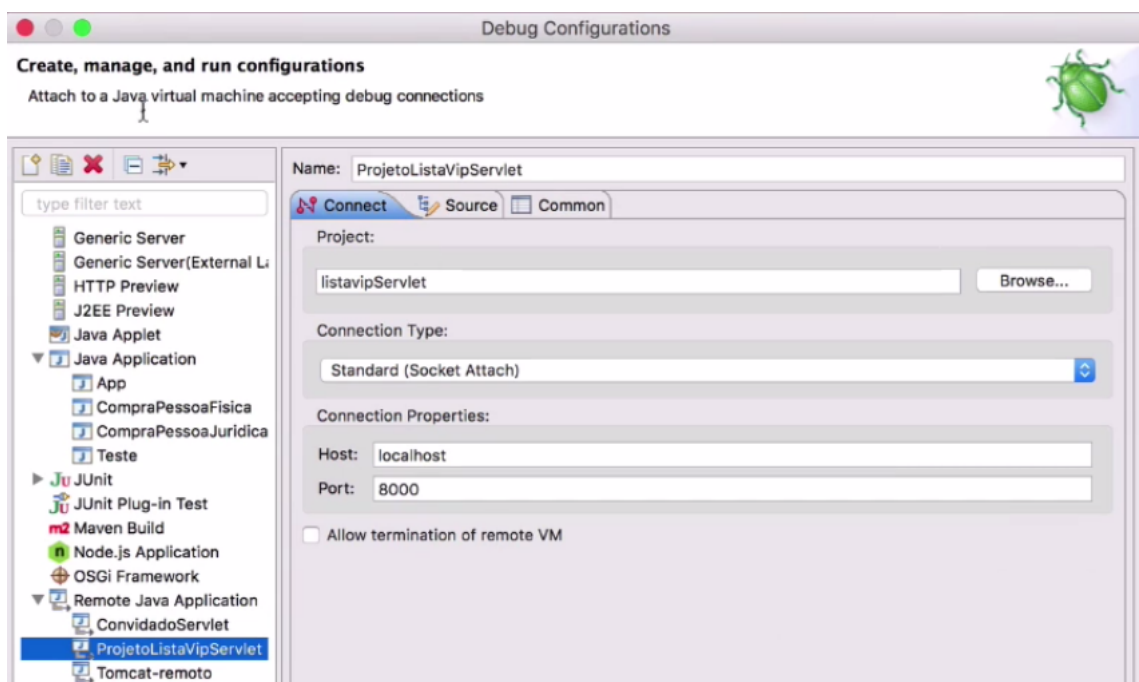
Resumo

Transcrição

Vamos revisar o que foi visto até agora: fizemos um *Debug* remoto instalando o Tomcat e colocando nele a aplicação web, simulando um servidor qualquer (da Amazon, Heroku, ou inclusive a sua própria máquina), a partir do qual fizemos *deploy* conectando esta aplicação ao Eclipse IDE para debugarmos utilizando o JPDA - lembrando que ele é configurado por meio de `./catalina.sh`, do Tomcat, editável em qualquer programa de sua preferência.

Subimos o servidor novamente, cujo container tem este JPDA configurado (`./catalina.sh jpda start`), e será liberada a porta `8000`, para então realizarmos o *Debug* remoto. Abrimos nossa aplicação no Eclipse e clicaremos no botão localizado ao lado do de *Debug*, selecionando "*Debug Configurations...*" e depois "*Remote Java Application*", criando-se nova conexão com esta nova aplicação.

Clicando com o lado direito do mouse em "*Remote Java Application*", configura-se nome, projeto, endereço de *host*, a porta em que se configurou o JPDA, entre outros. Durante a aula, criamos estas configurações:



Em seguida, colocamos a aplicação para rodar, algo que ocorre mesmo ela estando em outro local, outro *host*. Vimos também que, ao verificarmos um código legado no `ConvidadoRepository.java`, havia um *Try/Catch* inútil para capturar a exceção.

Realizamos então uma manutenção, colocando `e.printStackTrace();` para printar no log, lançando a exceção para o `servlet` e, em `ConvidadoServlet.java` a tratamos e relançamos por meio de um `servlet` de erro, o `ErrorController.java`.

Também alteramos o arquivo `web.xml` para a captura das exceções desta aplicação web, e modificamos o banco de dados para corrigir o erro e caberem mais caracteres na coluna de "nome", acrescentando-se `maxlength` na tag caso não se tenha acesso ao banco de dados. Assim, limitamos a quantidade de caracteres em `100`.

