

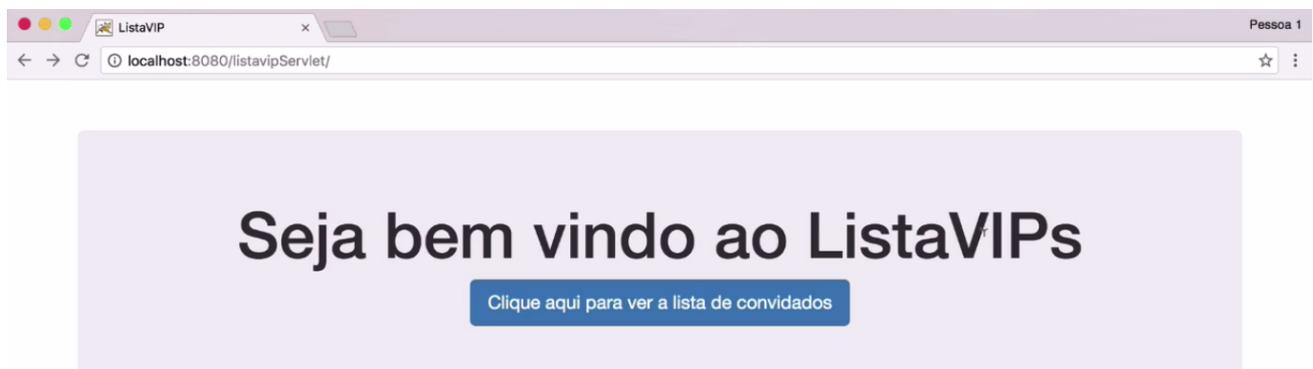
Preparando o ambiente de debug remoto

Transcrição

Agora que já debugamos uma aplicação desktop, rodando a partir do método `main()` no Eclipse IDE, podemos fazer o mesmo com aplicações web em produção. Vamos simular um Tomcat, nossa container, com esta web app, gerando seu artefato `.war` e colocando-o na container, realizando `deploy`.

Em relação a esta web app, recebemos uma reclamação de um cliente dizendo que tempo problemas ao tentar salvar um convidado. Precisaremos simular este erro para verificar o que realmente acontece na aplicação, para posteriormente encontrarmos o bug e resolvê-lo.

A aplicação é bem simples, com esta tela inicial:



Vamos clicar no botão "Clique aqui para ver a lista de convidados". Abre-se uma página contendo uma lista e um formulário de cadastro para convites. O cliente nos enviou os dados do amigo que quer convidar, a partir do qual vamos tentar simular o cadastro e descobrir onde está o erro. Nada aconteceu, sendo que o correto seria estes dados aparecem na lista logo em seguida.

Vimos que a requisição foi feita ao servidor, porém nenhum erro foi reportado pela aplicação, algo bem estranho. Inicialmente, precisaremos instalar um Tomcat na máquina para simularmos a existência de outro `host` à parte, deixando-o de um lado e o Eclipse do outro. Feito isto, debugaremos para encontrarmos o erro.

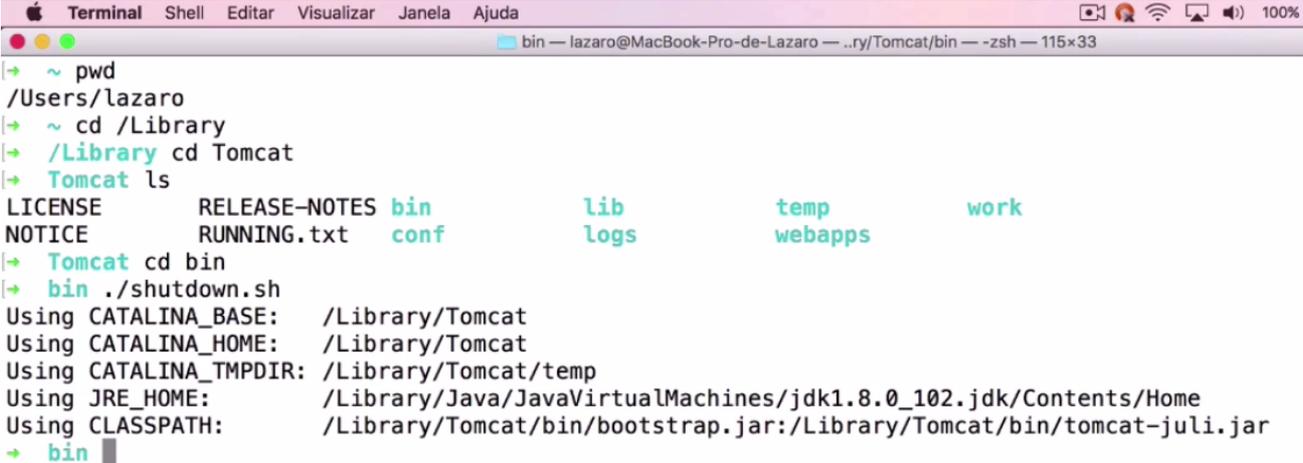
Tenho o Tomcat instalado aqui, mas disponibilizarei as instruções para instalação em todos os sistemas operacionais. Vou acessar meu programa por meio de "Macintosh HD > Biblioteca > Tomcat". Dentro da pasta do programa, temos as pastas "bin", bastante importante, com arquivos a serem rodados e capaz de pausar o servidor, a "lib", com bibliotecas do próprio container, a pasta "logs", e "webapps", onde ficam as aplicações deployadas contidas no servidor.

Abrindo a pasta "webapps", encontraremos o arquivo `listavipServlet.war`, em que se realiza o `deploy`. Após criação de um arquivo `.war`, ao subir-se o container, retira-se do `.war`, fazendo `deploy` criando uma pasta denominada "listavipServlet", mesmo nome da versão `.war`.

Para iniciarmos a simulação, precisamos pegar este artefato (`.war`) e fazer seu `deploy` no Tomcat, após o qual começa-se o processo de `Debug` propriamente dito. Primeiramente, vamos ver como se faz este `deploy`.

Queremos parar o Tomcat pelo terminal, tirar a aplicação de lá e colocar a nova, como se fosse uma atualização. Abriremos o terminal, por meio da qual acessaremos a pasta do Tomcat pela raiz. Acessando-se o "bin", pode-se

derrubar ou subir o servidor. Queremos, neste momento, derrubá-lo (`./shutdown.sh`), portanto precisaremos executar o arquivo.



```

Terminal Shell Editar Visualizar Janela Ajuda
bin — lazaro@MacBook-Pro-de-Lazaro — ../Tomcat/bin — zsh — 115x33
|→ ~ pwd
/Users/lazaro
|→ ~ cd /Library
/Library cd Tomcat
|→ Tomcat ls
LICENSE          RELEASE-NOTES bin          lib          temp          work
NOTICE           RUNNING.txt   conf         logs         webapps
|→ Tomcat cd bin
|→ bin ./shutdown.sh
Using CATALINA_BASE:  /Library/Tomcat
Using CATALINA_HOME:  /Library/Tomcat
Using CATALINA_TMPDIR: /Library/Tomcat/temp
Using JRE_HOME:       /Library/Java/JavaVirtualMachines/jdk1.8.0_102.jdk/Contents/Home
Using CLASSPATH:      /Library/Tomcat/bin/bootstrap.jar:/Library/Tomcat/bin/tomcat-juli.jar
|→ bin █

```

Recarregaremos a página para verificar se isto foi feito realmente, e em seguida, vamos voltar à pasta "webapps". Queremos refazer o *deploy* da mesma aplicação, sendo recomendável, portanto, apagar a versão anterior dela.

Também disponibilizarei na parte de exercícios o arquivo `.war`, para que vocês o coloquem no Tomcat assim que ele for instalado. Copiamos e colamos este `listavipServlet.war` à pasta "webapps". Quando ocorrer o "start", será gerada uma pasta homônima, que será a aplicação em si. Para subir a aplicação, voltaremos ao terminal, utilizando o `./catalina.sh`. Antes de rodar este comando, porém, precisamos que o container deixe uma porta aberta para quando o Eclipse for debugar e chegar ao Tomcat, à container. Como faremos isto?

Esta comunicação é feita pelo **JPDA** (ou *Java Platform Debugger Architecture*), "arquitetura de *Debug* da plataforma Java". O Tomcat precisa subir com esta configuração, encontrada no comando `./catalina.sh`.

Para entendermos isto melhor, abriremos a pasta "bin" no **Atom** (você pode escolher qualquer outro de sua preferência, como o **Notepad++**, o próprio **Eclipse**, bloco de notas, entre outros). Selecionaremos o arquivo `catalina.sh` e buscaremos a porta `8000`, que é a da JPDA. Precisaremos nos atentar a esta porta, através da qual o Tomcat subirá. Voltando ao terminal e dentro da pasta "bin", digitaremos `./catalina.sh jpda start`.

Vamos checar se as alterações foram feitas corretamente indo ao navegador e recarregando a página. Voltaremos àquela página inicial vista no início deste vídeo, portanto, o *deploy* foi realizado.

Já temos o artefato, colocamos o *deploy* neste Tomcat a ser instalado em seu *host*. Agora, ele está preparado para passar pelo processo de *Debug* remoto, com JPDA habilitado.