



Trabalho Nº 1 – Java API sockets

1. Introdução:

Este trabalho tem como objectivo o desenvolvimento de uma aplicação cliente/servidor que permita que vários utilizadores em modo distribuído consigam participar em votações. O trabalho será realizado com o API de sockets em Java. Este trabalho deverá ser feito individualmente ou em grupos de dois ou três alunos.

2. Descrição do trabalho:

A aplicação distribuída que se pretende implementar deverá suportar um sistema de votações. Este sistema possui múltiplos itens em votação, e recebe votos dos utilizadores. Os itens em votação têm um prazo máximo, findo o qual o servidor não aceita mais votações e calcula o total de votos, respectivas percentagens e item ganhador. Por sua vez, um utilizador começa por efectuar um login. Em resposta o servidor envia-lhe uma lista de itens que estão em votação, o que permitirá ao utilizador votar em um determinado item. Além dos votantes, existem também os administradores do sistema que têm a capacidade de introduzir itens para votação e que poderão enviar notas informativas para os utilizadores.

A primeira meta do trabalho deverá ser implementada usando o API de Sockets em Java. Toda a comunicação unicast será feita utilizando os sockets TCP, enquanto que toda a comunicação multicast será feita utilizando sockets UDP. O multicast será utilizado exclusivamente para as notas informativas enviada pelos administradores do sistema. No entanto, a implementação do multicast será feita com mensagens ponto a ponto, não havendo lugar à utilização de endereços multicast. O servidor deverá ser multi-threaded.

De seguida apresentam-se os tópicos fundamentais a implementar na aplicação:

3. Protocolo de Comunicação:

Com excepção das notas informativas em multicast, toda a comunicação entre clientes e servidor deverá ser feita através de sockets Stream. Caso ache conveniente poderá usar ObjectStreams para o envio de objectos. Tenha em atenção que deverá definir com rigor o formato das mensagens da sua aplicação distribuída e deverá oferecer pelo menos as seguintes mensagens no sentido cliente-servidor: (a) login; (b) obter lista de itens em votação; (c) votação.

No sentido servidor-cliente, deverão existir pelo menos as seguintes mensagens: (a) login válido ou inválido; (b) lista de itens em votação; (c) resposta a um voto (por exemplo, sucesso ou insucesso); (d) notificação do término da votação. Tenha especial atenção ao assincronismo de mensagens de cliente/servidor e servidor/cliente.

Para além dos utilizadores deve considerar ainda os administradores do sistema, que também interactivam com o servidor, utilizando, para o efeito, as seguintes mensagens: (a) login; (b) listar itens em votação; (c) adicionar um novo item; (d) enviar notas informativas. Por outro lado o servidor deverá responder com indicações de sucesso ou insucesso a cada uma destas mensagens.

4. Gestão de clientes e outra informação:

Nesta fase do trabalho a gestão de utilizadores poderá ser feita de uma forma manual, através da manipulação de um ficheiro do lado do servidor, onde se guarda a informação dos utilizadores registados na aplicação, os respectivos logins e passwords. Deverá ainda existir informação persistente com os itens existentes e com o valor das votações.

Esta informação deverá ser mantida num ficheiro à parte, de tal forma que o servidor consiga recuperar toda a informação necessária para manter o serviço em caso de uma falha transitória.

5. Interface Gráfica:

Os alunos poderão utilizar simplesmente linha de comando. O desenvolvimento de uma interface gráfica não será bonificado.

6. Arquitectura do Serviço

A arquitectura final do serviço será construída por partes ao longo das várias metas deste trabalho. Na primeira meta a arquitectura a construir será apenas composta por utilizadores/votante, um administrador e um servidor, que comunicam por TCP (e também por UDP, no caso das mensagens multicast de notas informativas). Esta modalidade está representada na figura 1.

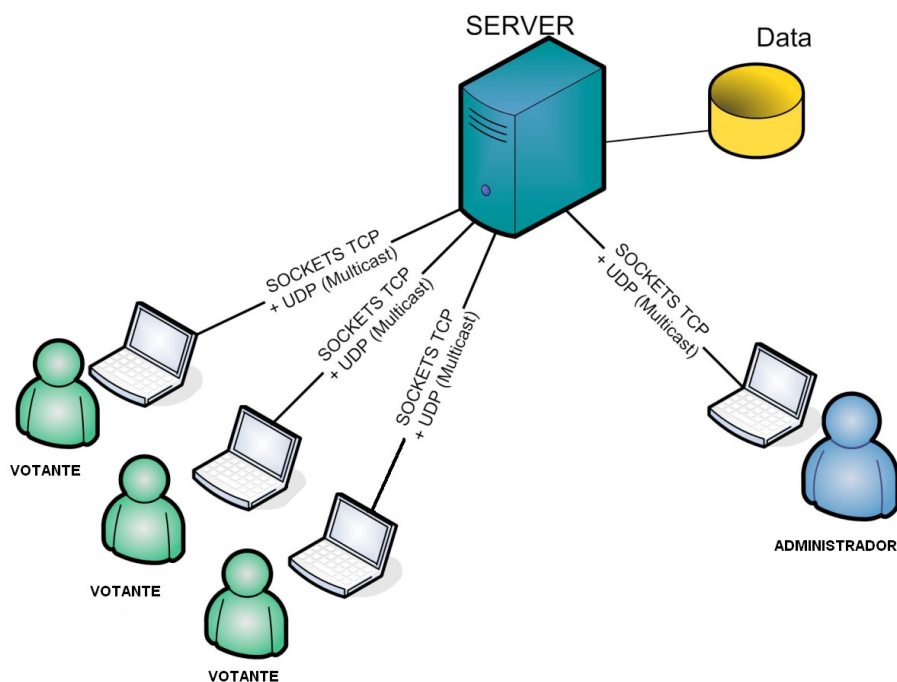


Figura 1: Arquitectura da aplicação, utilizando Java Sockets

7. Tratamento de Excepções

A sua aplicação deverá ser tolerante a falhas temporárias de curta duração no canal de comunicação TCP. Ou seja, se a rede ou o servidor ficarem indisponíveis por alguns instantes, a sua aplicação irá receber uma excepção. Esta excepção deverá ser tratada do lado do cliente de forma a tentar de novo a abertura do socket com o servidor. O cliente não pode visualizar a ocorrência de excepções na sua aplicação. Quando não existe conectividade com o servidor o cliente deverá simplesmente indicar que aguarda ligação ao servidor. Se ao fim de algum tempo (ex: 5 minutos) o cliente continuar sem conectividade para o servidor é que deverá considerar a ligação ao servidor como perdida.

Para simular estas situações poderá desligar o servidor (ctrl-c) e voltar a iniciá-lo. Tenha em atenção que durante o período de tempo em que a conectividade com o servidor esteja inactiva todas as mensagens enviadas pelos clientes e os dados introduzidos pelos utilizadores (utilizadores e administradores) devem ser mantidas em buffers locais e não devem ser perdidas, sendo enviadas para o servidor assim que este voltar a estar activo.

8. Futuras Fases do Trabalho

Deverá ainda ter em conta que este é apenas o primeiro trabalho, que funciona sobre um canal de comunicação: Sockets Stream. No trabalho Nº 2, irá ser colocado um Front-End de múltiplas máquinas entre o cliente e o servidor, que comunicarão entre si utilizando RMI. No trabalho Nº 3 os clientes comunicarão com o servidor utilizando JSPs/JavaBeans e finalmente, no trabalho Nº 4, utilizando SOAP. O objectivo do trabalho é único: fornecer uma aplicação de gestão de votações, que funcione sobre várias tecnologias de comunicação distribuída.

9. Relatório do Trabalho

Além do software desenvolvido deverá ainda escrever um relatório do trabalho prático. O relatório deverá ter as seguintes secções:

- Introdução
- Arquitectura da aplicação
- Formato das mensagens da aplicação
- Tratamento de falhas transitórias nos sockets
- Manual de utilização
- Manual de instalação e configuração
- Descrição dos testes efectuados à aplicação

10. Prazo de Entrega do Trabalho

O prazo para a entrega do trabalhos é de quatro semanas a contar da aula em que o enunciado foi apresentado. Não serão aceites trabalhos fora do prazo. A entrega deverá ser feita por e-mail num ficheiro zip (ou rar) contendo o relatório (**obrigatório**) em PDF e as listagens dos programas realizados, obedecendo OBRIGATORIAMENTE ao seguinte formato:

a123456-trabalho-1-CD.zip

11. Referências

SUN - Essentials of the Java Programming Language

<http://java.sun.com/developer/onlineTraining/Programming/BasicJava1>

<http://java.sun.com/developer/onlineTraining/Programming/BasicJava2>

SUN - The Java Tutorials, Networking

<http://java.sun.com/docs/books/tutorial/networking/index.html>