

Relatório

João Verdilheiro, nº38156

5 de julho de 2021

1 Introdução

O trabalho realizado foi implementado em Java, usando JFrame para criar uma pequena interação quando o servidor e cliente são iniciados. Quando iniciado o servidor irá pedir a porta em que deve ser ligado e o cliente pedirá o host, porta e nome do cliente. Depois de conectado o cliente irá abrir uma janela de chat onde terá todo o feedback e mensagens necessários. Durante a execução de ambos, estes irão ter alguma informação na linha de comandos facilitando a interpretação do que está a decorrer no código. São utilizados vários comandos para tratar mensagens especiais sendo que cada um deles tem argumentos específicos (que estão explicados no ficheiro README.txt). Para implementar este trabalho foi utilizado como base o código disponibilizado em <https://www.devmedia.com.br/como-criar-um-chat-multithread-com-socket-em-java/33639>, sendo que foram feitas bastantes alterações para que o pretendido fosse implementado.

2 Sistema de “chat”

O sistema de chat implementado, utiliza o socket aberto pelo servidor e pelo cliente para comunicar entre si. Cada mensagem enviada pelo cliente é recebida pelo servidor e devidamente enviada a cada cliente ou a um cliente. Por padrão as mensagens enviadas pelo cliente serão entregues a todos os clientes online, mas no caso da mensagem pretendida ser enviada a algum cliente específico (privada) esta terá de ser enviada com um comando específico. No caso para mensagens privadas será necessário inserir o comando `"/p [nome do cliente] [mensagem]"`. Assim quando a mensagem for enviada para o servidor este saberá que se trata de uma mensagem privada e deve ser enviada apenas a um cliente. Se a mensagem enviada for para um cliente que não esteja online no momento esta será guardada em um ArrayList para que posteriormente possa ser enviada ao cliente em questão.

3 Sistema de transferência de ficheiros

À semelhança do sistema de “chat”, o sistema de transferência de ficheiros também depende de um comando para que possa ser tratado como tal, `"/f [cliente/s a enviar] [ficheiro]"`. Por norma o ficheiro enviado será enviado do ambiente de trabalho do cliente. O cliente que enviar o ficheiro irá abrir um novo socket para enviar o ficheiro, sendo que irá informar o servidor de que pretende enviar o ficheiro, para que este faça o mesmo. Depois de abertos os sockets, e conectados o cliente enviará o ficheiro ao servidor que irá guardar o ficheiro. Assim que o ficheiro for devidamente transferido para o servidor este irá reencaminhar o ficheiro para os clientes pretendidos, consoante o comando inserido pelo cliente, no caso “all” ou algum em específico. Para transmitir o ficheiro para os clientes o processo é bastante semelhante, sendo que será aberto um novo socket para isso.

4 Sistema de jogos com um jogo de exemplo

O sistema de jogo implementado é o jogo do galo e tem por base o sistema de chat sendo que será necessário que o cliente/s em questão troquem mensagens específicas entre si, isto é, será necessário que os jogadores em questão usem o comando `"/g [cliente] [jogada]"` para que possam jogar entre si. Inicialmente um cliente irá “pedir” para jogar com outro, sendo que será verificado se o jogador em questão está ou não disponível para jogar. Caso esteja disponível o jogo terá início e irá informar o

jogador. Assim que cada jogador jogar o outro cliente irá receber um esboço do estado do jogo e irá fazer a sua jogada.

5 Notas

O trabalho submetido não implementa na totalidade a transferência de ficheiros, sendo que um ficheiro será apenas enviado a clientes online, e é assumido que apenas um ficheiro é transmitido por vez.