“IRC-Trabalho de Sockets”

Augusto Vítor Sousa Campos (501070324)

ascampos@student.dei.uc.pt
http://www.dei.uc.pt/~ascampos

Objectivo:
O objectivo deste trabalho é o desenvolvimento de um ambiente de comunicação, o IRCChat, para troca de mensagens entre dois ou mais utilizadores. O ambiente de comunicação é constituído por um servidor e um ou vários clientes podendo os clientes trocar entre si mensagens com ligações P2P.

Visão Geral:
Assim que a ligação é realizada, é perguntado pelo Servidor ao Cliente qual o Nickname que o utilizador pretende, após validar se não existe outro Nickname igual, é atribuído ao utilizador e é incluído na lista que é então enviada a todos os utilizadores ligados nesse instante para o utilizador. Para os restantes, apenas vai a indicação de que foi adicionado o utilizador X.
Após o utilizador se encontrar validado, pode enviar ao servidor comandos ou mensagens.
Os comandos são uma série de palavras-chave precedidas por uma "/".

Comandos:

/help	-> Mostra uma lista dos comandos disponíveis
/quit	-> Termina a conexão com o servidor
/users	-> Força a obtenção da lista dos utilizadores logados
/to <Utilizador> <mensagem>	-> Envia a mensagem privada apenas para o Utilizador

Qualquer outro texto trocado com o servidor é encarado como uma mensagem que será posteriormente enviada para todos os utilizadores.
Quando um utilizador é adicionado, os outros recebem a mensagem "+User: X", sendo X o Nickname do utilizador adicionado e, quando um utilizador deixa de estar conectado quer pelo comando "/quit" ou por quebrar a ligação com o servidor, todos os outros recebem "-User: X".
O Servidor guarda no ficheiro Logs.txt registo de todas as conexões dos clientes.

Ferramentas:
Foram utilizadas para o desenvolvimento deste trabalho a linguagem de programação Java (http://en.wikipedia.org/wiki/Java_(programming_language)) na edição Standard (Java Platform, Standard Edition) na Versão 1.6.0_10 (Java(TM) SE Runtime Environment - build 1.6.0_10-b33).
Como IDE foi usado o Netbeans(http://en.wikipedia.org/wiki/Netbeans).
Foram ainda utilizados como suporte e ajuda no desenvolvimento Telnet, Vim, Gedit, Javadoc, Man Pages, Internet, etc.
Todas as aplicações foram usadas e/ou desenvolvidas em ambiente Linux (http://en.wikipedia.org/wiki/Linux).

Descrição funcional:
Foi desenvolvido um Servidor em Java e fica à escuta no número do porto que é passado como primeiro parâmetro na altura do arranque.
Exemplo: java -jar SServer.jar 6222 - O servidor arranca na porta 6222, o que quer dizer que é a esta porta que os Cliente se tem que ligar.
Nota: Só se pode arrancar uma instância do Servidor para cada porta especificada.

Foi desenvolvido um cliente em Java que aceita como parâmetro o servidor e porto ao qual se deve ligar.
Exemplo: java -jar SClient.jar 192.168.1.1 6222 - O Cliente arranca, ligando-se ao servidor 192.168.1.1 no porto 6222, pode em vez do IP ser usado o nome do servidor, mas garantindo a sua resolução DNS.
Para as ligações P2P, foi convencionado a utilização da porta 6223 entre os clientes
Podem ser arrancados quantos clientes se pretender, dependendo apenas dos limites do sistema. Caso sejam arrancados de máquinas diferentes, o servidor tem que ser visível à máquina cliente a

nível de rede.
Nota:Todas as opções mencionadas para o cliente foram pensadas para também ser possível a utilização de uma ligação simples ligação telnet ao servidor e respectivo porto. Claro está que, usando este tipo de ligação, apenas se poderá utilizar a arquitectura Cliente/Servidor, ficando de fora as ligações P2P.

As Classes do SServer – Servidor de Chat:
Main.java - é a primeira classe a ser instanciada e tem como objectivo inicializar o servidor com os parâmetros passados.
SServer.java – é a classe responsável por ficar à escuta no porto passado na altura da criação e, a quando de cada conexão dos clientes, desencadeia a criação dos restantes componentes.
ServerDispatcher.java – é a classe responsável por gerir a lista de Clientes, validar os Nicknames e tratar da fila de mensagens.
ClientInfo.java – é a classe agregadora de toda a informação do cliente.
ClientListener.java – é a classe que tem a seu cargo a escuta de texto(comandos e/ou mensagens) vindo do cliente.
ClientSender.java – é a classe que serve entreposto de comunicação do servidor o/os cliente(s).

As Classes do SClient – Cliente chat:
Main.java - é a primeira classe a ser instanciada e tem como objectivo inicializar o cliente com os parâmetros passados.
SClient.java – é a classe responsável por criar a ligação ao servidor, enviar para a consola o retorno do servidor e ainda criar ele próprio o servidor P2P.
P2PClientInfo.java – é a classe responsável por albergar toda a informação dos clientes ligados via P2P e enviar para a consola as mensagens dos clientes ligados por P2P..
Sender.java – é a classe responsável por enviar mensagens a outro cliente, via P2P.