



Relatório de Projeto

Chat de atendimento a clientes

Autores:

David José Ressurreição Alves a79625

Pedro Rafael Carvalho Miranda a79662

Pedro Miguel Fonseca Sampaio a79668

Emanuel Blaj, e8457



Índice

| | |
|--|---|
| 1.Introdução..... | 2 |
| 2.Desenvolvimento | 3 |
| 2.1. Contexto do Projeto..... | 3 |
| 2.2. Estrutura do programa..... | 3 |
| 3.Falhas e limitações na implementação | 4 |
| 4.Conclusões..... | 5 |



1.Introdução

Para a realização deste projeto foi-nos pedido para desenvolvermos um servidor para um *chat* de atendimento a clientes, no qual existem dois tipos de utilizadores, *genious* (caso vá tirar dúvidas a utilizadores) e *user* (caso vá pedir apoio). O programa deve emparelhar *user* e *genious* dois a dois, e dar início a um *chat* privado entre ambos. O código do programa que nós desenvolvemos foi elaborado em linguagem *Java*. Serve este relatório para explicar o programa desenvolvido.



2.Desenvolvimento

2.1. Contexto do Projeto

Os objetivos deste projeto são:

- Iniciar sessão para dois tipos de utilizadores (*genious* e *user*);
- Emparelhar utilizadores dois a dois, *genious* e *user*;
- Iniciar um *chat* entre *genious* e *user*.

2.2. Estrutura do programa

Servidor

O servidor possui 2 classes, a classe *Server* e a classe *Client*.

O servidor comunica com os clientes através do protocolo TCP, ligando-se à porta 9999.

Cada vez que um cliente se liga ao servidor é iniciado uma *thread*, onde são ligados os clientes com recurso à classe *LigarClientes*. Nesta classe, no método *run* é verificado se o tipo introduzido é *user* ou *genious*. Depois disto se o tipo for aceite, é emparelhado com um cliente do outro tipo, se já houver algum, senão espera que se conecte um. Assim que são emparelhados os seus nomes são inseridos num *HashMap*. Posteriormente a cada mensagem enviada por cada um dos clientes, o servidor redirecciona a mensagem para o parceiro do cliente que enviou a mensagem.

Cliente

Para a construção do programa cliente criámos 1 classe em Java: a classe *Client*, que é a classe principal que serve para iniciar o programa com o qual o cliente irá interagir, este programa têm também métodos para obter o nome do cliente que se conecta (*getNome()*). Depois destes métodos terem sido executados com êxito o programa passa para o seu método *run()*, no qual é iniciado uma *socket* para *localhost* na porta 9999 pedido o nome de utilizador e verificado através da comunicação com o servidor se o nome introduzido é *user* ou *genious* e se existem utilizadores para os quais pode comunicar.



3. Falhas e limitações na implementação

Apesar de tudo, este programa de chat tem algumas limitações, as quais não conseguimos resolver por completo até à conclusão deste relatório:

- Quando *user* ou *genious* é inserido, este conecta-se ao último *user* ou *genious* inserido, o que não devia ocorrer, dado que deveria conectar-se ao primeiro *user* ou *genious* em fila de espera.
- Quando se fecha um painel de comunicação de *user* ou *genious* estes são apagados do servidor, impossibilitando a comunicação com novos *user* e *genious*



4. Conclusões

Depois de concluirmos a elaboração deste projeto podemos afirmar que estamos orgulhosos do resultado final, sabendo, no entanto que ainda poderíamos, e teríamos certamente conseguido, aperfeiçoar mais o projeto se tivéssemos mais tempo disponível, dado que esse foi o nosso principal problema, devido à elevada carga horária pedida para projetos de outras unidades curriculares. Contudo foi uma experiência bastante enriquecedora das nossas competências.