Cliente de chat

Universidade de Aveiro

Alexandre Lourenço, Ana Margarida Silva, Susana Dias



Cliente de chat

Departamento de Eletrónica, Telecomunicações e Informática

Universidade de Aveiro

Alexandre Lourenço, Ana Margarida Silva, Susana Dias

(79894) alexandre.lourenco@ua.pt, (77752) margarida
ocs@ua.pt, (80410) susanadias@ua.pt

21/04/2016

Resumo

Com este trabalho de aprofundamento desenvolvemos um cliente de chat em python, com recurso a sockets e json, que permite enviar e receber mensagens para os restantes utilizadores.

Conteúdo

1	Intr	Introdução														1							
2	Arquitetura da aplicação 2.1 getOn()															2							
	2.2	nicknam	e()																				2
	2.3	message	() .																				2
	2.4	main().	٠.																				2
3	3 Utilização do cliente 3.1 Opções de Utilização:											3											
	3.1	Opções o	de U	tiliz	zaç	ão	:																3
	3.2	Exemple	s .																				3

Capítulo 1

Introdução

Cada vez mais a comunicação a longas distâncias é facilitada com recurso a transmissões de pacotes de dados pela internet entre utilizadores, e foi nessa perspectiva que este trabalho foi concebido, tendo como vista a criação de uma interface que permitisse connectar vários clientes a um servidor criando assim um chat em python que recorre a json para formatação das mensagens enviadas, posteriormente, em sockets.

Capítulo 2

Arquitetura da aplicação

A ligação ao servidor LabiChat é feita pelo protocolo TCP/IP conectando-se ao endereço 37.247.48.69 pela porta 1863. O cliente é composto por três funções fora da main() que representam comandos que podem ser executados pelo utilizador.

2.1 getOn()

Converte o campo "to"e imprime no terminal todas as pessoas que se encontram online no momento.

$2.2 \quad \text{nickname}()$

Altera o nickname do utilizador. Quando esta função é invocada, o campo "from" é substituído pela string inserida.

2.3 message()

Recebe duas strings (data0 e data1) e converte-as em json como mensagem e destinatário respetivamente. Se a mensagem for vazia, não será enviada.

2.4 main()

Gere tudo o que é enviado e recebido no socket. Após receber o nickname do utilizador, fica à espera de uma resposta do servidor ou de um pedido do cliente. O módulo select.select() é utilizado para que apenas receba mensagens quando não está uma mensagem a ser digitada, pois isso daria origem a uma quebra. Ao digitar uma das opções, a respetiva função será executada, à exceção do caso do comando sair, em que o programa será terminado.

Capítulo 3

Utilização do cliente

Para aceder ao chat terá de ser introduzido no terminal "python clientFinal2.py".

3.1 Opções de Utilização:

sair - Para quando o utilizador quiser sair do programa online - O cliente será informado de quem se encontra online msg - Será possível para o cliente escolher o destinatário para quem deseja enviar a mensagem e escrever a mensagem nome - será permitido ao cliente mudar de nome

3.2 Exemplos



Figura 3.1: Aceder à consola



Figura 3.2: Nome de utilizador

```
Susana@susana-SATELLITE-L50-B:~/Área de Trabalho/LabI$ python clientLABI.py

------LABI CHAT------

Opcoes: online; msg; nome; sair

On-line: #12993 #12994 #13002 #13004 #13015 #13017 Miguel aibot mags robotN sal wisebot

Nome: Asdrubal msg
Para: sal
Mensagem: olá!
```

Figura 3.3: Envio de mensagem

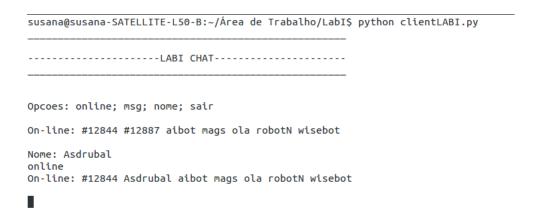


Figura 3.4: Utilizadores online

Figura 3.5: Terminar chat

Contribuições dos autores

- \bullet Alexandre Lourenço (40%)
- Ana Margarida Silva (30%)
- Susana Dias (30%)