



Universidade do Minho

Escola de Engenharia

Sistemas Distribuídos

ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA
2020/2021

(Docente: Carlos Miguel Ferraz Baquero Moreno)

1 de junho de 2021

Trabalho Prático

Sistema de vacinação

Marco António Cerqueira Araújo – a89387@alunos.uminho.pt

Rui Filipe Ribeiro Freitas – a84121@alunos.uminho.pt

Sandro Teixeira Ribeiro – a85316@alunos.uminho.pt

Tiago João Pereira Ferreira – a85392@alunos.uminho.pt

Índice

Índice de figuras	2
Introdução	2
Desenvolvimento.....	3
1. Contextualização	3
2. Modelo Cliente-Servidor	3
2.1. Cliente	3
2.2. Servidor	4
3. Fluxograma	5
Testes e discussão de resultados.....	6
Conclusão.....	7

Índice de figuras

Figura 1 – Comando para iniciar o cliente.....	3
Figura 2 - Comandos de interação entre cliente e servidor.....	3
Figura 3 - Base de dados em formato CSV.	4
Figura 4 - Exemplo do servidor em funcionamento.....	4
Figura 5 - Fluxograma do servidor.....	5
Figura 6 - Testes exemplo 1.....	6
Figura 7 - Testes exemplo 2.....	6

Introdução

No âmbito da unidade curricular de Sistemas Distribuídos foi proposto pelo docente a implementação de um servidor capaz de agendar vacinações em tempo lógico e em várias zonas do país. Um cliente tem de ser capaz de se conectar a um servidor através de *sockets* e agendar uma vacinação. Este trabalho foi implementado com o auxílio da linguagem de programação JAVA e como software de desenvolvimento o Visual Studio Code.

O objetivo deste projeto passa por melhorar as nossas capacidades tanto em linguagem JAVA como no trabalho em grupo. Para isto tivemos de colocar em prática todos os conhecimentos adquiridos ao longo do semestre na cadeira de Sistemas Distribuídos.

Desenvolvimento

1. Contextualização

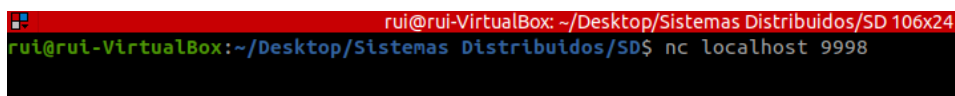
O objetivo do trabalho prática passa pela implementação de um servidor capaz de comunicar com vários clientes concorrentemente com o simples objetivo de cada cliente poder realizar um agendamento ou uma desmarcação de uma vacinação.

De modo a sermos capazes de implementar este servidor temos de obter conhecimentos aprofundados na manipulação de *threads*, de *sockets* e de controlo de concorrência. Como estes tópicos foram todos lecionados nas aulas de Sistemas Distribuídos já tínhamos uma vaga ideia de como realizar a implementação. Com um pouco de pesquisa para obtenção de uma melhor perceção relativamente aos conceitos a utilizar neste projeto conseguimos obter tudo o que necessitávamos na elaboração deste.

2. Modelo Cliente-Servidor

2.1. Cliente

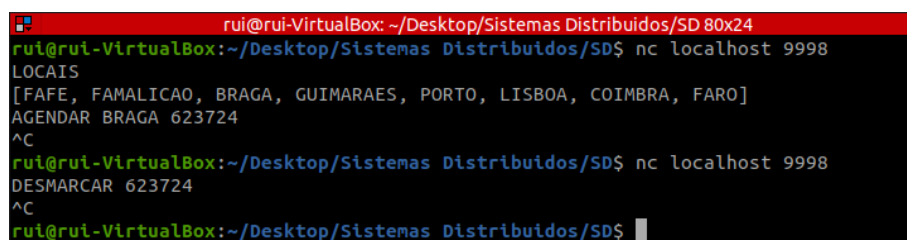
Relativamente ao cliente utilizamos um comando do Linux que permite a conexão a um servidor através de uma porta sem a necessidade de implementação de código próprio. Este comando foi sugerido pelo docente e corresponde a escrever na consola a seguinte linha de código com *nc* como diminutivo de *netcat*, *localhost* que corresponde ao endereço do servidor a utilizar e a porta da comunicação, a 9998.



```
rui@rui-VirtualBox: ~/Desktop/Sistemas Distribuidos/SD 106x24
rui@rui-VirtualBox:~/Desktop/Sistemas Distribuidos/SD$ nc localhost 9998
```

Figura 1 – Comando para iniciar o cliente.

O cliente tem à sua disposição 3 comandos para interagir com o servidor, *LOCALIS* para receber uma lista de locais possíveis para realizar o agendamento. *AGENDAR* [LOCAL] indicando um local onde pretende ser agendada a marcação e por último o *DESMARCAR* [SNS] onde o cliente com o número SNS liberta um agendamento realizado anteriormente. De seguida apresentamos um exemplo de um utente a utilizar os vários comandos.



```
rui@rui-VirtualBox: ~/Desktop/Sistemas Distribuidos/SD 80x24
rui@rui-VirtualBox:~/Desktop/Sistemas Distribuidos/SD$ nc localhost 9998
LOCALIS
[FAFE, FAMALICAO, BRAGA, GUIMARAES, PORTO, LISBOA, COIMBRA, FARO]
AGENDAR BRAGA 623724
^C
rui@rui-VirtualBox:~/Desktop/Sistemas Distribuidos/SD$ nc localhost 9998
DESMARCAR 623724
^C
rui@rui-VirtualBox:~/Desktop/Sistemas Distribuidos/SD$
```

Figura 2 - Comandos de interação entre cliente e servidor.

2.2. Servidor

Quanto ao servidor este foi desenvolvido com o objetivo possa atender vários clientes ao mesmo tempo. Para a implementação de clientes em simultâneo foram usadas *threads* e de modo que vários clientes não acedam às mesmas variáveis ao mesmo tempo foi utilizado controlo de concorrência através de *locks*.

O programa servidor por nós implementado contém 3 classes, a classe *server.java* onde é realizada a criação e abertura da *socket* para a comunicação com os clientes e por cada cliente conectado é criado uma *thread*. Relativamente à classe *Marcacao.java* esta tem como objetivo a criação de um objeto *Marcação* com os seguintes atributos: número de SNS, local, hora de marcação e estado do agendamento. Por último temos a classe *ListaMarcacoes.java* onde é criado um *ArrayList* com os vários agendamentos e é feita toda a parte lógica do trabalho no que toca a agendar, desmarcar e a verificar o número SNS.

Relativamente às funcionalidades do servidor este apresenta algumas características como as marcações apenas poderem ser agendadas entre as 9 e as 19 horas com um intervalo de 20 minutos entre cada uma assim como se a marcação for feita num dia apenas será marcada para o dia seguinte caso haja vagas. Para além disso é criado uma base de dados em formato CSV de modo que se possa aceder a uma lista completa das marcações de forma mais simples (Figura 3).

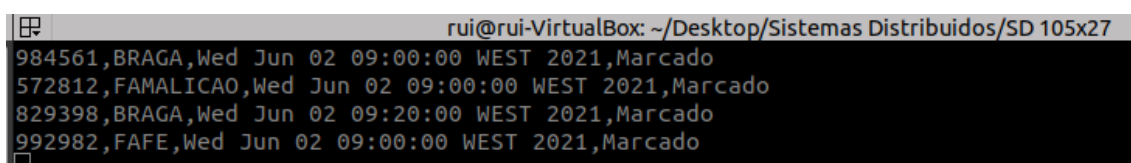


The screenshot shows a code editor with four tabs: Marcacao.java, ListaMarcacoes.java, server.java, and logs.csv. The logs.csv tab is active, displaying a CSV file with 10 rows of appointment data. The columns are SNS, Local, Hora, Marcacao, and Estado.

SNS	Local	Hora	Marcacao	Estado
128392	FAFE	Wed Jun 02 09:00:00 WEST 2021		Marcado
289233	BRAGA	Wed Jun 02 09:00:00 WEST 2021		Marcado
892834	FAMALICAO	Wed Jun 02 09:00:00 WEST 2021		Marcado
829825	GUIMARAES	Wed Jun 02 09:00:00 WEST 2021		Marcado
293824	FAFE	Wed Jun 02 09:20:00 WEST 2021		Marcado
291823	BRAGA	Wed Jun 02 09:20:00 WEST 2021		Marcado
123892	FAMALICAO	Wed Jun 02 09:20:00 WEST 2021		Marcado
237827	BRAGA	Wed Jun 02 09:40:00 WEST 2021		Marcado
283273	BRAGA	Wed Jun 02 10:00:00 WEST 2021		Marcado

Figura 3 - Base de dados em formato CSV.

Em seguida apresentamos um exemplo do servidor após este ter recebido 4 marcações de 4 utentes diferentes, 2 de Braga, 1 de Famalicão e 1 de Fafe. Também podemos observar que apesar de locais diferentes as horas começam sempre a ser marcadas às 9 do dia seguinte, sendo estas incrementadas quando se trata da mesma localidade.



The screenshot shows a terminal window with the prompt 'rui@rui-VirtualBox: ~/Desktop/Sistemas Distribuidos/SD 105x27'. The output displays four lines of appointment data, matching the first four rows of the CSV file in Figure 3.

```
984561,BRAGA,Wed Jun 02 09:00:00 WEST 2021,Marcado
572812,FAMALICAO,Wed Jun 02 09:00:00 WEST 2021,Marcado
829398,BRAGA,Wed Jun 02 09:20:00 WEST 2021,Marcado
992982,FAFE,Wed Jun 02 09:00:00 WEST 2021,Marcado
```

Figura 4 - Exemplo do servidor em funcionamento.

3. Fluxograma

Em seguida apresentamos um fluxograma simples de modo a tornar mais simples a visualização da solução por nós implementada.

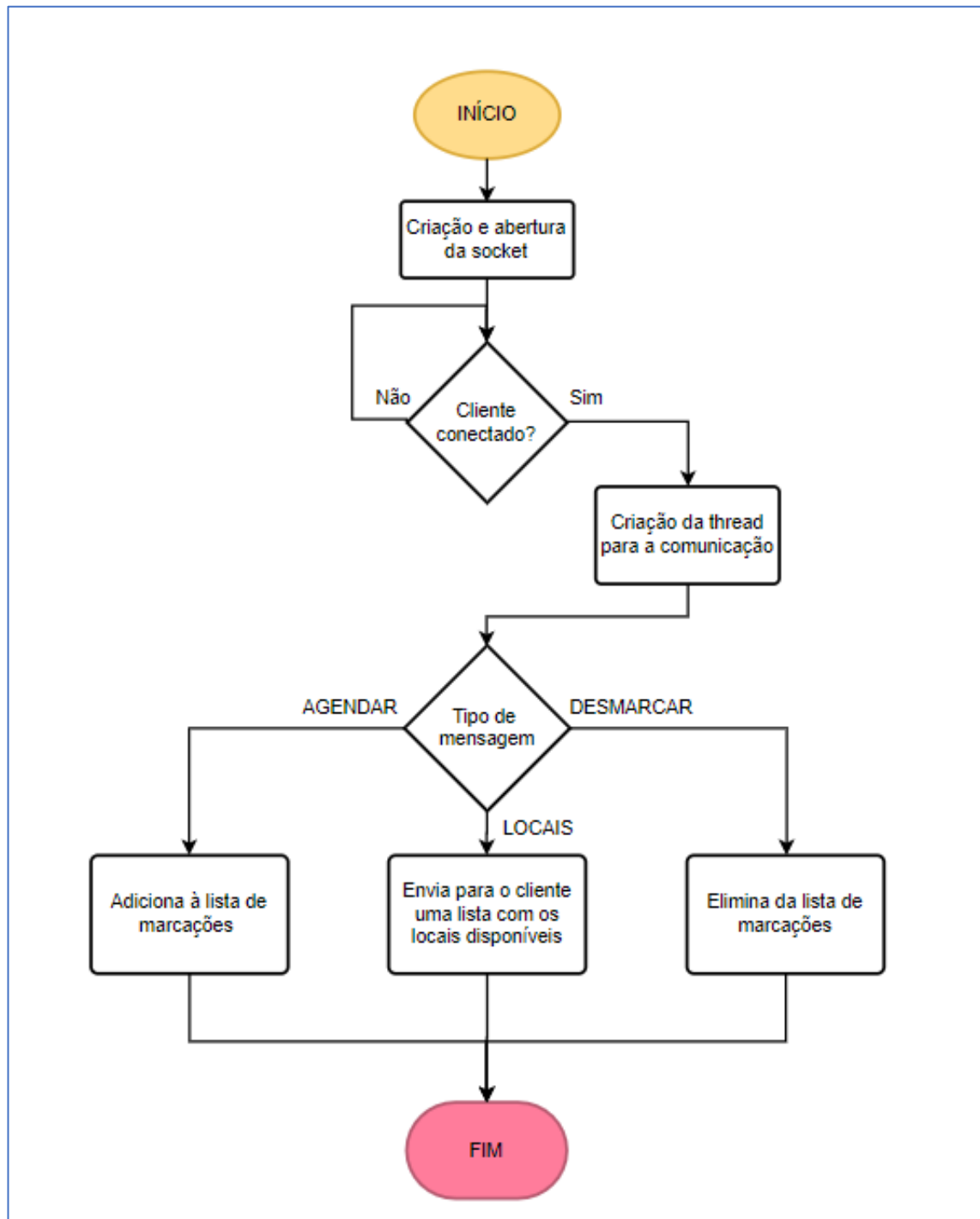


Figura 5 - Fluxograma do servidor.

Testes e discussão de resultados

De modo a comprovar se o programa por nós implementado funcionava de acordo com as expectativas realizamos alguns testes. Na seguinte imagem apresentamos a criação de 4 utentes e um servidor com o primeiro cliente a consultar os locais disponíveis para realizar a marcação e após isso faz o agendamento em Braga com o seu número de SNS. Os restantes utentes fazem também a marcação, mas 2 em Braga e um em Fafe. Na imagem à direita observamos a lista de marcações realizada no servidor.



Figura 6 - Testes exemplo 1.

Após isso decidimos realizar uma desmarcação do segundo utente e realizar uma marcação por parte de um quinto utente de modo a comprovar que este ficava com a posição previamente ocupada pelo segundo utente.

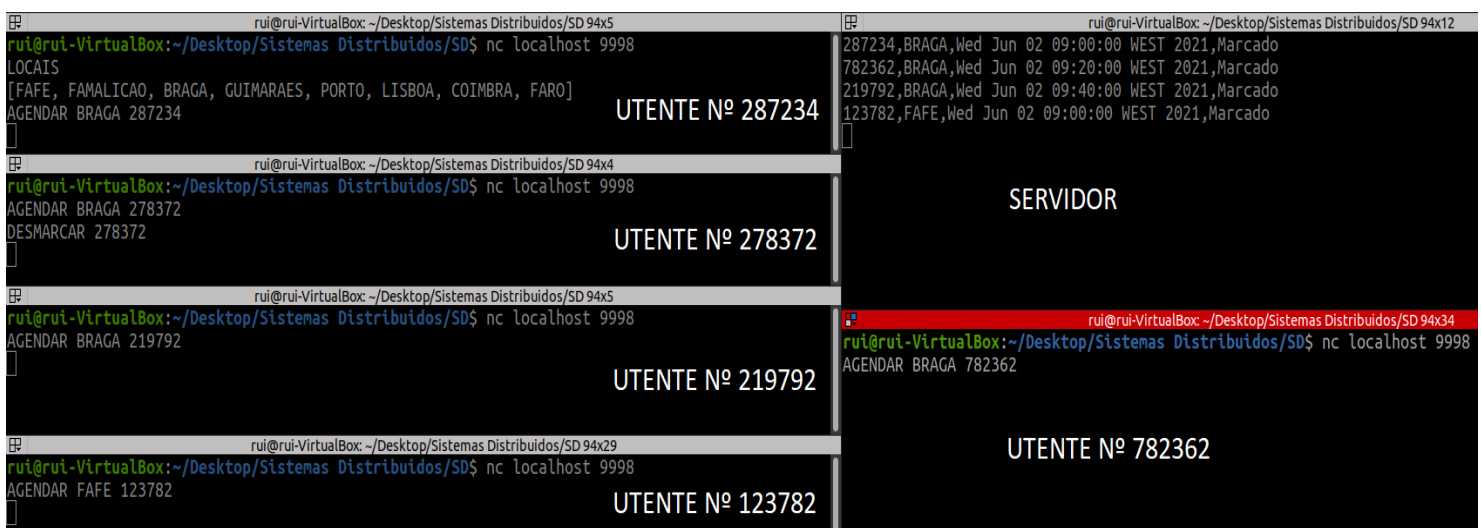


Figura 7 - Testes exemplo 2.

Conclusão

Com a realização deste projeto o grupo pôs em prática conhecimentos adquiridos ao longo do semestre na cadeira de Sistemas Distribuídos bem como em cadeiras de anos anteriores. Através destes conhecimentos e de pesquisa sobre os vários tópicos que eram necessários compreender conseguimos obter um resultado satisfatório pois conseguimos implementar tudo o que era pedido e ainda resolvemos adicionar funcionalidades extra como a criação de uma simples base de dados para as marcações.

Este projeto foi importante pois permitiu ao grupo consolidar os conhecimentos sobre a funcionalidade e uso de *sockets* bem como trabalhar com várias *threads* ao mesmo tempo. O grupo considera que o objetivo deste trabalho prático foi concluído com sucesso e sem grandes problemas visto cumprir com todos os objetivos.