



Universidade do Minho

Mestrado Integrado em Engenharia Biomédica – Ramo de Informática Médica

Unidade Curricular: Sistemas Inteligentes

Sistemas Inteligentes – Serviço de Farmácias

Discentes: Ana Machado (75088), Ana Ramos (74727), Ana Sousa (74753)

Docentes: Cesar Analide, João Ricardo, Paulo Novais,

Braga

Maio 2018

Capítulo 1

Sumário

A implementação de sistemas multiagentes permite a comunicação entre diferentes agentes através da troca de mensagens. Contudo, é necessário encontrar equilíbrio no número de agentes e as mensagens trocadas para que não ocorra a morte ou esquecimento dos mesmos.

No presente trabalho, implementou-se um sistema de multiagentes de um ambiente farmacêutico. Deste modo, utilizando o *Java Agent Development Framework*(JADE), criaram-se diferentes agentes (utentes e fornecedores) que interagem com o agente farmácia. O propósito deste sistema é garantir o equilíbrio entre a quantidade de medicamentos consumidos pelo utente, enviados da farmácia para o utente e enviados do fornecedor para a farmácia.

Índice

Objetivos	1
Introdução	2
Conceitos Teóricos.....	3
4.1. Agentes Inteligentes.....	3
4.2. Sistemas Multiagente	3
4.3. Construção de Agentes - JADE	3
4.3.3. <i>Behaviours</i>	4
Implementação	6
5.1 Arquitetura	6
5.1.1 Farmácia.....	7
5.1.2 Fornecedor	7
5.2.3. Utente.....	7
Demonstrações.....	9
6.1. Iniciação do Programa	9
6.2 Funcionamento do Programa	10
6.2.1 Relação Utentes-Farmácia	11
6.2.2 Relação Farmácia-Fornecedor	11
Conclusões	14
Bibliografia	15

Capítulo 2

Objetivos

O presente relatório tem como objetivo o desenvolvimento de um sistema multiagente que simule uma farmácia interativa. Este deve apresentar os seguintes requisitos:

- Criação de vários utentes;
- Criação de várias farmácias;
- Criação de fornecedores;
- Manutenção de Stock garantida;
- Capacidade de comunicação entre os agentes;
- Sistema de competição entre os diferentes fornecedores.

De forma geral, é necessário que o sistema tenha capacidade de gerir os stocks de medicamentos dos utentes, farmácias e fornecedores. Os utentes devem consumir os seus medicamentos de acordo com as suas prescrições, requisitando medicamentos à farmácia quando necessário (quantidades baixas em stock). Por sua vez, a farmácia deve garantir ter no seu stock quantidades que satisfaçam os pedidos dos utentes, requisitando aos fornecedores quando necessário.

Capítulo 3

Introdução

Os sistemas multiagente (SMA), tal como o nome indica, são compostos por diversos agentes que agem de forma autónoma, mas interagem com os restantes agentes presentes no sistema. Cada agente pode ter funcionalidades diferentes tais como: coordenação, cooperação, competição ou ainda negociação.^[1]

Os SMA podem ser aplicados em vários contextos na área da saúde, sendo viáveis para resolver problemas a nível de gestão de pacientes, acesso e gestão de informação médica. Assim sendo, entende-se que estes serão viáveis para a gestão de uma farmácia interativa.^[2]

Num ambiente real, o utente que contenha uma determinada receita dirige-se à farmácia para obter os seus medicamentos. Esta por sua vez deve garantir ter medicamentos suficientes para satisfazer os pedidos, recorrendo a fornecedores que a abasteça. A escolha dos fornecedores deve ter em conta os preços e os prazos de entrega que estes oferecem. A simulação deste ambiente num sistema multiagente apresenta uma elevada complexidade, é necessário relacionar, através de mensagens, um elevado numero de agentes, podendo levar à morte ou esquecimento dos mesmos.

Capítulo 4

Conceitos Teóricos

4.1. Agentes Inteligentes

A definição de Agente Inteligente não é explícita, sendo muitas vezes complicada a elucidação acerca da sua entidade. Atualmente, define-se um agente inteligente como sendo uma figura metafórica e que pode ser aplicado a contextos tais como a Economia ou as Ciências de Computação. No presente trabalho aprofunda-se a utilização destes agentes no contexto das Ciências da Computação.

Um agente inteligente tem como particularidade a resolução de problemas, através da pesquisa consecutiva da melhor ação existente. Estes agentes são utilizados a larga escala em toda a internet (sites comparadores de preços, mecanismo de busca, aplicação de filtros de pesquisa). Esta definição levou a outra noção, em termos de evolução, os Sistemas Multiagente. ^[3]

4.2. Sistemas Multiagente

Os SMA têm propriedades como autonomia, reatividade, pro-atividade e sociabilidade. Além disso, têm um objetivo final a alcançar, que pode ser comum a todos os agentes ou não. ^[1]

O grande interesse destes sistemas é a interação entre os agentes, apesar da sua autonomia. A comunicação entre os agentes deve ser feita através de um meio físico, neste caso, do JADE. É também necessária a definição de uma linguagem de comunicação (FIPA ACL), que deve ser comum a todos os agentes. ^[2]

4.3. Construção de Agentes - JADE

A Framework JADE é um software que tem como base a linguagem de programação JAVA com a particularidade de ser baseada em agentes. Na sua essência foi criado para simplificar a implementação de sistemas multiagentes. Este Framework é muitas vezes visto como um *middleware* para os agentes. É uma ferramenta bastante interessante para o programador uma vez que tem ferramentas gráficas para facilitar a visualização dos acontecimentos, como por exemplo, o *sniffer*. ^[3]

4.3.1. *Containers*

Em ambiente JADE, um *container* é interpretado como um contentor onde serão alocados todos os agentes. Cada instância que é executada no JADE é um *container*, e ao conjunto de todos os *containers* dá-se o nome de *Platform*.

Em cada projeto JADE existem um *main container* que deve sempre estar ativo na plataforma. Todos os outros que são criados ao longo do projeto devem estar na plataforma registados como (*normal*) *containers*. Para que todos estejam na mesma plataforma os *normal containers* devem ser guiados até ao *main container*.

No *main container* pode encontrar-se três tipos de agentes: MAS (*Agent Management System*), DF(*Director Facilitator*) e ainda RMA(*Remote Management Agent*).^[4]

4.3.2. *Agents*

A criação dos agentes é uma das ações mais facilitadas pela utilização do JADE. Assim sendo, nesta etapa apenas é necessária a extensão da classe *Agent*. Ainda assim, para estes terem qualquer tipo de ação é necessário a adição de um tipo de *behaviour*, na programação do seu Setup (). Para além desta especificação, a cada *agent* será associado um identificador pois apenas assim será possível a sua distinção.

4.3.3. *Behaviours*

Os *behaviours* definem o comportamento dos agentes. Existem três grandes categorias de *behaviours*, são elas: *Generic Behaviours*, *One-shot behaviours* e ainda *Cyclic behaviours*. Em categorias mais específicas, existem ainda as categorias: *Ticker behaviour* e o *Waker behaviours*.

4.3.3.1. *Cyclic Behaviours*

Cyclic Behaviours é uma das categorias mais utilizadas para a construção de sistemas multiagente. A sua capacidade de espera atenta permite que o agente associado fique à escuta até receber uma mensagem.^[2]

4.3.3.2. *One-Shot Behaviours*

Os *One-shot Behaviours* são utilizados quando se necessita que um agente, associado a uma determinada ação, execute o pretendido e pare a sua ação de seguida. ^[2]

4.3.3.3. *Ticker Behaviours*

Os *Ticker Behaviours* são associados a comportamentos repetidos em intervalos de tempo. Apresentam os métodos `action()`, `done()` e o `onTick()`.^[2]

Capítulo 5

Implementação

5.1 Arquitetura

A arquitetura do presente trabalho é composta por três grupos de agentes, nomeadamente, utentes, farmácia e fornecedores. Assim sendo, este sistema multiagente é constituído pelas seguintes classes:

- CriarFornecedores;
- CriarUtentes;
- FarmaciaC;
- Fornecedor;
- MediamentosU;
- MedicamentosForn;
- Utente.

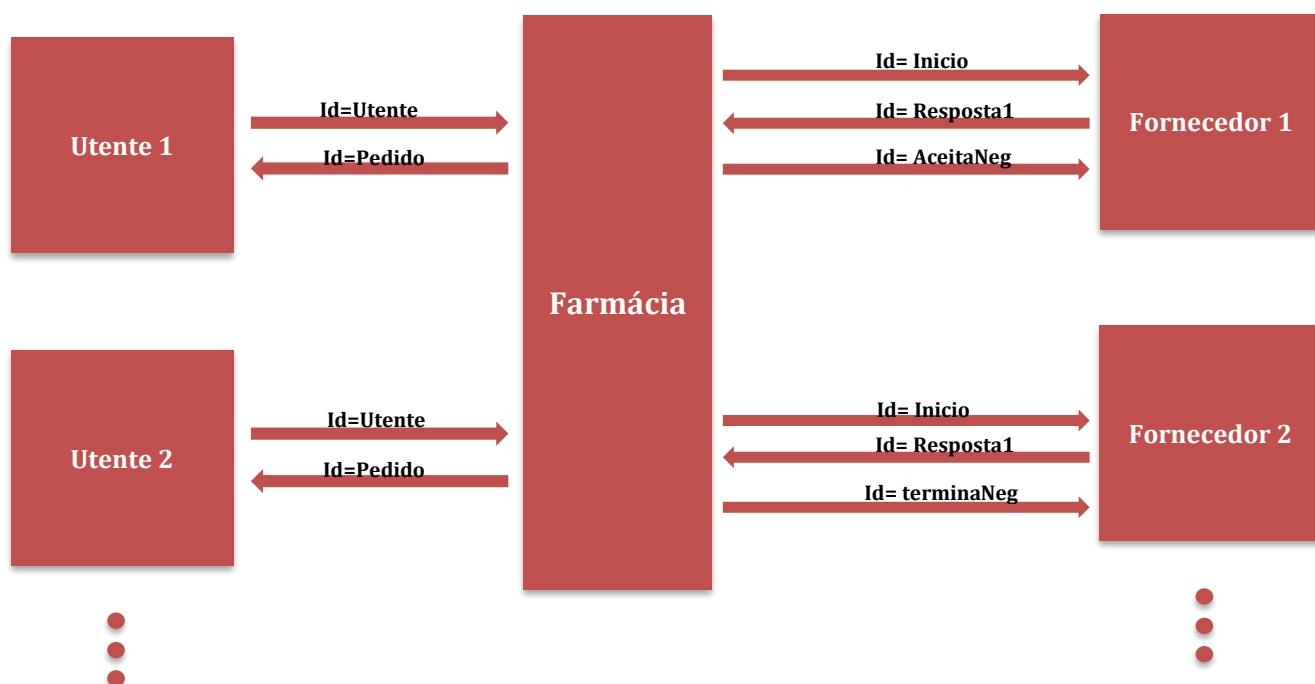


Figura 1-Arquitetura geral do sistema multi-agente.

5.1.1 Farmácia

O agente farmácia tem contacto com os outros agentes do sistema, quer com os utentes, quer com os fornecedores. Relativamente aos utentes, ela permite que estes reponham o seu stock pessoal de medicamentos, de acordo com as suas necessidades.

Por outro lado, relativamente ao fornecedor, esta pode iniciar uma negociação de medicamentos sempre que algum medicamento do seu stock esteja abaixo da quantidade mínima (5 unidades). Além disso, pode aceitar possíveis ofertas de medicamentos desde que o seu stock final não ultrapasse o valor máximo de 150 unidades. Qualquer proposta cujo o tempo de entrega seja superior a 20 dias é rejeitada, ou seja, será aceite apenas a proposta com o menor preço desde que o seu tempo de entrega seja inferior a 20 dias.

5.1.2 Fornecedor

Para a criação do agente fornecedores é necessário a aplicação de três classes: a classe (Fornecedor) que representa o comportamento de um fornecedor, a classe (CriarFornecedores) responsável por gerar vários fornecedores e a classe (MedicamentosForn) responsável pelas características dos medicamentos de cada fornecedor.

Cada fornecedor tem o seu próprio stock de medicamentos que, por sua vez, apresentam diferentes nomes, preços, quantidades e tempos de entrega, sendo os três últimos gerados de forma aleatória(*random*).

Cada fornecedor recorre a um *Cyclic Behaviour* para estar sempre à escuta de possíveis pedidos por parte da farmácia. Ao iniciar uma nova negociação, todos os preços e tempos de entrega de cada medicamento são alterados de forma aleatória, aumentando a concorrência entre os fornecedores.

Recorre ainda a um *Ticker Behaviour* para questionar periodicamente a farmácia se pretende adquirir mais medicação.

5.2.3. Utente

Para a criação do agente utentes é necessário a aplicação de três classes: a classe (Utente) que representa o comportamento de um utente, a classe (*CriarUtentes*) responsável por gerar vários utentes e a classe (*MedicamentosU*) responsável pelas características dos medicamentos de cada utente.

Cada utente tem o seu próprio stock de medicamentos que, por sua vez, apresentam diferentes nomes e quantidades, sendo a quantidade gerada de forma aleatória(*random*).

Cada utente recorre a um *Ticker Behaviour* para administrar periodicamente a quantidade de medicamento que necessita, de acordo com a sua prescrição, contactando a farmácia quando não possuir a quantidade mínima (1 unidade) ou quando não tiver quantidade suficiente para a toma. É de notar que os utentes têm períodos de toma distintos.

Capítulo 6

Demonstrações

6.1. Iniciação do Programa

A figura 2 representa a ordem pela qual os agentes se iniciam. É importante salientar que os utentes devem ser iniciados depois da farmácia uma vez que estes enviam pedidos à farmácia quando há necessidade de repor stock.

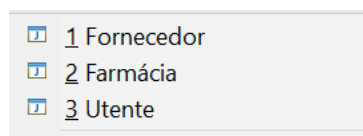


Figura 2-Ordem dos run configuration.

Inicialmente, são criados os diferentes fornecedores, sendo possível ajustar o número dos mesmos. Quando são gerados, é-lhes atribuído um stock distinto, tendo diferentes preços, quantidades e tempos de entrega (figura 3).

```
***Lista de Medicamentos para Fornecedor -> Fornecedor_1***
Medicamento: M_0      Preco: 3      Qtd: 992      Tempo de entrega: 15
Medicamento: M_1      Preco: 5      Qtd: 1867     Tempo de entrega: 0
Medicamento: M_2      Preco: 6      Qtd: 1002     Tempo de entrega: 2
Medicamento: M_3      Preco: 3      Qtd: 1267     Tempo de entrega: 15
Medicamento: M_4      Preco: 6      Qtd: 1777     Tempo de entrega: 18
Medicamento: M_5      Preco: 3      Qtd: 1615     Tempo de entrega: 2
Medicamento: M_6      Preco: 3      Qtd: 958      Tempo de entrega: 16
Medicamento: M_7      Preco: 6      Qtd: 521      Tempo de entrega: 5
Medicamento: M_8      Preco: 4      Qtd: 114      Tempo de entrega: 18
Medicamento: M_9      Preco: 4      Qtd: 232      Tempo de entrega: 7
Medicamento: M_10     Preco: 6      Qtd: 1032     Tempo de entrega: 11
*****
```

Figura 3-Console de criar fornecedores.

De seguida, é criada a farmácia com o seu stock associado (figura 4).

```

*****
*****Farmácia*****
*****Stock da Farmácia*****

Medicamento:      Quantidade:

M_0                82
M_1                95
M_2                82
M_3                11
M_4                77
M_5                4
M_6                19
M_7                34
M_8                14
M_9                47
M_10               0

*****

```

Figura 4-Consola de criar a farmácia.

Por último, geram-se os utentes com stocks e períodos de prescrição distintos (figura 5).

```

*****Stock do Utente -> Utente_0*****
Nome:      M_0   Qtd:   6
Nome:      M_7   Qtd:   4
Nome:      M_1   Qtd:   6
Nome:      M_8   Qtd:   6
*****
*****Stock do Utente -> Utente_1*****
Nome:      M_2   Qtd:   3
Nome:      M_7   Qtd:   3
Nome:      M_1   Qtd:   3
Nome:      M_2   Qtd:   4
Nome:      M_5   Qtd:   3
Nome:      M_9   Qtd:   3
*****

```

Figura 5 Consola de criar utentes.

6.2 Funcionamento do Programa

Na presente secção pretende-se explicar como se sucede o funcionamento do sistema desenvolvido, no qual a comunicação entre agentes é feita por troca de mensagens com diferentes *id's*.

6.2.1 Relação Utentes-Farmácia

Uma vez gerados os utentes, estes começam a contabilizar o seu tempo para a toma de medicamentos, iniciando o seu *Ticker*, tempos esses gerados de forma aleatória.

Cada vez que o *Ticker* é acionado, é selecionado um medicamento e quantidade de prescrição de forma aleatória. Antes de tomar o medicamento, o utente tem de se certificar que, após a toma, a quantidade mínima do medicamento é assegurada. Caso não se verifique, ele realiza o pedido (repondo o seu stock máximo de 30 unidades) à farmácia, se este ainda não tiver sido feito, adicionando-o à lista de espera.

A farmácia ao receber o pedido do utente verifica se tem o medicamento e a quantidade suficiente para satisfazer o pedido. Se a quantidade de medicamento for insuficiente e não nula, a farmácia envia o que tem e regista a quantidade em falta para enviar posteriormente. Pelo contrário, se a quantidade em stock for nula, regista o pedido, não sendo possível enviar qualquer quantidade. Ao receber resposta ao seu pedido, o utente atualiza o seu stock.

<pre> *****Atualizacao (Utente_2)***** Agent : Utente_2 Medicamento : M_7 Qtd Antes : 4 Qtd Depois : 1 </pre> <p>A</p>	<pre> *****Mensagem do Utente_2***** Pretende adquirir o Medicamento: M_7 Com a quantidade: 29 Medicamento: M_7 Quantidade Inicial: 48Quantidade Final: 19 Pedido enviado </pre> <p>B</p>	<pre> ***Atualizao Farmacia Utente_2*** Agent : Utente_2 Medicamento : M_7 After Qtd : 31 </pre> <p>C</p>
---	--	---

Figura 6 - A. Após a toma, o utente fica com quantidade inferior ao mínimo. **B.** Farmácia recebe o pedido do utente e responde. **C.** O Utente atualiza o stock com o que recebeu.

6.2.2 Relação Farmácia-Fornecedor

Uma vez que a farmácia é responsável por garantir que os utentes têm medicamentos suficientes para as suas prescrições, esta tem também de garantir que tem stock suficiente, recorrendo ao fornecedor quando necessário.

Periodicamente a farmácia verifica se nenhum medicamento tem quantidade inferior ao mínimo (5 unidades), tendo inicia uma negociação com os fornecedores, caso esta não tenha sido iniciada anteriormente. Nesta negociação, a quantidade de medicamentos pedida garante que serão entregues aos utentes todas as quantidades pendentes.

Os fornecedores recebem o pedido e verificam se têm o medicamento e a quantidade pedida e responde à farmácia. Caso não tenha medicamento suficiente, a sua proposta não será competitiva (preço e tempos de espera bastante elevados), sendo descartada pela farmácia.

Por fim, a farmácia compara todas as respostas obtidas e opta pelo fornecedor que apresente menor preço e um tempo de entrega inferior a vinte dias, informando-os da sua decisão (figura 7).

```
*****Mensagem do Utente_2*****
Pretende adquirir o Medicamento: M_0 Com a quantidade: 29
Medicamento: M_0 Quantidade Inicial: 17Quantidade Final: 0
Enviar negociação fornecedor_0
Enviar negociação fornecedor_1
Enviar negociação fornecedor_2
Enviar negociação fornecedor_3
Enviar negociação fornecedor_4
*****
Recebi resposta ao meu pedido...
A mensagem vem do: Fornecedor_0e o conteúdo é M_0;75;3;5*****
Recebi resposta ao meu pedido...
A mensagem vem do: Fornecedor_2e o conteúdo é M_0;75;9;13*****
Recebi resposta ao meu pedido...
A mensagem vem do: Fornecedor_1e o conteúdo é M_0;75;7;14*****
O melhor fornecedor é Fornecedor_0e o seu preço é 3e o tempo de entrega é 5
Medicamento: M_0Quantidade 75
*****Atualizacao*****
Medicamento: M_0 Qtd Final: 75

Recebi Pedido de negociação da farmácia
Conteúdo : M_0;75*****
Recebi Pedido de negociação da farmácia
Conteúdo : M_0;75A farmácia está a pedir unidades: 75do medicamento M_0
Preço 3
A farmácia está a pedir unidades: 75do medicamento M_0
Preço 9
*****
Recebi Pedido de negociação da farmácia
Conteúdo : M_0;75A farmácia está a pedir unidades: 75do medicamento M_0
Preço 7
A farmácia aceitou a proposta.

*****Já atualizei o stock*****
Fornecedor_0 Nome: M_0 Preço: 3 Qtd: ( 950 ) -> 875
A farmácia não aceitou a proposta...
A farmácia não aceitou a proposta...
*****
```

A **B**

Figura 7- A. A farmácia inicia uma negociação e recebe resposta do fornecedor. **B.** O fornecedor responde ao pedido de negociação da farmácia.

Para além na negociação invocada pela farmácia, o fornecedor pode ser autónomo e questionar a farmácia se pretende adquirir um determinado medicamento informando a sua disponibilidade e quantidade para envio. A farmácia só aceita se o tempo de entrega e o preço forem inferiores ao limite máximo e se o stock para esse medicamento for inferior ao mínimo (figura 8).

```
Vou perguntar a farmácia se necessita de medicamentos...
Resposta : A farmácia não está interessada em negociar...
Vou perguntar a farmácia se necessita de medicamentos...
Resposta : A farmácia não está interessada em negociar...
Vou perguntar a farmácia se necessita de medicamentos...
Resposta : A farmácia não está interessada em negociar...
*****

*****
A mensagem vem do: Fornecedor_0 Nao preciso do medicamento pedido
A mensagem vem do: Fornecedor_1 Nao preciso do medicamento pedido
A mensagem vem do: Fornecedor_2 Nao preciso do medicamento pedido
```

A **B**

Figura 8- A. O fornecedor questiona a farmácia se necessita de medicamento e recebe a resposta. **B.** a farmácia responde ao fornecedor, rejeitando a proposta.

As trocas de mensagens descritas anteriormente encontram-se representadas na figura 9. A figura mostra a janela *Sniffer* obtida pelo JADE, apresentando as diversas trocas de mensagens entre três utentes, três fornecedores e uma farmácia.

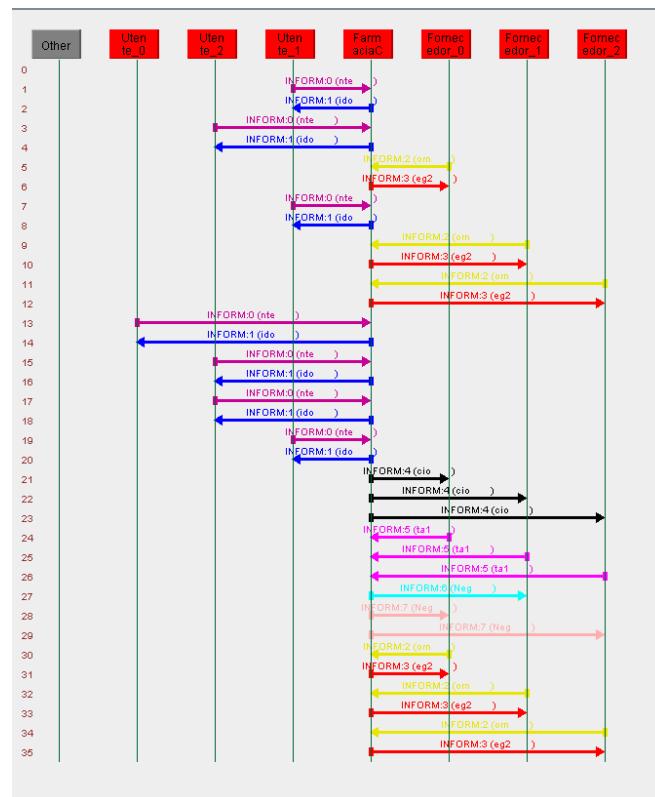


Figura 9-Sniffer Agent.

Capítulo 7

Conclusões

O sistema multiagente desenvolvido simula um sistema de Farmácia interativa. O principal objetivo do sistema passa por garantir que os utentes têm medicamento suficiente para realizar a ingestão de medicamentos segundo as suas prescrições, sendo para isso necessário que os utentes peçam medicamentos à farmácia quando necessário. Por outro lado, a farmácia negocia medicamentos e as respetivas quantidades, preços e tempos de envio com os fornecedores. Assim sendo, o sistema procurou recriar casos em que apenas uma farmácia interage com diferentes agentes, com vários utentes e com vários fornecedores.

O sistema final está de acordo com os resultados esperados, todas as mensagens estão a ser trocadas corretamente, levando a uma correta interação entre os agentes. Ou seja, quer os utentes, quer a farmácia obtêm resposta aos seus pedidos, conseguindo gerir o seu stock pessoal. Além disso, as negociações entre farmácia e fornecedor estão a ser feitas de forma a respeitar os parâmetros definidos pela farmácia.

Contudo, o sistema apresenta algumas limitações, nomeadamente no número de agentes envolvidos e na realização de pedidos de medicamentos. Por um lado, aumentando o número de utentes e fornecedores, a farmácia acaba por não ter capacidade de responder a todas as mensagens. Por outro lado, a realização de pedidos não é realizada de forma mais eficiente, os utentes e a farmácia solicitam apenas um medicamento por pedido, o mais correto seria solicitar um conjunto de medicamentos em cada pedido realizado à farmácia.

Em suma, o sistema multiagente desenvolvido satisfaz os requisitos pedidos, simulando um sistema de uma farmácia interativa, a qual interage com utentes e fornecedores, que garante uma correta gestão de stocks respondendo a pedidos e fazendo negociações.

Capítulo 8

Bibliografia

- [1] Sistemas Multi-Agente: <https://paginas.fe.up.pt/~lpreis/Tese/Capitulo3.PDF>, visitado em 25/04/2018.
- [2] Bellifemine F., Caire G., Greenwood D., *Developing Multi-Agent Systems with JADE*, John Wiley & Sons, ISBN:978-0470057476, 2007.
- [3] Novais,P.,Analide,C., *Agentes Inteligentes*, Universidade do Minho, 2006.
- [4] Caire,G. *Jade tutorial jade programming for beginners*, Telecom Italia,2007.