Agentes e Sistemas Multiagente

Relatório do Trabalho Prático

MEI - 2023/2024

Grupo 2

Hugo Martins
A95125

João Escudeiro
A96075

José Rocha
A97270

Universidade do Minho

Índice

1 Introdução

Sistemas multiagente são sistemas que recorrem a Agentes Inteligentes, entidades autónomas capazes de compreender os ambientes em que são inseridos e tomar decisões com base em objetivos definidos. Estes sistemas são provenientes da área de Inteligência Artificial Distribuída e, embora estes sistemas sejam distribuídos pelos agentes, cada agente contribui para o objetivo do sistema como um todo. São diversas as áreas que utilizam sistemas multiagente quer para modelação ou simulação de comportamentos.

No âmbito da componente prática da Unidade Curricular de *Agentes e Sistemas Multiagente*, foi-nos solicitado a implementação de um sistema multiagente, baseado no contexto escolhido pelo grupo para o Trabalho de Investigação, utilizando a biblioteca SPADE para o desenvolvimento de agentes.

Para tal, foi necessário conceber e desenvolver uma arquitetura distribuída para gerir um sistema multiagente.

Ao longo deste relatório, serão mencionados os conceitos fundamentais do sistema multiagente, as normas FIPA utilizadas para a implementação do mesmo, bem como a arquitetura e o modo funcionamento do sistema desenvolvido. Serão também discutidos os desafios encontrados durante o processo de desenvolvimento e as estratégias adotadas para superá-los.

2 Definição do Domínio

O domínio escolhido para este trabalho prático tem como base os transportes inteligentes, tendo em conta a escolha do grupo no Trabalho de Investigação. Os transportes inteligentes constituem uma área em constante crescimento que visa melhorar a eficiência, segurança e sustentabilidade dos sistemas de transporte através da integração de tecnologias avançadas, como inteligência artificial, sensores e comunicação entre dispositivos.

O sistema concebido concentra-se na simulação de um porto marítimo baseado em agentes, com o objetivo de simular a gestão do tráfego portuário, incluindo a chegada e o atracar de embarcações, partida das mesmas, ocupação dos canais de acesso ao porto e operações de carga e descarga.

Decidimos construir uma figura que simula o modo de funcionamento do porto, para que o leitor entenda de uma forma simples e sugestiva, a forma como o porto funciona.

Qualquer barco que pretenda realizar alguma ação, seja atracar ou abandonar o porto, comunica com o **Farol(*LightHouse*)** que é o maior responsável por estas ações. O farol gere os canais que dão acesso ao porto, e tudo o que sejam tarefas relacionadas com o atracar dos barcos, o farol encaminha para o **Gestor de Cais(CaisManager)**, representado pelo edifício em baixo na figura.

Existem 3 tipos de Barcos: **Private, Cargo Transport ou Passenger Transport**. Os barcos do tipo Private podem atracar apenas em cais normais, enquanto que os barcos Cargo e Passenger podem atracar apenas em cais de

Cargas e Descargas(Comerciais).

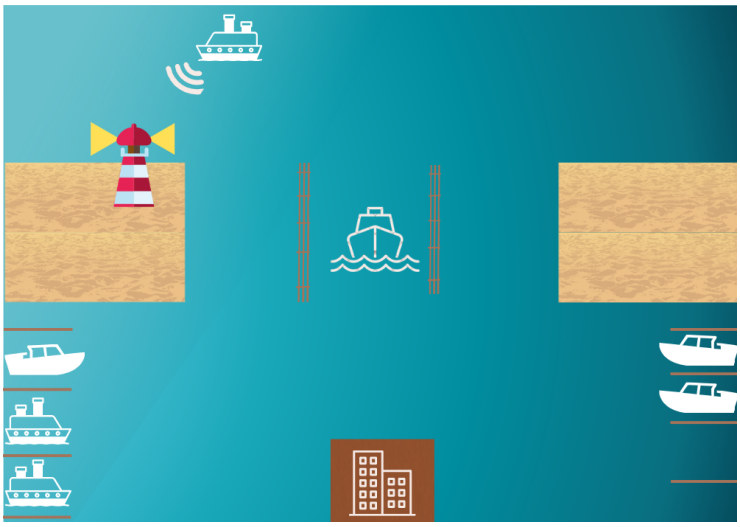

Figure 1: Esquema do Porto Marítimo.

3 Sistema Multiagente

Neste capítulo será apresentada uma descrição do sistema multiagente desenvolvido, acompanhada de diagramas que permitirão uma melhor compreensão da sua arquitetura e funcionamento. A modelação da arquitetura foi feita com base no problema anteriormente descrito, tendo-se assim tentado idealizar todos os componentes necessários para construção do sistema. Ao longo deste capítulo serão também analisadas as interações entre os agentes que constituem o sistema e que garantem a integridade do mesmo.

3.1 Agentes

Como principais agentes do sistema multiagente temos:

Agente	Função
Boat	Este agente comunica com o *LightHouse*, solicitando a operação que pretende realizar que pode ser atracar na marina ou, caso esteja atracado na mesma, abandoná-la.
LightHouse	O agente mais importante do sistema. Recebe os pedidos dos barcos, gere a utilização dos canais, envia informação do estado da marina para o *MarinePolice*, e encaminha os pedidos de parque para o*CaisManager*. Além disto, este agente também gere a fila dos barcos à espera e as alterações climáticas. Todas as interações passam por este agente.
CaisManager	Agente que recebe os pedidos de parque, redirecionados pelo *LightHouse* e gere a utilização dos cais.
MarinePolice	Agente que de 2 em 2 segundos faz um pedido ao *LightHouse*, solicitando informações acerca do estado atual da marina, e atualiza a informação no terminal.

Table 1: Função de cada um dos agentes no sistema.

3.2 Arquitetura do Sistema

Entender a arquitetura do sistema tornou-se importante, pois possibilitou uma melhor organização do projeto e das estruturas a serem criadas.

3.2.1 Diagrama de Classes

O primeiro diagrama que decidimos elaborar foi o diagrama de classes, que nos permitiu uma melhor organização entre os agentes e as classes constituintes e essenciais ao funcionamento do sistema. O nosso sistema multiagente é constituído por uma classe principal *Marine* que é responsável por agregar todos os agentes que compõem o sistema desenvolvido. Como mencionado acima, existem 4 agentes **Marine Police**, **Boat**, **CaisManager** e ***LightHouse***.

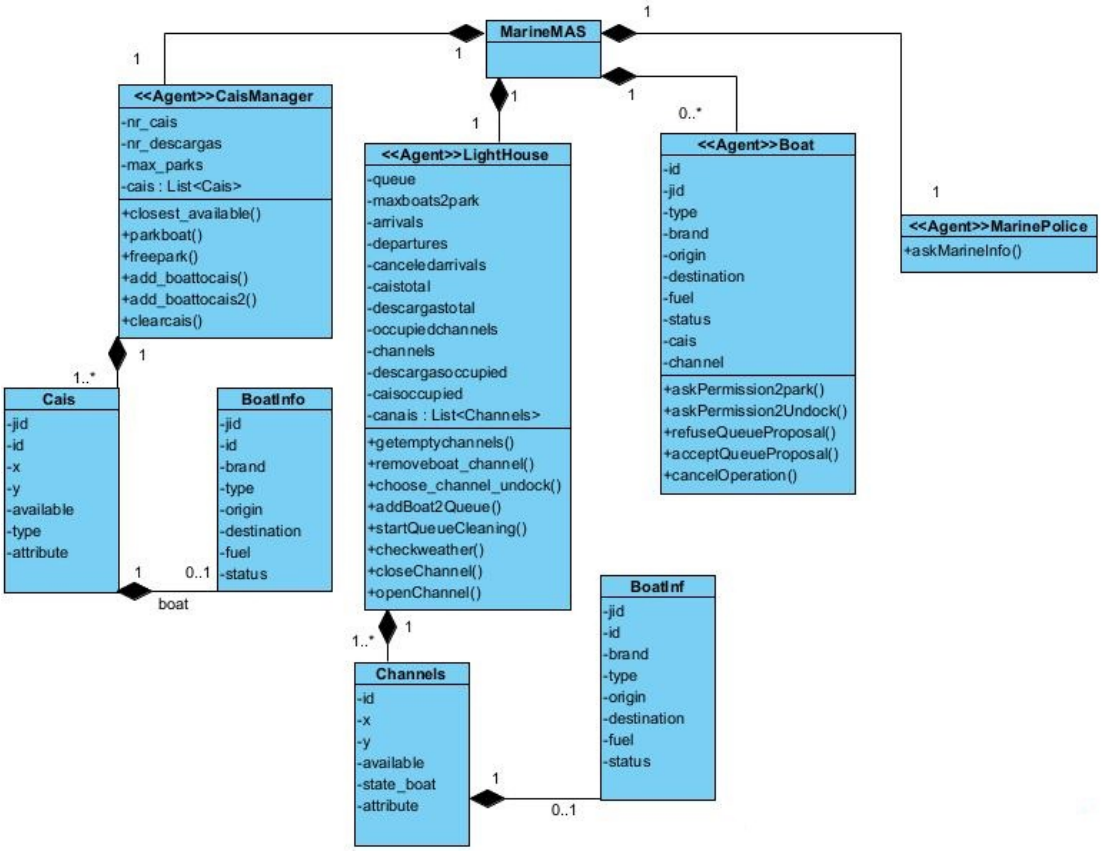

Figure 2: Diagrama de Classes.

Cada barco comunica com o *LightHouse*, fornecendo-lhe as seguintes informações:

- **jid**: Identificador do Barco
- **Type**: Tipo do Barco (*Private*, *Cargo Transport* ou *Passenger Transport*)
- **Brand**: Marca do Barco
- **Fuel**: Quantidade de combustível restante
- **Status**: Informa se o barco pretende atracar ou desatracar
- **Cais**: Cais em que o barco se encontra
- **Channel**: Canal em que o barco se encontra

O agente principal *LightHouse* é responsável por receber os pedidos, e orientar os mesmos, mediante o tipo de pedido. Este também regista a lista de barcos

em fila de espera na **Queue**, que pretendem utilizar os canais, gerindo o número máximo de barcos que podem estar à espera na fila.

Quando o agente *LightHouse* recebe uma solicitação de um barco para atracar, este verifica quais os canais que estão livres no momento. Sempre que um barco é atribuído a um canal, este fica "not available" até que o mesmo a abandone. Assim, quando existe um pedido de um barco, é feita a verificação da disponibilidade dos canais e dos cais. Caso haja disponibilidade, o farol escolhe um canal (**choose_channel_undock()**) e, se for o caso, redireciona o pedido para o gestor de cais, que é responsável por escolher o cais mais próximo através da função (**closest_available()**). Caso não haja disponibilidade de canais ou de cais, é proposto ao barco a possibilidade de ir para a fila de espera, podendo o mesmo aceitar ou não. Para a gestão dos cais, o *CaisManager* guarda informações sobre estes, entre as quais o número total de cais, o número de cais comerciais e o número de cais privados. É este que adiciona e remove os barcos do seu cais. Para além de todo este processo, o agente *LightHouse* gere a situação meteorológica que é atualizada de 15 em 15 segundos. Decidimos incluir quatro estados de tempo diferentes: **Cloudy**, **Stormy**, **Sunny** e **Rainy**. Caso o estado seja **Cloudy** ou **Sunny**, os canais que dão acesso ao porto funcionam de forma normal.Caso o tempo seja **Stormy**, todos os canais que não estão a ser ocupados por um barco, no momento em que acontece a alteração do tempo ficam encerrados até que o tempo volte a ficar **Cloudy** ou **Sunny**. Por fim, se o tempo for **Rainy**, e caso exista um total de canais superior ou igual a 2, o farol encerra um canal para garantir uma maior segurança dos barcos.

Os estados possíveis do tempo são: **Rainy** (se houver mais do que 1 canal livre ele fecha 1), **Stormy**(fecha todos os canais) ou então **Sunny** ou **Cloudy** em que os canais funcionam de forma normal.

Para otimizar o funcionamento da marina, são armazenadas coordenadas X e Y relativas aos cais e aos canais. Assim, cada Barco é encaminhado de forma a percorrer a menor distância possível, mediante a disponibilidade dos mesmos.

3.2.2 Diagrama de Colaboração

De seguida, desenvolvemos um diagrama de colaboração, com o objetivo de mostrar as possíveis interações entre os agentes e quais os comportamentos que deveriam ser implementados. As comunicações da figura abaixo estão ordenadas pela ordem pela qual podem aparecer.

Como é possível observar na figura, foi necessário implementar alguns tipos de comunicação (*behaviours*) entre os quais:

- Um barco solicita autorização para entrar/abandonar o porto
- Caso um barco pretenda atracar no porto, o *LightHouse* pergunta ao *CaisManager* se existem cais daquele tipo disponíveis
- O *LightHouse* avisa um barco que tem permissão para usar um canal e a qual cais se deve dirigir

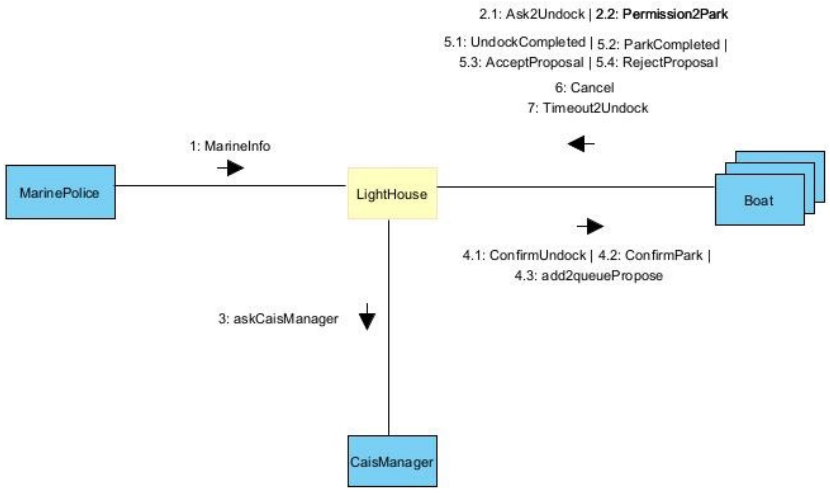

Figure 3: Diagrama de Comunicação.

- Em caso de impossibilidade de realizar a operação do imediato o Farol propõe ao barco uma possível entrada na fila de espera, à qual o barco responde de uma forma afirmativa ou negativa
- Caso esteja à espera há demasiado tempo, o barco pode solicitar ao Farol o cancelamento da operação
- Uma vez terminada a operação, o barco avisa o Farol acerca do sucesso da operação
- Quando um barco já se encontra atracado há algum tempo no cais, o mesmo informa o *LightHouse* que o seu tempo nele terminou e que pretende desatracar
- Periodicamente, o agente MarinePolice solicita informações sobre o estado atual da marina ao Farol

3.3 Modo de Funcionamento

Para além da compreensão da arquitetura, é necessário entender a forma como se processa o funcionamento do sistema multiagente desenvolvido.

3.3.1 Protocolo de Mensagens

De uma forma similar à que foi lecionada nas aulas, decidimos utilizar objetos para o envio de mensagens. Para tal, criamos uma classe **MessageInfo**, que contém o tipo da mensagem, e a informação do barco. Este objeto é utilizado em todas as trocas de mensagens no sistema, em que no tipo da mensagem é espeficado o intuito da mesma, e no **boat_info()** a informação sobre o barco

em causa. Sempre que este objeto é associado ao body de uma mensagem, o mesmo é feito através da biblioteca *jsonpickle*, codificando a mensagem. No recetor acontece de uma forma semelhante, sendo que este terá de descodificá-la para ter acesso ao conteúdo da mesma.

3.3.2 Diagrama de Atividades

Para melhor compreendermos a direção dos fluxos, e representarmos de forma organizada as possíveis ações que um barco pode tomar, construímos um diagrama de atividades que resume o modo funcionamento do sistema, presente na Figura 4. É através deste diagrama que conseguimos compreender as ações de cada agente, e a importância de cada um deles no funcionamento global do sistema.

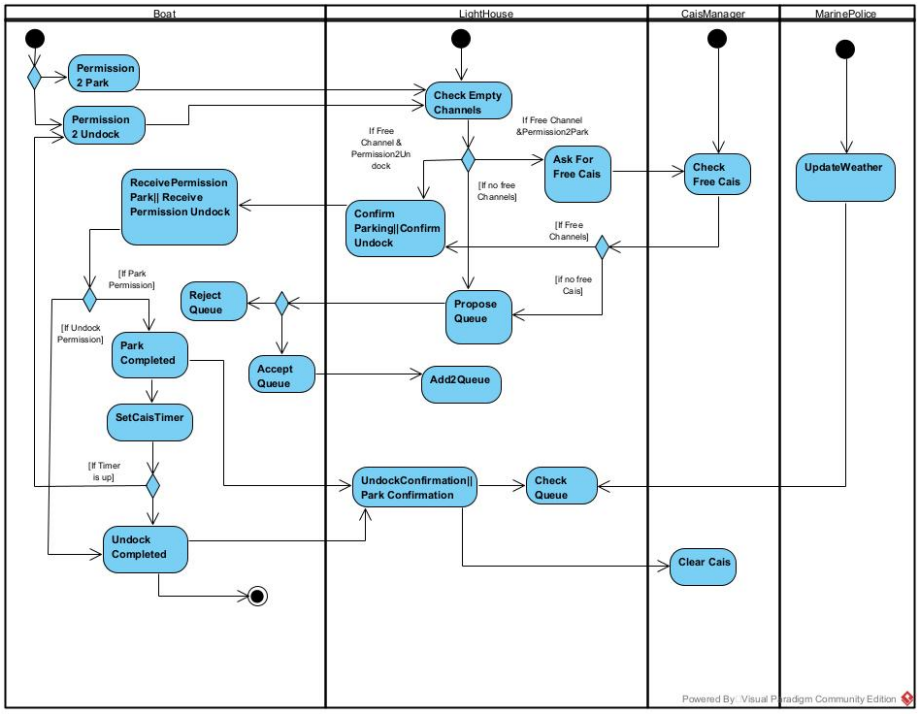

Figure 4: Diagrama de atividades.

3.3.3 Diagramas de Sequêcia

A integração dos diagramas de sequência foi fundamental para entender a ordem que envolve os processos de atracar no cais, desatracar e solicitação de informações nos portos marítimos. Estes diagramas fornecem uma representação dinâmica e detalhada das interações entre os diversos elementos do sistema, promovendo uma gestão mais eficiente das operações portuárias.

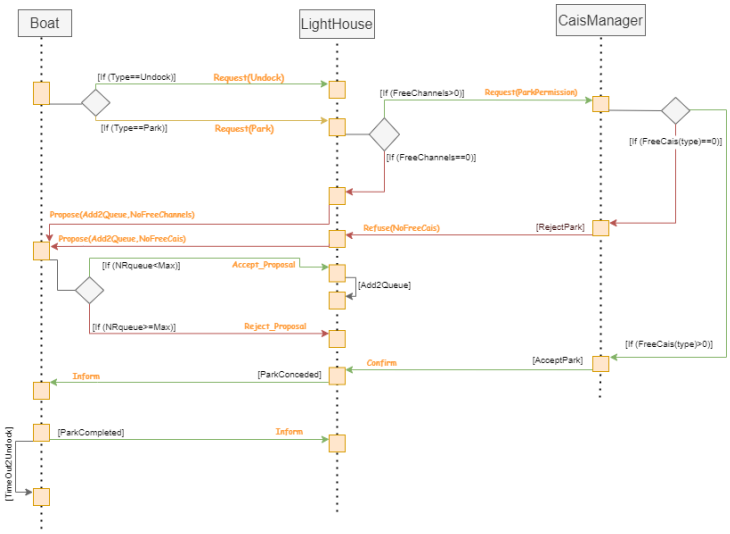

Figure 5: Diagrama de sequência do comportamento de atracar.

Na Figura 5, é percetível a sequência que é seguida ao longo do sistema, bem como as performatives associadas a cada uma das mensagens enviadas. Para o comportamento de parque, o barco solicita operação de atracar ao farol. Caso tenha canais livres, este pergunta ao gestor de cais se há cais disponíveis. Caso todas as respostas sejam afirmativas, o farol fornece permissão ao barco, que ocupa o canal indicado por um intervalo de tempo. Caso contrário, envia uma proposta para entrar na **Queue**. O barco, aceita ou não a fila, mediante o número de elementos que estão na fila no momento, e o máximo que poderão estar. Caso um barco esteja na fila há algum tempo, ele pode iniciar um cancelamento, enviando esta informação para o farol. Quando um barco termina a sua operação de parque, este avisa o Farol que a operação foi bem sucedida e inicia um comportamento para *Undock*, após um timeout especificado.

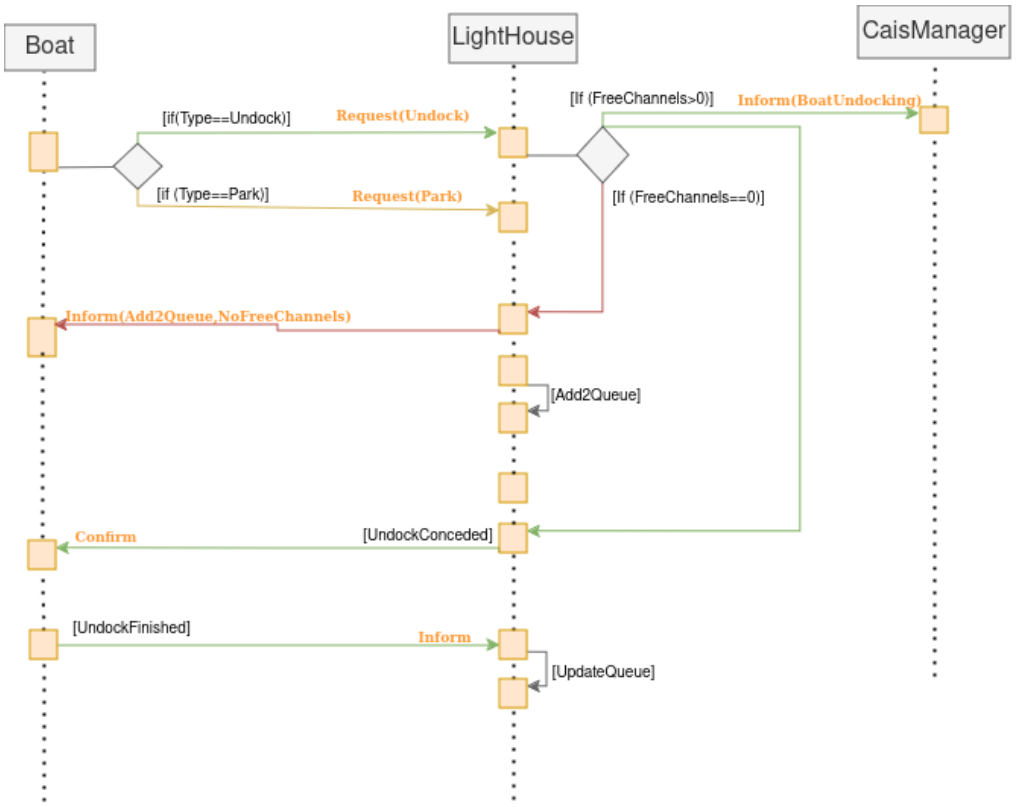

Figure 6: Diagrama de sequência do comportamento de Desatracar.

Na Figura 6, podemos observar como funciona o comportamento de desatracar. O barco solicita ao farol a permissão. Este responde afirmativo ou informação sobre adição automática à fila mediante a disponibilidade dos canais. Caso seja uma resposta negativa, então o farol informa que o adicionou de imediato à fila, para garantir que este barco vai acabar por abandonar o porto, para evitar o congelamento do cais no qual está atracado.

Quando o barco termina a operação, informa o farol sobre o sucesso da operação, que inicia um comportamento que desencadeia a limpeza da **Queue**, tendo em conta que um cais ficou livre.

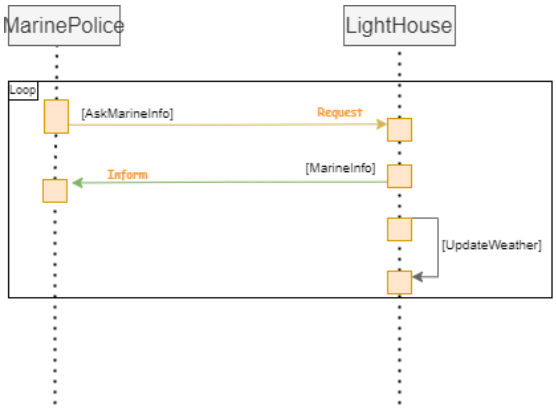

Figure 7: Diagrama de sequência do comportamento do agente MarinePolice

Na Figura 7, está representado de forma simples como acontecem as mensagens de solicitação sobre o estado atual do porto, do lado do agente da polícia. Isto ocorre num loop de 2 em 2 segundos. Na Figura está também representado o comportamento de atualização do estado climatérico de 15 em 15 segundos.

3.3.4 Mecanismo de Negociação

O mecanismo de negociação utilizado foi bastante simples. Sempre que um barco pretende entrar no porto para atracar, ele propõe esta entrada ao *LightHouse*. Caso não haja canais ou cais livres, o Farol envia uma proposta de queue indicando o número máximo de elementos que podem estar na fila e o número de elementos que estão na fila naquele momento. O barco, por sua vez verifica se este número lhe agrada e em caso positivo aceita a fila, caso contrário recusa. No caso de um barco pretender sair do porto, este aceita sempre a fila.

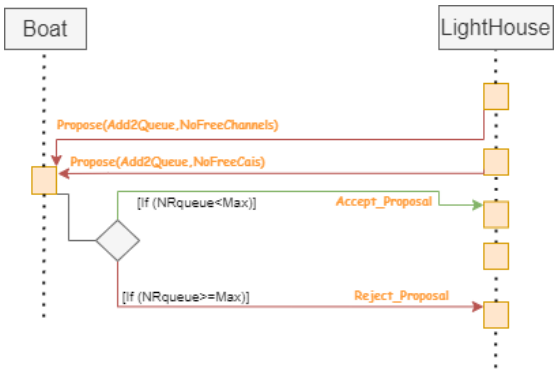

Figure 8: Diagrama do mecanismo de negociação

3.3.5 Funcionamento de alguns behaviours

Detalhes de funcionamento de alguns behaviours e casos específicos

Behaviour	**Tipo**	**Descrição**
ParkFinished	TimeoutBehaviour	Comportamento que é adicionado a um barco quando lhe é concedida permissão para atracar. O tempo de operação no canal é 10 segundos.
Cancel	TimeoutBehaviour	Quando um tempo de 4 segundos na fila é excedido, o barco envia uma mensagem de cancelamento para o farol.
TimeoutToUndock	TimeoutBehaviour	Timer de 15 segundos definido quando um barco atraca no porto. Após esses 15 segundos, o barco pede para desatracar.
UndockFinished	TimeoutBehaviour	Tempo de operação no canal é 10 segundos. Quando acaba este comportamento, o barco envia confirmação para o Farol.
AddTOrandCais	OneShotBehaviour	Behaviour executado no início da simulação para atribuir um cais aos barcos que são definidos para pedir para desatracar.
Marine_Info	PeriodicBehaviour	O agente MarinePolice solicita frequentemente ao farol, informação sobre os barcos e sobre o estado atual do porto.
WeatherForecast	PeriodicBehaviour	Comportamento que de 15 em 15 segundos atualiza os canais livres, mediante um estado climatérico definido aleatoriamente.

Table 2: Tabela com a descrição de alguns behaviours.

3.3.6 Ficheiro de Configuração

Com o objetivo de facilitar e acelerar o arranque do sistema multiagente projetado, decidimos criar um ficheiro de configuração com o nome "*settings.json*", onde são definidas as variáveis que contêm a informação do XMPP domain name, a password do XMPP, bem como o número de barcos que inicialmente se encontram a chegar e a abandonar o porto, o número de canais que dão acesso ao porto, o número de cais privados e de cargas e descargas e o número máximo

de barcos que pretendem atracar no porto que poderão à espera na queue. Por fim decidimos adicionar um campo *weather*, que remete para o estado do tempo inicial, quando o sistema arranca.

```
{
  "jid": @xmpp,
  "pass": "NOPASSWORD",
  "arriving": 10,
  "leaving": 2,
  "channels": 3,
  "cais": 10,
  "descargas": 2,
  "maxboats2park": 16,
  "weather": "Rainy"
}
```

Figure 9: Exemplo da estrutura de um Ficheiro de Configuração

4 Resultados Obtidos

Sempre que necessário o agente responsável por apresentar as informações do porto (Marine Police) imprime-as no terminal.

```
Marine Info at 17:50:53.866138: 0
****************************************************************
Cais Privados: 3                              Weather: Cloudy

Cais Occupied: 0

Commercial Cais : 2

Commercial Cais Occupied: 0

Queue: Empty

Boats In Operation:

    Channel 0. Empty
    Channel 1. Empty

****************************************************************
```

Figure 10: Informações sobre o Porto ao Iniciar o Sistema.

De 2 em 2 segundos estas informações são atualizadas no terminal com os detalhes dos barcos que chegam pedindo para atracar, ou que estão no porto e pretendem abandoná-lo.

No cenário de teste iniciaram-se 3 barcos que inicialmente estão no porto e 10 barcos para chegarem. Desta forma, a seguinte figura representa o estado do porto quando os 3 barcos iniciam o processo de undocking. Dois entram para os canais e um aceita entrar na fila.

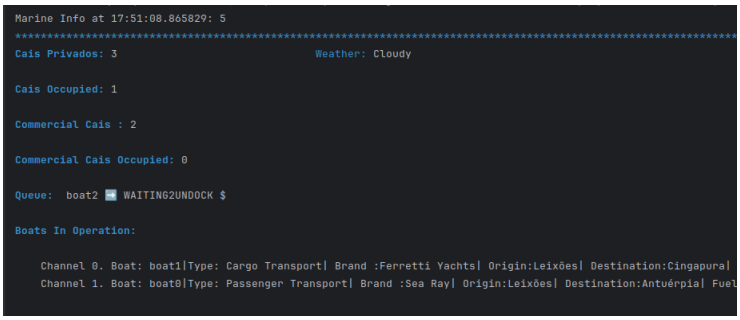

Figure 11: Informações sobre o Porto após três barcos desatracarem.

Após os 3 barcos definidos para inicialmente destacarem no ficheiro de configuração, começam a chegar os 10 barcos que vão atracar no porto.

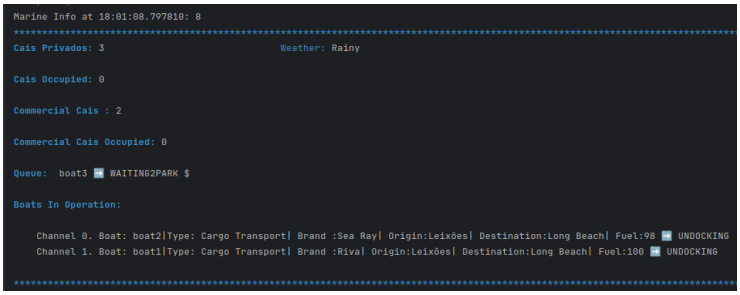

Figure 12: Informações sobre o Porto após chegarem barcos para atracar.

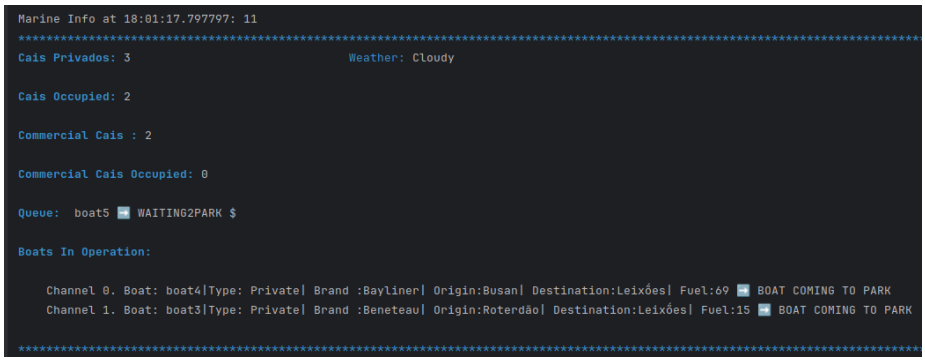

Figure 13: Informações sobre o Porto após dois barcos atracarem.

Como podemos ver na Figura 13, dois barcos ocuparam os canais que dão acesso ao porto e dois cais privados. O terceiro barco que chega vai para a fila de espera.

Figure 14: Informações sobre o Porto após os dois primeiros barcos atracarem, iniciarem o comportamento de desatracar.

No final da execução da simulação, é gerado um ficheiro json com informações acerca dos parques bem sucedidos, parques e operações que foram canceladas quer pelo barco, quer pelo farol devido a ter a fila de espera lotada. Como exemplo, apresenta-se o formato deste ficheiro:

{"Parks":8, "Undocks":11, "Canceled":2}

4.1 Cenários de Teste

Para os cenários de teste foi utilizado o ficheiro de configuração presente na Figura 15.

```
{
  "jid": @xmpp,
  "pass": "NOPASSWORD",
  "arriving": 20,
  "leaving": 5,
  "channels": 2,
  "cais": 10,
  "descargas": 8,
  "maxboats2park": 4,
  "weather": "Sunny"
}
```

Figure 15: Ficheiro de configuração para o cenário de teste.

Os testes foram realizados alterando o valor de número de canais que dão acesso ao porto. Verificou-se que, como seria de esperar, ao aumentar o número de canais, o porto consegue dar resposta ao pedido de mais barcos e diminuir o número de operações canceladas, aumentando a taxa de sucesso. De seguida são apresentadas três simulações: uma apenas com um canal, outra com dois canais e outra com 4 canais. Os resultados são apresentados em baixo:

Teste 1 (1 canal): {"Parks": 6, "Undocks": 11, "Canceled": 14}

Teste 2 (2 canais): {"Parks": 15, "Undocks": 20, "Canceled": 5}

Teste 3 (4 canais) : {"Parks": 20, "Undocks": 25, "Canceled": 0}

Teste	**Taxa de Sucesso**
Teste 1	54%
Teste 2	87.5%
Teste 3	100%

Table 3: Tebela da taxa de sucesso.

A taxa de sucesso é calculada utilizando a fórmula abaixo indicada:

$$\frac{(Parks + Undocks)}{(Parks + Undocks + Canceled)}$$

5 Sugestões e Recomendações

Tendo em conta os resultados obtidos, o sistema multiagente desenvolvido cumpre na íntegra os objetivos propostos no enunciado do projeto. Existem agentes, são utilizados behaviours, são utilizadas as normas FIPA, e foi incluído uma mecanismo de negociação simples.

No entanto, é importante mencionar algumas melhorias para o sistema apresentado. Uma das melhorias possíveis seria a implementação de um retry nos barcos, quando o farol nega a operação. Outra possibilidade seria a possibilidade de negociação e compra de cais, que tornaria o projeto mais robusto. Para além disto, poderia ser incluído uma espécie de prioridade nas filas de espera para que barcos com especificidades especiais possuíssem prioridade em relação a barcos privados.

Outro aspeto a mencionar seria a implementação da interface gráfica do SPADE. Face ao tempo disponível, optamos por uma demonstração simples via terminal, utilizando cores diferentes, o estado atual do porto.

6 Conclusões

Através da elaboração deste trabalho prático, foi-nos possível aplicar e consolidar todo o conteúdo lecionado ao longo do semestre, tanto das aulas teóricas como nas teorico-práticas. Os conceitos crucial inerentes à realização deste trabalho são os agentes, a comunicação, a colaboração, os comportamentos, a negociação, entre outros.

Os objetivos propostos foram cumpridos na íntegra, e o sistema elaborado está inserido numa arquitetura que dá resposta a uma gestão eficiente de um porto marítimo.

O grupo considera que a maior dificuldade ultrapassada foi a organização das perfomatives, e a estruturação do problema. Inicialmente foi difícil saber por onde começar, mas com o decorrer do trabalho o produto final foi surgindo com alguma naturalidade, facilitado pela elaboração de diagramas que ajudavam a perceber melhor como funcionaria a comunicação dentro do sistema.

Concluindo, consideramos que o trabalho desenvolvido é positivo, estando cientes que no futuro existiriam aspetos já foram mencionados, que poderiam ser alvo de melhoria.