



Universidade do Minho

Mestrado Engenharia Informática

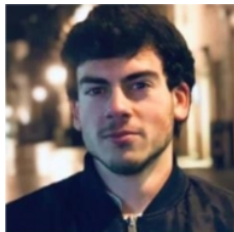
Agentes e Sistemas Multiagente

SISTEMA DE GESTÃO DE AEROPORTOS

Grupo 7

PG50384 Francisco Reis Izquierdo,

PG50528 José Pedro Martins Magalhães



25 de abril de 2023

Conteúdo

1	Introdução	5
1.1	Descrição do Problema	5
1.2	Objetivos	5
1.3	Proposta de Solução	5
2	Definição do Sistema Multiagente	6
2.1	Descrição do Sistema Multiagente	6
2.2	Arquitetura	6
2.3	Definição e Descrição dos Agentes	7
2.3.1	Avião	7
2.3.2	Gestor de Gares	9
2.3.3	Administrador	10
2.3.4	Torre de Controlo	10
2.4	Estrutura do Sistema	11
2.5	Interações dos Agentes	12
2.5.1	Diagrama de Sequência	13
2.5.2	Diagrama de Colaboração	13
2.6	Requisito Adicional	14
3	Resultados	15
3.1	Cenário 1 (Default)	15
3.2	Cenário 2	16
3.3	Cenário 3	17
3.4	Cenário 4	17
4	Análise Crítica	19

4.1	Análise Crítica sobre os Resultados	19
4.2	Análise Crítica sobre o Sistema Multiagente	19
5	Trabalho Futuro e Conclusão	21
5.1	Trabalho Futuro	21
5.2	Conclusão	21

Lista de Figuras

2.1	Arquitetura do Sistema	7
2.2	Máquina de Estados do agente Avião	9
2.3	Diagrama de Packages	11
2.4	Diagrama de Classes	12
2.5	Diagrama de Sequência	13
2.6	Diagrama de Colaboração	13
2.7	Cenário com nevoeiro	14
2.8	Cenário com sol	14
3.1	Cenário 1 (Default)	16
3.2	Cenário 2 (Período de chegada de aviões = 2 segundos)	16
3.3	Cenário 2 (Período de chegada de aviões = 15 segundos)	16
3.4	Cenário 3 (Tempo de aterragem/descolagem = 15 segundos)	17
3.5	Cenário 3 (Tempo de aterragem/descolagem = 3 segundos)	17
3.6	Cenário 4 (Número de gares = 15)	18
3.7	Cenário 4 (Número de gares = 5)	18

Lista de Tabelas

2.1	Tabela de behaviours do agente Avião	8
2.2	Tabela de behaviours do agente Gestor de Gares	9
2.3	Tabela de behaviours do agente Torre de Controlo	11

1. Introdução

O presente relatório surge no âmbito da disciplina de Agentes e Sistemas Multiagente, pertencente ao perfil de Sistemas Inteligentes, no qual foi proposto ao grupo de trabalho a elaboração de um sistema multiagente. Este contém agentes inteligentes capazes de sentirem e perceberem o ambiente em que se inserem e desta forma, os agentes são capazes de adquirir informação e posteriormente tomarem decisões de forma coordenada com outros agentes.

1.1 Descrição do Problema

O trabalho prático proposto pela equipa docente tem por base a implementação de um sistema multiagente cujo objetivo é a gestão dos processos de aterragem e descolagem num aeroporto por parte de aviões. De forma geral, o sistema multiagente contém uma infraestrutura capaz de gerir todas as etapas envolvidas nos processos mencionados, bem como disponibilizar toda a informação relativa ao estado do aeroporto. No âmbito geral, os agentes inteligentes envolvidos atuam de forma cooperativa, no qual em certas etapas realizam processos de negociação e/ou cooperação. Com isto, os agentes envolvidos no sistema multiagente, dada a vertente do problema, são representativos de entidades presentes num aeroporto real, no qual temos os seguintes agentes: **Avião**, **Torre de Controlo**, **Gestor de Gares** e **Administrador**.

1.2 Objetivos

No que diz respeito aos objetivos propostos a atingir com implementação do sistema multiagente apresentado, estes encontram-se em conformidade com o agente definido. Assim, consoante os agentes existentes no sistema e detalhados em 2.3, temos de forma geral como objetivo a simulação dos processos de aterragem e descolagem efetuados por aviões. Agregados a este objetivo, advêm também outros como a simulação dos processos de gestão e/ou negociação de pistas e gares relativos ao aeroporto. Podemos também enaltecer outros objetivos como a simulação de situações adversas presentes no aeroporto como a limitação do número de gares e/ou pistas para aviões. Por fim, pretende-mos analisar as diversas interações entre os agentes envolvidos no sistema, bem como avaliar o estado do mesmo.

1.3 Proposta de Solução

A implementação da solução teve por base a definição da arquitetura do sistema multiagente desenvolvido na linguagem de programação **Python**, no qual através da biblioteca **SPADE**, foram definidos os agentes. Consequentemente, foram definidos os seus **behaviours** e correspondentes **performatives**. Deste modo, conseguimos retratar as diversas tarefas e ações a serem realizadas pelos diversos agentes, além das interações entre estes, consoante o ambiente em que os mesmos se inserem.

2. Definição do Sistema Multiagente

2.1 Descrição do Sistema Multiagente

O sistema multiagente definido caracteriza-se por ser **aberto**, na medida em que o sistema não tem um desenho ou arquitetura prévios, apenas agentes no seu seio, além dos agentes não terem, necessariamente, consciência da existência dos outros (por exemplo, os aviões entre si). Além disto, os agentes e/ou grupos de agentes são concebidos separadamente (modularidade), sendo flexíveis e tolerantes a falhas.

2.2 Arquitetura

No que concerne à arquitetura desenvolvida, convém salientar aspetos relativos à sua tipologia, nomeadamente acerca da localização da tomada de decisão e o seu funcionamento. Relativamente à tomada de decisão, a arquitetura desenvolvida na qual assenta o sistema multiagente descrito é **centralizada**. Nesta arquitetura, é notório a presença do agente central Torre de Controlo, que é responsável pela gestão dos aviões que se encontram a querer aterrar ou descolar no aeroporto, tomando decisões com base nas informações fornecidas por outros agentes, como por exemplo o Gestor de Gares. Desta forma, a Torre de Controlo coordena todos os agentes envolvidos no sistema, no qual esta coordenação é efetuada através da comunicação bidirecional entre a Torre de Controlo e os agentes, sendo que recebe informações de todos os outros agentes, determina decisões e emite-as com base nessas informações.

No que concerne ao funcionamento da arquitetura, esta classifica-se como **híbrida**, na medida em que o sistema envolve uma combinação de processos deliberativos e reativos. Destes processos destacam-se, por exemplo, a verificação da disponibilidade de pistas e gares aquando de um pedido de aterragem, sendo este um processo reativo. Por outro lado, temos a disponibilização da informação das gares por parte do Gestor de Gares à Torre de Controlo, sendo este um exemplo de um processo deliberativo.

Na figura 2.1 é ilustrada a arquitetura na qual assenta o sistema multiagente desenvolvido, na qual é enaltecida apenas a troca de mensagens entre os agentes (e não a ordem pela qual estas são efetuadas, tendo em conta o processo que desencadeou a troca de mensagens). Deste modo, dados os agentes supramencionados, ao evidenciar as trocas de mensagens, conseguimos perceber o que foi referido acerca da tipologia da arquitetura, isto é, do facto de se tratar de uma arquitetura centraliza e híbrida.

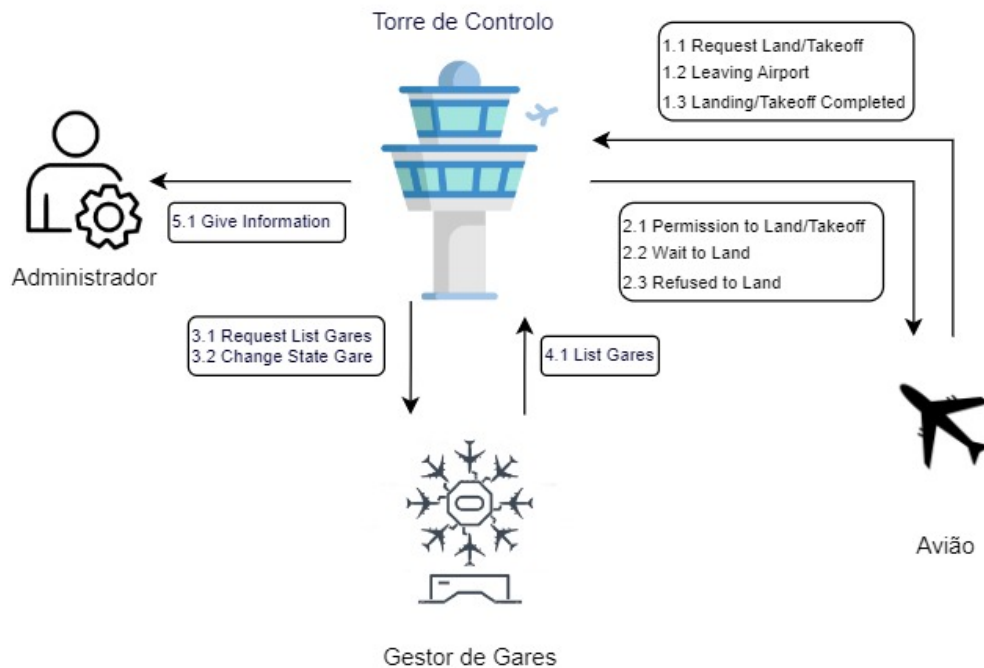


Figura 2.1: Arquitetura do Sistema

2.3 Definição e Descrição dos Agentes

No que concerne aos agentes existentes e envolvidos no sistema multiagente, tal como referido anteriormente, estes são representativos de entidades presentes num aeroporto real. Com isto, estes agentes têm como função simular alguns dos processos efetuados pelas entidades que representam.

2.3.1 Avião

O agente Avião é o agente que realiza os processos de aterragem e descolagem, no qual existem vários destes agentes no sistema. O agente interage com o agente Torre de Controlo para comunicar da sua pretensão (descolagem ou aterragem), no qual fornece alguns elementos identificadores e deve aguardar a aprovação da mesma para a realização do processo em causa. Em ambos os processos, o agente utiliza uma pista e uma gare (no qual o tipo de gare utilizada depende do tipo do avião), cuja informação foi-lhe atribuída pela Torre de Controlo. Este contém ainda um estado consoante a sua atividade no sistema multiagente, além de possuir tempos de aterragem/descolagem, estacionamento e espera. Estes tempos estão relacionados com os processos de: aterragem/descolagem, duração do avião no aeroporto e aguardar por aprovação da Torre de Controlo para poder realizar a aterragem (caso este tempo expire, o avião dirige-se a outro aeroporto), respetivamente. Com isto, foram definidos os seguintes parâmetros para o agente.

- **id**: Identificador único do avião;
- **type**: Tipo do avião, no qual cada avião pode ser do tipo comercial (avião de passageiros) ou do tipo de mercadorias;
- **destination**: Cidade de destino do avião;

- **origin:** Cidade de origem do avião;
- **company:** Companhia aerea do avião;
- **state:** Estado do avião. O avião apenas se encontra num dos seguintes estados: **OnAir**, **Parked**, **WaitingTakeOff**, **TakingOff**, **WaitLandPermission** e **Landing**;
- **gare:** Gare na qual se encontra estacionado o avião;
- **runaway:** Pista de aterragem/descolagem utilizada pelo avião aquando do processo correspondente;
- **landing time:** Tempo associado ao processo de aterragem;
- **parked time:** Tempo associado ao período de inativação do avião quando este está estacionado na gare;
- **max land wait time:** Tempo associado ao período de espera relacionado ao processo de aterragem, no qual o avião aguarda a aprovação e informação por parte da Torre de Controlo;

Posto isto, consoante os processos em causa, o avião irá ter behaviours com performatives associadas que descrevam as várias etapas envolvidas. Desta forma, cada avião terá a seguinte tabela de behaviours.

Behaviour	Performative	Descrição
ListenBehaviour		Escuta mensagens provenientes da Torre de Controlo. Consoante a performative recebida na mensagem, o avião executará um dos behaviours.
RequestLandBehaviour	Request for Landing	Pede autorização à Torre de Controlo para poder realizar o processo de aterragem, com a performative indicada.
RefusedLandBehaviour		Encaminha o avião a dirigir-se a outro aeroporto para poder aterrar.
WaitLandBehaviour	Leaving Airport	Aguarda o avião pela autorização de aterragem. Pode procurar outro aeroporto enviando uma mensagem à Torre de Controlo com a performative indicada.
LandBehaviour	Landing Completed	Gere o processo de aterragem e atualiza as informações do avião. Quando concluído envia uma mensagem à Torre de Controlo com a performative indicada.
RequestTakeOffBehaviour	Requet fo Takeoff	Pede autorização à Torre de Controlo para poder realizar o processo de decolagem, com a performative indicada.
TakeoffBehaviour	Takeoff Completed	Gere o processo de aterragem e atualiza as informações do avião. Quando concluído envia uma mensagem à Torre de Controlo com a performative indicada.

Tabela 2.1: Tabela de behaviours do agente Avião

Além disto, o agente avião, tal como descrito através do parâmetro **state**, irá descrever diversos estados ao longo do seu funcionamento no sistema. Assim, consoante as ações decorrentes, o agente irá transacionar entre estados, conforme os processos. Posto isto, podemos então definir a seguinte máquina de estados, associados aos diversos estados que um agente avião pode adotar no decorrer do sistema.

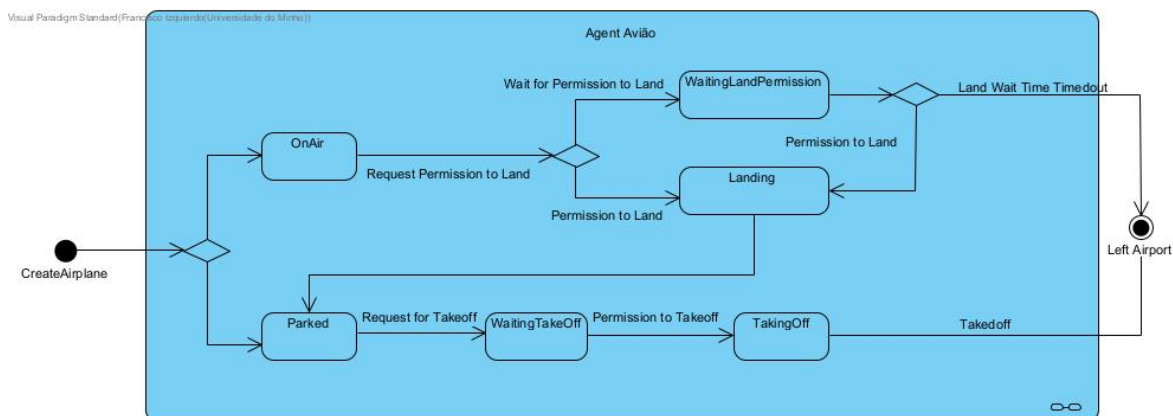


Figura 2.2: Máquina de Estados do agente Avião

2.3.2 Gestor de Gares

Agente representativo de Gestor de Gares, responsável por agregar e gerir a informação relativa às gares do aeroporto, local de estacionamento dos aviões. Este agente é único e comunica com o agente Torre de Controlo, disponibilizando-lhe o estado das gares a pedido deste, isto é, caso estejam ocupadas ou livres, aquando de um processo de aterragem/descolagem. Com isto, foram definidos os seguintes parâmetros.

- **gares:** Lista de gares presentes no aeroporto. Cada gare tem um tipo associado (comercial ou mercadorias) e um estado (ocupada ou livre). Além disso, contém um identificador único, bem como uma posição definida pelas coordenadas x e y ;

Posto isto, consoante os processos em causa, o Gestor de Gares irá ter behaviours com performatives associadas que descrevam as várias etapas envolvidas. Desta forma, o Gestor de Gares terá a seguinte tabela de behaviours.

Behaviour	Performative	Descrição
ListenBehaviour		Escuta mensagens provenientes da Torre de Controlo. Consoante a performative recebida na mensagem, o Gestor de Gares executará um dos behaviours.
ListGaresBehaviour	Gares List	Lista todas as gares de um dado tipo que estão livres.
ChangeStateBehaviour		Altera o estado de uma dada gare para ocupado ou livre.

Tabela 2.2: Tabela de behaviours do agente Gestor de Gares

2.3.3 Administrador

Agente responsável por apresentar a informação geral sobre os voos em operação no qual a agrega e a disponibiliza, apresentando toda a informação sobre os aviões que estão para descolar ou aterrar. Este agente é único e está periodicamente (de 1 em 1 segundo recebe a informação) em contacto com a Torre de Controlo.

Posto isto, consoante os processos em causa, o Administrador irá ter behaviours com performatives associadas que descrevam as várias etapas envolvidas. Desta forma, o Administrador terá a seguinte tabela de behaviours.

Behaviour	Performative	Descrição
UpdateBehaviour	Request Report	Pede a informação sobre o estado do aeroporto, nomeadamente sobre os aviões que estão para aterrar e descolar.

2.3.4 Torre de Controlo

Agente representativo de Torre de Controlo, que gere os pedidos de aterragem/descolagem por parte dos agentes aviões. Este agente é único e contém/gere as pistas a serem utilizadas pelos agentes supramencionados. Comunica com os agentes aviões para resolver os seus pedidos, com o agente Gestor de Gares relativamente à informação das gares e com o agente Administrador para disponibilizar a informação do estado do aeroporto. Aquando de um pedido de aterragem por parte de um avião, a Torre de Controlo primeiramente pede a informação das gares disponíveis consoante o tipo de avião que quer aterrar. Seguidamente, determina qual a melhor pista e gare (o par que tem a menor distância) e envia esta informação ao avião que pediu para aterrar. Caso haja indisponibilidade (quer seja por gares ou pistas indisponíveis), informa o avião que deve aguardar assim que a aterragem seja possível de efetuar, colocando-o numa *queue* de aviões que estão em espera para aterrar. Por outro lado, quando a Torre de Controlo recebe um pedido de descolagem por parte de um avião, determina qual a pista mais próxima e informa o avião qual a pista a utilizar. Caso haja indisponibilidade de pistas, coloca o avião numa *queue* de aviões que estão em espera para descolar, sendo posteriormente resolvidos. Além disto, periodicamente envia informação do estado do aeroporto ao administrador. Com isto, foram definidos os seguintes parâmetros.

- **runaways:** Lista de pistas presentes no aeroporto. Cada pista contém um estado associado (ocupado ou livre), bem como um identificador único e uma posição definida pelas coordenadas x e y ;
- **landing queue:** *Queue* associada aos aviões que aguardam confirmação para aterrar;
- **takeoff queue:** *Queue* associada aos aviões que aguardam confirmação para descolar;
- **max land queue size:** Tamanho inicial associado à *queue* de aterragem;

Posto isto, consoante os processos em causa, a Torre de Controlo irá ter behaviours com performatives associadas que descrevam as várias etapas envolvidas. Desta forma, a Torre de Controlo terá a seguinte tabela de behaviours.

Behaviour	Performative	Descrição
ListenBehaviour		Escuta mensagens provenientes de aviões e do Gestor de Gares. Consoante a performative recebida na mensagem, a Torre de Controlo executará um dos behaviours.
RequestLandBehaviour	Gares List Refused to Land Wait to Land Change Gare Permission to Land	Gere todas as etapas e opções associadas ao processo de aterragem, através das performatives respetivas e indicadas.
TakeoffRequestBehaviour	Permission to Takeoff	Gere todas as etapas e opções associadas ao processo de descolagem, através da performative indicada.
LandAbortBehaviour		Remove um avião da queue de descolagem ao receber a indicação por parte de um avião de que este irá procurar outro aeroporto para aterrar.
ConfirmBehaviour	Gares List Change Gare Permission to Land Permission to Takeoff	Gere e atualiza as queues de aterragem e descolagem, através das performatives indicadas.

Tabela 2.3: Tabela de behaviours do agente Torre de Controlo

2.4 Estrutura do Sistema

Uma vez detalhados e definidos os diversos agentes envolvidos no sistema desenvolvido, convém também salientar aspetos relativos à constituição do sistema, isto é, como o mesmo se organizou. No que concerne à estrutura do sistema, esta pode ser definida com recurso ao diagrama de *package* apresentado na figura 2.3.

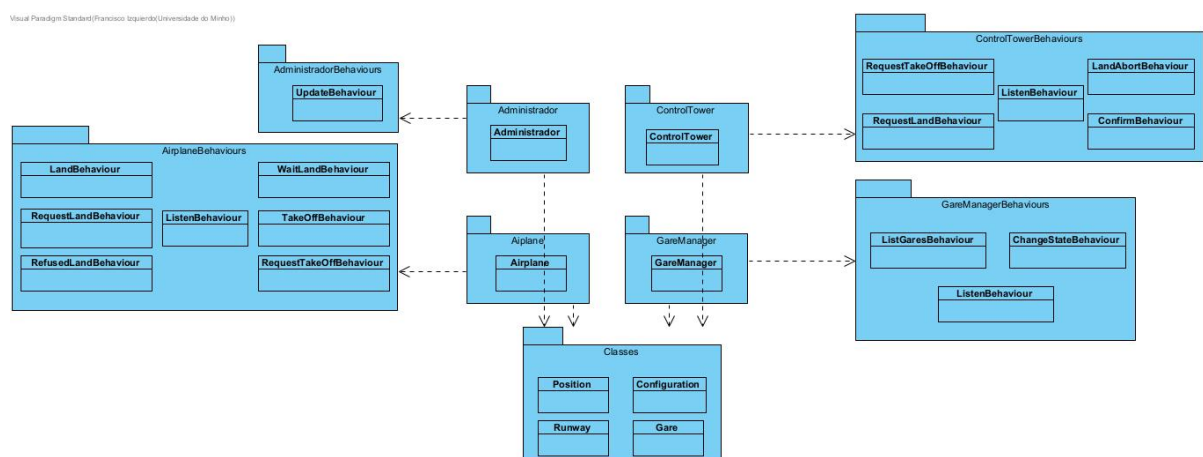


Figura 2.3: Diagrama de Packages

Uma vez definida a estrutura do sistema desenvolvido, podemos agora definir a organização do sistema. Assim, através do supramencionado em 2.3, no qual foram detalhados todos os aspetos relativos aos agentes, bem como os seus behaviours e correspondentes performatives,

temos o seguinte diagrama de classes.

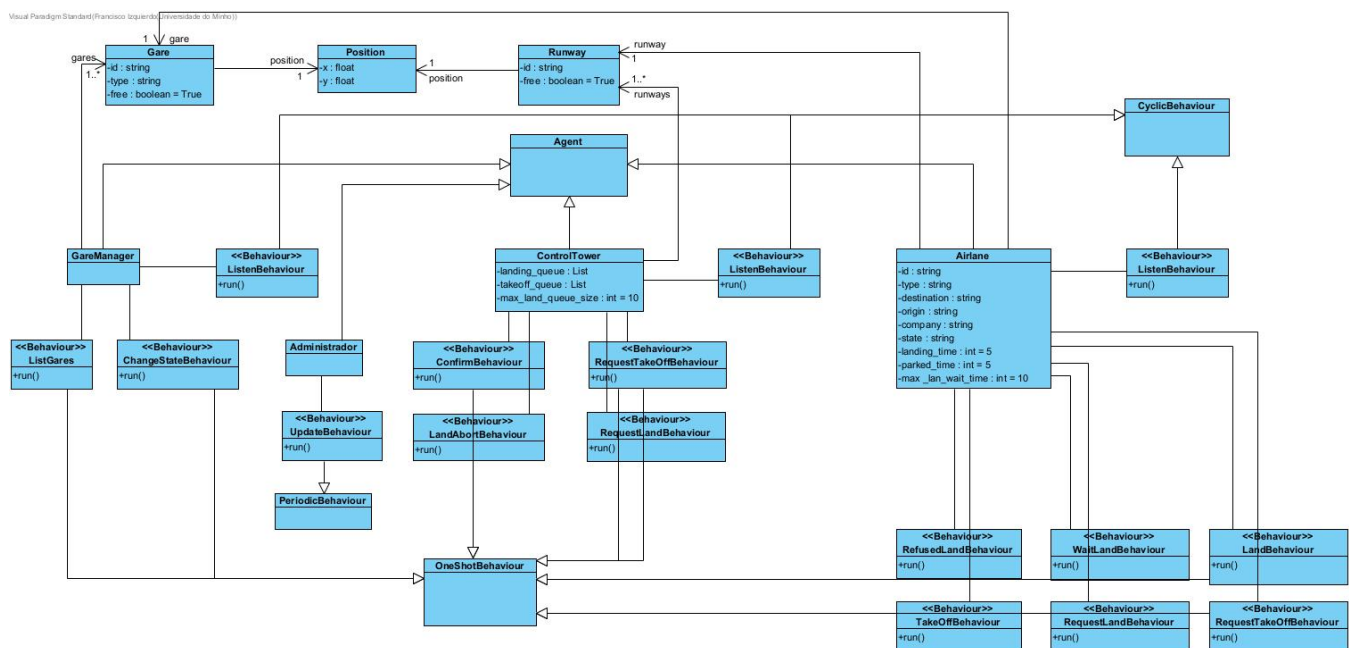


Figura 2.4: Diagrama de Classes

2.5 Interações dos Agentes

De forma a perceber todos os processos envolvidos no sistema multiagente desenvolvido pelo grupo de trabalho, foram elaborados diagramas de interações que elucidem as diferentes interações cooperativas entre os diversos agentes. Com isto, surgiram logicamente os diagramas de **colaboração**, **sequência** e **atividades**, no qual é salientado o papel do agente avião, uma vez que é este o impulsor dos principais processos (aterragem e descolagem) do funcionamento do aeroporto. Assim, em cada um dos diagramas, este agente será o cerne da definição dos mesmos, no qual as interações para com os restantes agentes cooperativos são elucidadas e motivadas pelo agente avião. Relativamente ao agente Administrador, uma vez que este apenas interage periodicamente sempre da mesma forma com o agente Torre de Controlo, por forma a simplificar os diagramas, as interações deste agente não serão enaltecidas.

Posto isto, podemos então explicar as diversas interações entre os agentes dado o supramencionado, ou seja, tendo como agente impulsor do fluxo de interações o agente avião que deseja realizar o processo de aterragem. Assim, tomando como ponto de partida um avião que queira aterrar, este envia a mensagem de pedido de aterragem para a Torre de Controlo. Posteriormente, a Torre de Controlo irá enviar a mensagem de pedido da lista de gares disponíveis dado o tipo do avião ao Gestor de Gares. Ao obter a lista de gares envia ao avião uma das duas mensagens: aguardar permissão para que possa aterrar ou permissão para aterrar. Seguidamente, o avião pode abandonar o aeroporto (caso o seu tempo de espera expire) ou prosseguir para o processo de aterragem assim que tenha permissão para o fazer, no qual o avião envia a mensagem a confirmar a conclusão do processo à Torre de Controlo e estaciona na gare que lhe foi atribuída. Ao fim de algum tempo estacionado, inicia o processo de descolagem enviando o pedido de descolagem à Torre de Controlo, ao qual esta lhe envia uma das duas respostas: aguardar permissão para descolar ou permissão para descolar. Por fim, quando é concedida

permissão ao avião para descolar, ao concluir este processo envia uma mensagem à Torre de Controlo. Além disto, a Torre de Controlo envia a mensagem para alterar o estado da gare desocupada ao Gestor de Gares.

2.5.1 Diagrama de Sequência

No que concerne ao diagrama de sequência, podemos então através das definições das interações e consequente fluxo das mesmas ao longo do sistema, definir a sequência tendo em conta os diversos *behaviours* dos agentes envolvidos. É de notar que o seguinte diagrama de sequência exemplifica a comunicação entre os agentes desde o momento da chegada até à partida de um avião não abordando o caso em que avião que já se encontre no aeroporto ou as comunicações realizadas pela torre para decidir quais os aviões em fila de espera que deve chamar.

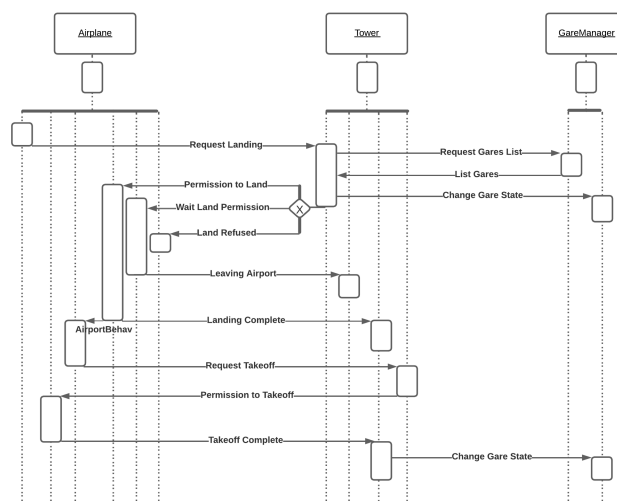


Figura 2.5: Diagrama de Sequência

2.5.2 Diagrama de Colaboração

O seguinte diagrama enaltece as trocas de mensagens entre os diversos agentes e o fluxo bem como o rumo que estas tomam em prol dos acontecimentos, tomando sempre como ponto de partida um avião que queira aterrar.

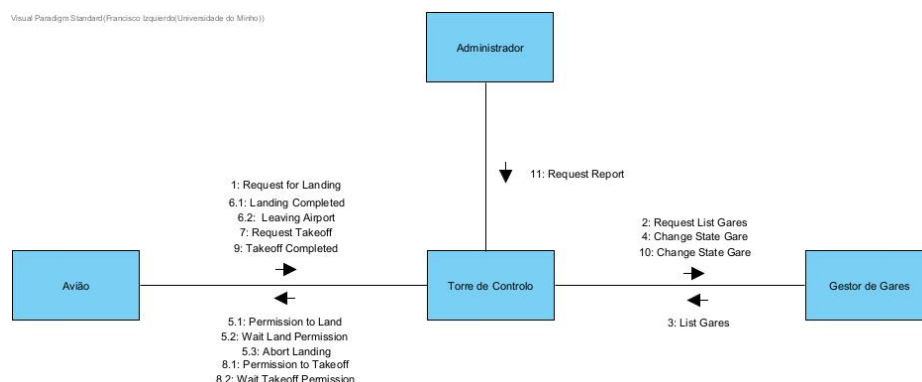


Figura 2.6: Diagrama de Colaboração

2.6 Requisito Adicional

No que toca ao requisito adicional, o objetivo deste passou por executar o sistema de forma mais realista às condições com que os aviões no dia a dia se deparam no mundo real. Como tal, desenvolvemos um sistema meteorológico simples capaz de simular 4 condições meteorológicas entre as quais: **sol**, **chuva**, **nevoeiro** e **tempestade**. Estas condições adicionam um pequeno atraso aos tempos de aterragem e descolagem dos aviões tornando assim os níveis de ocupação do aeroporto menos uniformes criando situações de desistência de aviões e elevado tráfego assim como situações mais calmas e sem filas de espera.

Assim, relativamente ao cenário de *default* e tendo em conta as mudanças climatológicas, resultou a seguinte tabela de exemplo relativo à uma mudança de clima para uma mesma execução.

Gares		
ID	Type	Free
0	Merchandise	True
1	Merchandise	True
2	Comercial	False
3	Comercial	False
4	Merchandise	True
5	Comercial	False
6	Comercial	False
7	Comercial	False
8	Merchandise	False
9	Merchandise	False

Weather state: Fog

Planes

ID	Origin	Destination	Company	Type	State	Gare	Runway
plane0@192.168.56.1	Rome	Lisbon	EasyJet	Comercial	parked	3	None
plane4@192.168.56.1	Lisbon	Barcelona	RyanAir	Merchandise	waiting_takeoff_permission	9	None
plane5@192.168.56.1	Lisbon	Rome	Norwegian Air Shuttle	Comercial	taking_off	None	2
plane6@192.168.56.1	Lisbon	Milan	RyanAir	Comercial	waiting_takeoff_permission	6	None
plane7@192.168.56.1	Lisbon	Paris	RyanAir	Comercial	waiting_takeoff_permission	7	None
plane8@192.168.56.1	Lisbon	Madrid	Icelandair	Merchandise	waiting_takeoff_permission	8	None
plane10@192.168.56.1	Rome	Lisbon	EasyJet	Comercial	landing	2	1
plane11@192.168.56.1	Amestardan	Lisbon	RyanAir	Comercial	landing	3	3
plane12@192.168.56.1	Manchester	Lisbon	Wizz Air	Comercial	landing	5	0

Aborts

ID	Origin	Destination	Company	Type	State
plane9@192.168.56.1	Rome	Lisbon	EasyJet	Comercial	aborted

Figura 2.7: Cenário com nevoeiro

Gares

ID	Type	Free
0	Commercial	False
1	Merchandise	True
2	Merchandise	True
3	Merchandise	False
4	Merchandise	False
5	Merchandise	False
6	Merchandise	False
7	Merchandise	True
8	Commercial	False
9	Merchandise	False

Weather state: Sun

Planes

ID	Origin	Destination	Company	Type	State	Gare	Runway
plane0@192.168.56.1	Barcelona	Lisbon	RyanAir	Commercial	parked	0	None
plane1@192.168.56.1	Oporto	Lisbon	Icelandair	Commercial	landing	8	0
plane2@192.168.56.1	Barcelona	Lisbon	Icelandair	Merchandise	landing	6	2
plane3@192.168.56.1	Barcelona	Lisbon	TAP Air Portugal	Merchandise	parked	3	None
plane4@192.168.56.1	Amestardan	Lisbon	Air Europa	Merchandise	parked	4	None
plane5@192.168.56.1	Barcelona	Lisbon	RyanAir	Merchandise	parked	5	None
plane6@192.168.56.1	Milan	Lisbon	Eurowings	Commercial	waiting_land_permission	None	None
plane7@192.168.56.1	Manchester	Lisbon	Icelandair	Commercial	waiting_land_permission	None	None
plane8@192.168.56.1	Oporto	Lisbon	Norwegian Air Shuttle	Commercial	waiting_land_permission	None	None
plane9@192.168.56.1	Milan	Lisbon	Vueling Airlines	Merchandise	parked	9	None

Figura 2.8: Cenário com sol

3. Resultados

Por forma a percebermos a capacidade do funcionamento e gestão feito pelos diversos agentes no sistema multiagente, foram realizados alguns cenários de teste. Nestes cenários, foram tidas em conta métricas que iriam afetar o desempenho do sistema, no qual foram tomadas variações de valores destes, por forma a forçar *bottlenecks*, no qual os testes foram avaliados após a estabilização do sistema (o sistema executou durante três minutos). Após a execução dos testes, foi feita uma análise crítica geral sobre as métricas alteradas, tal como descrito em 4.1. Posto isto, convém referir que as três primeiras métricas dizem respeito aos tempos associados nos processos por parte dos aviões, enquanto que a última métrica diz respeito à constituição do aeroporto. Deste modo, foram variadas as seguintes métricas:

- Intervalo de tempo entre o qual são criados aviões e estes chegam ao aeroporto (período de chegada de aviões);
- Tempo de espera para aterrar;
- Tempo de descolagem/aterragem;
- Tempo de estacionamento;
- Número de gares e/ou pistas;
- Tamanho das *queues* de aterragem;

3.1 Cenário 1 (Default)

De modo a poder comparar os diversos resultados obtidos através dos cenários de testes realizados, foi tido em conta um primeiro cenário tido como referência de funcionamento. Neste cenário, os valores das métricas mencionadas têm o valor *default* de funcionamento do aeroporto. Assim, temos os seguintes valores para as métricas utilizadas:

- Período de chegada de aviões = 10;
- Tempo de espera para aterrar = 30;
- Tempo de descolagem/aterragem = 5;
- Tempo de estacionamento = 10;
- Número de gares = 10;
- Número de pistas = 5;
- Tamanho da *queue* de aterragem = 10;

Posto isto, através das informações fornecidas pelo Administrador que está em contacto periódico com a Torre de Controlo, obteve-se a seguinte tabela de resultados.

Gares

ID	Type	Free
0	Comercial	True
1	Comercial	True
2	Merchandise	False
3	Merchandise	True
4	Merchandise	True
5	Merchandise	False
6	Comercial	True
7	Merchandise	True
8	Comercial	True
9	Comercial	True

Planes

ID	Origin	Destination	Company	Type	State	Gare	Runway
plane13@192.168.56.1	Lisbon	Manchester	Norwegian Air Shuttle	Merchandise	taking_off	5	2
plane14@192.168.56.1	Rome	Lisbon	Air Europa	Merchandise	parked	2	None

Figura 3.1: Cenário 1 (Default)

3.2 Cenário 2

Para o cenário 2, a métrica cujo valor foi variado, foi o período de chegada de aviões, no qual foram realizados dois testes, com o período de chegada de aviões igual a **2 e 15 segundos**, respetivamente. Com a alteração do valor desta métrica, pretendeu-se perceber como a frequência de novos aviões a chegarem ao aeroporto e a quererem aterrar poderia influenciar o funcionamento do sistema. Consequentemente, pretendeu-se perceber como a *queue* dos aviões que estavam em espera para aterrar era afetada e gerida pela Torre de Controlo, bem como perceber se a quantidade de aviões que desistiam de aterrar crescia drasticamente. Assim, relativamente ao valor de *default*, resultaram as seguintes tabelas.

Gares							
ID	Type	Free					
0	Commercial	False					
1	Merchandise	True					
2	Merchandise	True					
3	Merchandise	True					
4	Merchandise	True					
5	Commercial	True					
6	Merchandise	True					
7	Commercial	False					
8	Merchandise	True					
9	Commercial	False					
Planes							
ID	Origin	Destination	Company	Type	State	Gare	Runway
plane37@192.168.56.1	Lisbon	London	RyanAir	Merchandise	taking_off	None	0
plane38@192.168.56.1	Lisbon	Barcelona	Eurowings	Merchandise	taking_off	None	1
plane39@192.168.56.1	Lisbon	Amestradan	Air Europa	Merchandise	taking_off	None	2
plane40@192.168.56.1	Lisbon	Oporto	Wizz Air	Commercial	waiting_takeoff_permission	0	None
plane41@192.168.56.1	Lisbon	Amestradan	RyanAir	Merchandise	waiting_takeoff_permission	2	None
plane43@192.168.56.1	Lisbon	Rome	Eurowings	Commercial	waiting_takeoff_permission	0	None
plane44@192.168.56.1	Lisbon	Madrid	EasyJet	Commercial	waiting_takeoff_permission	7	None
plane45@192.168.56.1	Lisbon	Paris	Vueling Airlines	Commercial	waiting_takeoff_permission	9	None
plane46@192.168.56.1	Lisbon	Barcelona	TAP Air Portugal	Merchandise	waiting_takeoff_permission	2	None
plane47@192.168.56.1	Milan	Lisbon	RyanAir	Commercial	parked	7	None
plane48@192.168.56.1	Milan	Lisbon	Norwegian Air Shuttle	Merchandise	parked	8	None
plane49@192.168.56.1	Manchester	Lisbon	Vueling Airlines	Merchandise	parked	6	None
plane50@192.168.56.1	Milan	Lisbon	Eurowings	Commercial	parked	9	None
plane51@192.168.56.1	Amestradan	Lisbon	Wizz Air	Commercial	parked	7	None
plane52@192.168.56.1	Milan	Lisbon	Eurowings	Merchandise	landing	6	3
plane53@192.168.56.1	London	Lisbon	Icelandair	Commercial	landing	9	4
plane54@192.168.56.1	London	Lisbon	Vueling Airlines	Commercial	waiting_land_permission	None	None

Figura 3.2: Cenário 2 (Período de chegada de aviões = 2 segundos)

Gares		
ID	Type	Free
0	Merchandise	False
1	Comercial	True
2	Merchandise	True
3	Merchandise	False
4	Comercial	True
5	Merchandise	True
6	Comercial	True
7	Comercial	True
8	Merchandise	True
9	Merchandise	True

Planes							
ID	Origin	Destination	Company	Type	State	Gare	Runway
plane14@192.168.56.1	Lisbon	Madrid	TAP Air Portugal	Merchandise	taking_off	0	2
plane15@192.168.56.1	Milan	Lisbon	RyanAir	Merchandise	landing	3	3

Figura 3.3: Cenário 2 (Período de chegada de aviões = 15 segundos)

3.3 Cenário 3

Já no cenário 3, a métrica que teve o seu valor alterado foi o tempo de descolagem/partida. Este, tal como a métrica anterior, teve o seu valor alterado para um valor superior e inferior, tendo valores de **15 e 3 segundos**, respetivamente, a fim de forçar *bottlenecks* relativamente aos processos de aterragem/descolagem. Com isto, o objetivo era analisar em que medida a duração associada aos processos de aterragem/descolagem pode influenciar as *queues* geridas pela Torre de Controlo, isto é, se estas a dado ponto do funcionamento do sistema, iriam crescer/diminuir drasticamente. Além disso, pretendeu-se perceber com que frequência as pistas e gares eram ocupadas e/ou desocupadas, uma vez que isto era uma propriedade emergente da alteração do valor da métrica em estudo. Assim, relativamente ao valor de *default*, resultou a seguinte tabela.

Gares		
ID	Type	Free
0	Comercial	True
1	Merchandise	True
2	Comercial	False
3	Comercial	True
4	Merchandise	False
5	Comercial	True
6	Merchandise	False
7	Comercial	False
8	Comercial	False
9	Comercial	False

Planes								
ID	Origin	Destination	Company	Type	State	Gare	Runway	
plane3@192.168.56.1	Rome	Lisbon	RyanAir	Comercial	parked	9	None	
plane4@192.168.56.1	Lisbon	Paris	Vueling Airlines	Merchandise	taking_off	4	4	
plane5@192.168.56.1	Lisbon	Milan	Icelandair	Comercial	taking_off	None	1	
plane6@192.168.56.1	Lisbon	Milan	Air Europa	Merchandise	taking_off	None	3	
plane7@192.168.56.1	Lisbon	Madrid	RyanAir	Comercial	waiting_takeoff_permission	7	None	
plane8@192.168.56.1	Lisbon	Paris	TAP Air Portugal	Comercial	waiting_takeoff_permission	8	None	
plane9@192.168.56.1	Oporto	Lisbon	Vueling Airlines	Comercial	parked	2	None	
plane10@192.168.56.1	Milan	Lisbon	TAP Air Portugal	Merchandise	landing	6	2	
plane11@192.168.56.1	Barcelona	Lisbon	Icelandair	Merchandise	waiting_land_permission	None	None	

Abortos								
ID	Origin	Destination	Company	Type	State	Gare	Runway	
plane2@192.168.56.1	Paris	Lisbon	Norwegian Air Shuttle	Merchandise	waiting_land_permission	None	None	

Figura 3.4: Cenário 3 (Tempo de aterragem/descolagem = 15 segundos)

Gares		
ID	Type	Free
0	Comercial	True
1	Comercial	True
2	Comercial	True
3	Comercial	True
4	Comercial	True
5	Merchandise	False
6	Comercial	True
7	Comercial	False
8	Comercial	True
9	Merchandise	True

Planes								
ID	Origin	Destination	Company	Type	State	Gare	Runway	
plane18@192.168.56.1	Lisbon	Manchester	Norwegian Air Shuttle	Comercial	taking_off	7	3	
plane19@192.168.56.1	Manchester	Lisbon	Norwegian Air Shuttle	Merchandise	parked	5	None	

Figura 3.5: Cenário 3 (Tempo de aterragem/descolagem = 3 segundos)

3.4 Cenário 4

No que diz respeito ao cenário 4, a métrica que teve o seu valor reduzido e aumentado foi a métrica relativa ao número de gares, no qual se realizaram dois testes em que o aeroporto teve **5 e gares**. Uma vez que as gares têm tipos associados (comercial ou mercadorias) e dado que a chegada do tipo dos aviões é aleatória, bem como a criação do tipo das gares aquando do arranque do sistema, pretendeu-se perceber em que medida o sistema era capaz de ser gerido tendo em conta as características do aeroporto, nomeadamente as gares. Desta forma, o objetivo

passou por perceber se, em caso de circunstâncias esporádicas (como haver apenas 1 gare do tipo comercial e as restantes do tipo mercadorias, por exemplo), o sistema era sustentável, isto é, se o *bottleneck* causado tanto pela redução do número de gares como a aleatoriedade do tipo das gares seria capaz de ser ultrapassado e contronado, sem haver mudanças radicais no funcionamento do sistema multiagente. Assim, relativamente ao valor de *default*, resultou a seguinte tabela.

Gares		
ID	Type	Free
0	Merchandise	True
1	Comercial	True
2	Merchandise	True
3	Merchandise	True
4	Comercial	True
5	Merchandise	False
6	Comercial	True
7	Merchandise	True
8	Merchandise	False
9	Comercial	False
10	Merchandise	True
11	Comercial	True
12	Merchandise	True
13	Merchandise	True
14	Merchandise	True

Planes								
ID	Origin	Destination	Company	Type	State	Gare	Runway	
plane19@192.168.56.1	Lisbon	Amestradan	Wizz Air	Comercial	taking_off	None	0	
plane20@192.168.56.1	Lisbon	Madrid	Norwegian Air Shuttle	Merchandise	taking_off	None	3	
plane21@192.168.56.1	Lisbon	Amestradan	EasyJet	Merchandise	taking_off	None	4	
plane22@192.168.56.1	Lisbon	London	Vueling Airlines	Merchandise	waiting_takeoff_permission	5	None	
plane23@192.168.56.1	London	Lisbon	Norwegian Air Shuttle	Comercial	parked	9	None	
plane24@192.168.56.1	London	Lisbon	Norwegian Air Shuttle	Merchandise	landing	8	1	
plane25@192.168.56.1	Rome	Lisbon	Vueling Airlines	Merchandise	landing	3	2	
plane26@192.168.56.1	Madrid	Lisbon	Icelandair	Merchandise	waiting_land_permission	None	None	

Figura 3.6: Cenário 4 (Número de gares = 15)

Gares		
ID	Type	Free
0	Merchandise	False
1	Comercial	True
2	Comercial	True
3	Merchandise	True
4	Comercial	True

Planes								
ID	Origin	Destination	Company	Type	State	Gare	Runway	
plane5@192.168.56.1	Lisbon	Barcelona	Vueling Airlines	Merchandise	taking_off	3	2	
plane6@192.168.56.1	Lisbon	Manchester	TAP Air Portugal	Merchandise	taking_off	None	1	
plane7@192.168.56.1	Oporto	Lisbon	Eurodings	Merchandise	parked	0	None	
plane9@192.168.56.1	Manchester	Lisbon	TAP Air Portugal	Merchandise	landing	0	0	
plane10@192.168.56.1	London	Lisbon	EasyJet	Merchandise	waiting_land_permission	None	None	

Aborts					
ID	Origin	Destination	Company	Type	State
plane8@192.168.56.1	Oporto	Lisbon	Norwegian Air Shuttle	Merchandise	aborted

Figura 3.7: Cenário 4 (Número de gares = 5)

4. Análise Crítica

4.1 Análise Crítica sobre os Resultados

Podemos aqui salientar para o facto de que a variação de algumas métricas irão surtir o mesmo efeito no funcionamento do sistema. Destas podemos destacar a variação do tamanho da *queue* de aterragem com a variação do tempo de descolagem/aterragem, no qual surtem o mesmo efeito. Do mesmo modo, a variação do tempo de chegada de aviões e a variação do tempo de estadia no aeroporto aparentam ter um efeito inversamente proporcional na quantidade de tráfego do aeroporto. É também de notar o possível valor de realizar uma pesquisa em grelha sobre uma grande quantidade de valores para cada parâmetro permitindo perceber qual o real impacto de cada um deste em combinação com os outros revelando de forma menos ambígua a dinâmica do sistema criado.

4.2 Análise Crítica sobre o Sistema Multiagente

O sistema multiagente desenvolvido constitui um ambiente no qual estão envolvidos as entidades propostas a serem implementadas sobre a forma de agentes. Estes cooperam entre si, através da troca de mensagens, mantendo o estado do sistema coerente. Além disso, há aspetos importantes a salientar, tais como os processos de gestão das *queues* por parte da Torre de Controlo, que se demonstra fulcral em circunstâncias adversas. Estas dizem respeito a situações como a indisponibilidade de um par gare-pista no processo de aterragem ou como a indisponibilidade de uma pista no processo de descolagem. Nestas circunstâncias, devido à sua implementação e mecanismo de prioridade, é dada a prioridade a aviões que queiram aterrar e se encontrem na *queue* de espera de aterragem (isto deve-se ao facto de haver associado um tempo limite tolerável de espera para o avião) ao invés de aviões que queiram descolar. Isto permite que haja menos aviões a não utilizarem o aeroporto, cujo um dos objetivos principais é ter uma maior afluência de aviões no aeroporto tanto maior quanto possível. Por fim, tal como mencionado em 4.1, o sistema mostrou ser escalável para um grande número de aviões, simulando situações muito próximas a eventos reais.

No que concerne a aspetos pertencentes à arquitetura sugerida pela equipa docente, esta no seu cerne demonstra-se simples e bastante intuitiva de implementar. Os agentes e as ações descritas por estes são relativamente fáceis de implementar e resolver, sendo que podiam ter sido mais extensivamente detalhadas. Além disso, devido à sua simplicidade, faltam alguns elementos importantes e que iriam redefinir e estender a arquitetura proposta, no qual podiam ter sido abordados outros fatores presentes nos aeroportos, tais como:

- **Gestor de reboques:** Os reboques seriam utilizados para deslocar os aviões das gares para as pistas;
- **Horário das viagens:** As viagens ao invés de terem tempos fixos, teriam um horário de partidas e chegadas. Com isto, o mecanismo de prioridades seria altamente estendido

e mais difícil de implementar, no qual a prioridade de aterragem teria de ter em conta o horário associado ao voo do avião;

- **Aviões privados ou de serviço:** Estes aviões teriam uma prioridade superior, independentemente do horário da sua chegada;
- **Aterragens de emergência:** Em situações esporádicas e aleatórias, surgiriam aterragens de emergência, que seriam o topo da prioridade;

5. Trabalho Futuro e Conclusão

5.1 Trabalho Futuro

No que concerne a trabalho futuro, poderia ser explorado o desenvolvimento de uma interface gráfica que permita a visualização em tempo real do estado do sistema, facilitando a monitorização das atividades dos diferentes agentes. Também seria interessante melhorar o mecanismo de prioridade para evitar conflitos e melhorar a eficiência do fluxo dos aviões pelos aeroportos. Além disso, é possível agregar funcionalidades como gestão de atrasos e cancelamentos de voos e coordenação com outros aeroportos e companhias aéreas. Para melhorar a eficácia e o realismo do sistema multiagente, outras características a serem consideradas em trabalhos futuros serão a implementação de modelos de inteligência artificial para otimizar os processos de gestão de pistas e gares.

5.2 Conclusão

Em suma, a realização deste trabalho prático permitiu consolidar e tirar proveito de diversos conceitos abordados na unidade curricular de Agentes e Sistemas Multiagente. Destes conceitos podemos destacar a biblioteca **SPADE** que foi uma ajuda fulcral na implementação dos agentes atuadores do sistema. Com isto, outros conceitos importantes que foram abordados e consolidados foram as noções de behaviours e correspondentes performatives associados à realização de um determinado agente. Outro conceito, foi a implementação de agentes cooperativos num sistema multiagente fechado tal como foi descrito em 2.1. Posta a implementação, esta permitiu tomar uma postura crítica acerca de todo o sistema, inferindo acerca deste e dos consequentes resultados obtidos através de diversos cenários de teste.

Podemos também enaltecer o facto do sistema multiagente realizado pelo grupo adequar-se ao proposto pela equipa docente, na medida que vai de encontro a este em todas as medidas. O sistema permite simular os processos envolvidos num aeroporto real, nomeadamente relativos aos processos de aterragem e/ou descolagem por parte de aviões, enaltecendo estes bem como condições adversas na gestão do aeroporto e que foram mencionadas ao longo do relatório. Contudo, o sistema multiagente não se encontra fixo à implementação desenvolvida, no qual se podem apontar aspetos para trabalho futuro, tal como evidenciado em 5.1.

Consideramos, concluindo, que o balanço do trabalho realizado é positivo e que os aspetos a melhorar podem ser facilmente atingidos em um trabalho futuro.