

Universidade do Minho
Escola de Engenharia

Sistemas Inteligentes
2024-2025

Relatório Prático

GRUPO 4

Beatriz Amorim, PG56112, DELTA= 0,5
Carolina Santos, PG56116, DELTA= 0,5
Catarina Nunes, PG56117 DELTA= 0,5
Sofia Correia, PG56152, DELTA= -1,5

Mestrado em Engenharia Biomédica - Informática Médica

03 de Janeiro de 2025

Conteúdo

| | | |
|----------|---|-----------|
| 1 | Introdução | 4 |
| 2 | Contextualização | 5 |
| 3 | Objetivos | 6 |
| 4 | Arquitetura do Sistema Multiagente | 6 |
| 5 | Descrição do Sistema Multiagente | 8 |
| 5.1 | Classes | 8 |
| 5.2 | Agentes | 9 |
| 5.3 | Behaviours | 10 |
| 5.3.1 | HospitalAvailable | 10 |
| 5.3.2 | TransportSubscribe | 11 |
| 5.3.3 | Donation | 11 |
| 5.3.4 | RequestOrgan | 11 |
| 5.3.5 | TransportOrgan | 11 |
| 5.3.6 | ReceiveInfo | 11 |
| 5.3.7 | ResendOrgan | 13 |
| 5.3.8 | MonitoringWaitlist | 14 |
| 5.3.9 | PatientReceives | 14 |
| 6 | Conclusão | 15 |

Lista de Figuras

| | | |
|---|---|---|
| 1 | Diagrama de sequência | 7 |
| 2 | Diagrama de colaboração/comunicação | 7 |
| 3 | Diagrama de atividades | 8 |
| 4 | Diagrama de classes | 9 |

1 Introdução

O presente relatório pretende caracterizar, em detalhe, o projeto elaborado no âmbito da Unidade Curricular de Sistemas Inteligentes, orientado para a conceção e implementação de um sistema multiagente direcionado à gestão de processos relacionados com transplantes de órgãos.

Este sistema, construído com recurso à plataforma SPADE, integra vários agentes inteligentes, capazes de captar e compreender informações sobre o mundo virtual em que operam. Além disso, os agentes comunicam entre si, tomado, assim, decisões coordenadas com vista ao sucesso e eficiência das operações associadas ao transplante.

O sistema construído foi concebido de modo a possibilitar a gestão eficiente de transplantes de órgãos, criando uma ponte entre doadores, recetores e unidades hospitalares. Para tal, é necessário considerar diversos fatores como a compatibilidade entre o órgão doado e o recetor, a localização de doadores, recetores e hospitais, a disponibilidade de transporte para os órgãos e a urgência clínica associada a cada caso.

No decorrer do relatório, serão detalhados e explicados os conceitos teóricos que sustentam a implementação deste projeto, destacando-se a forma como os agentes se organizam, comunicam e monitorizam diversos fatores, permitindo a tomada de decisões fundamentadas e a aplicação de planos de contingência quando necessário. Por fim, serão discutidas as limitações do sistema e enumeradas sugestões de melhoria que poderão ser consideradas em versões futuras.

2 Contextualização

O transplante de órgãos é um processo crucial na área da saúde. Esta operação é responsável por prolongar ou melhorar a qualidade de vida de imensas pessoas todos os anos. De acordo com o Instituto Português do Sangue e da Transplantação, em 2023, foram colhidos em Portugal 1066 órgãos, dos quais 963 foram transplantados, o que representa um aumento notável relativamente ao ano anterior [1].

No entanto, por trás de cada um dos transplantes realizados, existe uma gestão complexa de tempo, pessoas e recursos que precisa de ser executada com precisão, eficiência e eficácia, a fim de que a operação tenha sucesso. Diversos fatores de extrema importância devem ser considerados ao longo deste processo, tais como:

- O tempo de vida do órgão: Depois do órgão ser colhido, mesmo em condições ideais de preservação, só fica apto para ser transplantado durante um período limitado, o que exige rapidez nas operações subsequentes.
- A compatibilidade doador/recetor: Para executar um transplante de órgãos é necessário ter em conta vários aspectos biológicos e biomédicos intrínsecos ao órgão e/ou ao doador e recetor para concluir se estes são compatíveis.
- A prioridade do recetor: Em muitos casos, existem vários pacientes à espera do mesmo órgão, o que torna essencial a escolha do paciente mais adequado com base na urgência e necessidade clínica.
- A proximidade doador/recetor: A proximidade geográfica entre o doador e o recetor é um fator determinante, pois o tempo de transporte pode comprometer a viabilidade do órgão, mesmo que o paciente seja o mais adequado para o transplante.
- Disponibilidade de meios e recursos: A logística do transporte do órgão e a disponibilidade de espaços nos hospitais, como salas cirúrgicas, são fatores que podem afetar o sucesso do transplante. A falta de recursos ou a indisponibilidade de meios pode comprometer o procedimento.

Dada a alta procura, os órgãos são bens de enorme valor, pelo que a sua correta gestão é crucial. Por esta razão, torna-se essencial o desenvolvimento de sistemas que coordenem de forma eficaz os diversos agentes envolvidos no processo de transplante de órgãos. Tais sistemas devem ser capazes de aplicar medidas de contingência, redistribuindo um órgão de forma dinâmica e rápida sempre que surja qualquer adversidade que possa comprometer a viabilidade do transplante.

Portanto, o desafio consiste em criar soluções de gestão que consigam alojar os órgãos aos pacientes certos, no menor tempo possível, com todos os meios necessários, sempre com o objetivo central de salvar vidas.

3 Objetivos

O sistema multiagente desenvolvido visa otimizar a gestão de transplante de órgãos, coordenando a alocação de órgãos entre os dadores e os recetores, tendo em consideração a compatibilidade biológica, a urgência clínica e a logística do transporte de órgãos. Para isso, os agentes inteligentes são responsáveis por receber notificações sobre a disponibilidade de órgãos, analisar os dados de compatibilidade e urgência dos pacientes, e tomar decisões de forma coordenada e eficiente. A logística de transporte é otimizada para garantir que os órgãos sejam entregues ao hospital recetor dentro do tempo necessário para preservar a sua viabilidade. Em caso de falhas logísticas ou incompatibilidades, o sistema tem de implementar medidas de contingência, como a redistribuição de órgãos para outros pacientes ou hospitais adequados. O objetivo principal é "salvar" o maior número possível de vidas, assegurando a alocação rápida, precisa e segura dos órgãos.

4 Arquitetura do Sistema Multiagente

O sistema é composto por cinco tipos principais de agentes: DonorAgent, RecipientAgent, HospitalAgent, TransplantAgent e TransportAgent. Cada tipo de agente tem responsabilidades específicas e colaboram entre si para executar as operações de forma coordenada. A comunicação entre os agentes utiliza o protocolo de mensagens FIPA-ACL, suportado pela plataforma SPADE, garantindo uma troca de dados padronizada e eficiente.

Para ilustrar o funcionamento do sistema, foram elaborados diferentes tipos de diagramas que destacam aspectos distintos das interações e funcionalidades. O diagrama de sequência (Figura 1) apresenta o fluxo temporal das interações entre os agentes, detalhando as mensagens trocadas e os eventos disparados. O diagrama de colaboração (Figura 2) complementa a visão de sequência, enfatizando a estrutura das interações e as relações entre os agentes durante a execução. O diagrama de atividades (Figura 3) demonstra o fluxo de trabalho geral, destacando as etapas principais, as decisões críticas e os caminhos alternativos em cenários de contingência.

Através dos diagramas, observamos que o TransplantAgent desempenha um papel central, coordenando as decisões com base nas mensagens e respostas dos agentes envolvidos.

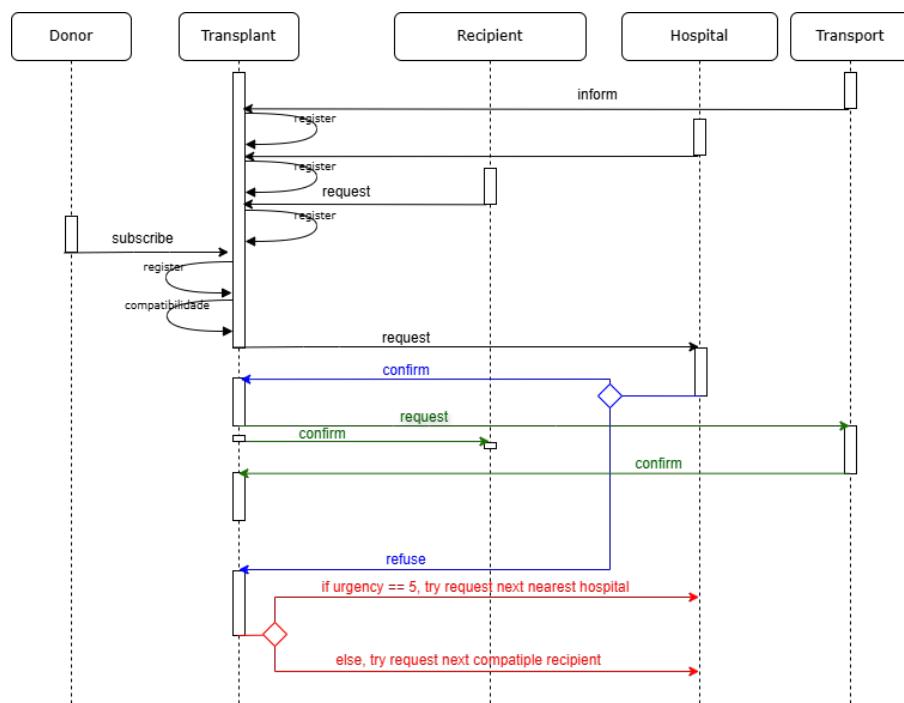


Figura 1: Diagrama de sequência

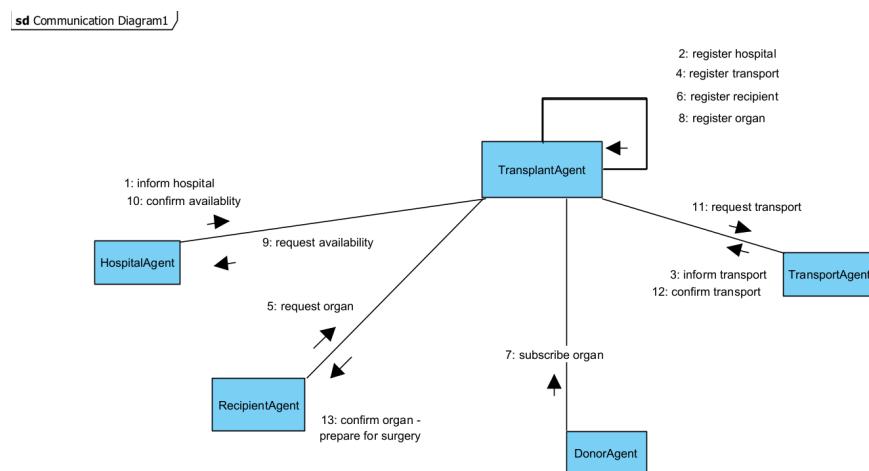


Figura 2: Diagrama de colaboração/comunicação

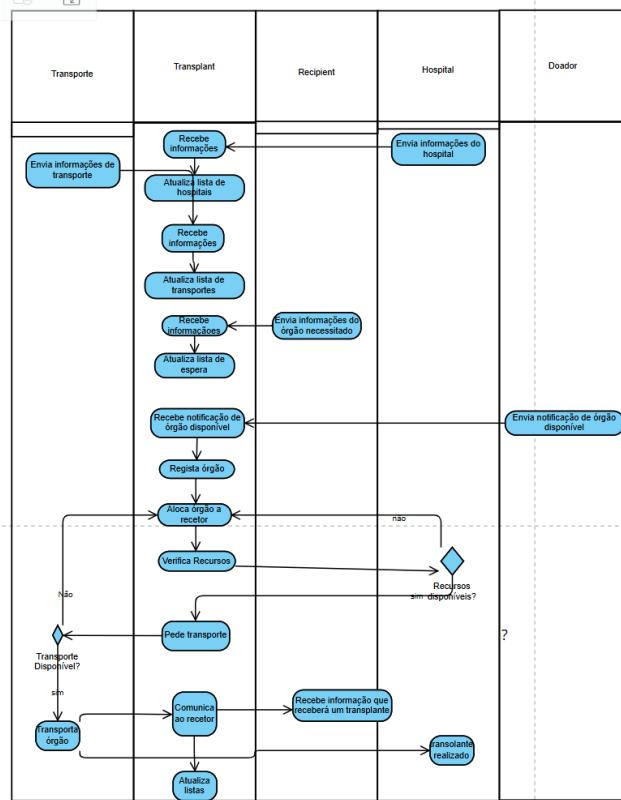


Figura 3: Diagrama de atividades

5 Descrição do Sistema Multiagente

O Sistema Multiagente desenvolvido é composto por diversas partes interligadas, onde agentes distintos colaboram e interagem para coordenar a doação, transporte e receção de órgãos.

De seguida, são descritas as principais classes, os agentes e os *behaviours* definidos no projeto.

5.1 Classes

Para organizar e estruturar as informações no sistema, foram definidas várias classes que representam os dados principais dos processos. Cada classe encapsula atributos e métodos relacionados a um determinado aspecto do sistema, como coordenadas geográficas, dados de hospitais, órgãos, receptores e transportes. As classes permitem a manipulação e a gestão eficaz das informações necessárias para o funcionamento adequado do sistema, além de facilitar a comunicação entre os agentes responsáveis pelas várias etapas do processo.

As classes definidas são as seguintes:

- **Coords:** Esta classe representa as coordenadas geográficas, com os atributos *x* e *y*. Inclui métodos para calcular a distância entre os dois pontos, verificar a igualdade de coordenadas e representar o objeto como uma string. A verificação da igualdade é particularmente relevante nos casos em que o dador e o recetor se encontram no mesmo hospital.

- **Hospital:** Modela uma unidade hospitalar com um identificador JID e a sua localização representada por um objeto Coords. Essa classe contém métodos para aceder e modificar a localização do hospital, além de obter o identificador único do hospital.
- **OrganData:** Representa os dados associados a um órgão disponível para transplante, incluindo o tipo de órgão (*organ_type*), o tipo de sangue (*blood_type*), o hospital de origem (*hospital*) e o agente doador (*agent_jid*). É essencial para armazenar e gerir as informações de órgãos disponíveis para transplante.
- **RecipientData:** Modela as informações de um recetor de órgão, incluindo o tipo de órgão necessário, o tipo sanguíneo, o nível de urgência, o hospital de localização e o agente recetor (*agent_jid*). É semelhante à classe *OrganData*, mas como uma adição do nível de urgência que é fundamental para a priorização dos transplantes urgentes.
- **TransportRequest:** Define os detalhes de um pedido de transporte, especificando o ponto de partida (*start*), o destino (*finish*) e o objeto a ser transportado. É essencial para organizar e gerir os pedidos de transporte dentro do sistema.
- **Transporte:** Representa o recurso responsável pelo transporte de órgãos, incluindo atributos como: o identificador (*jid*), a localização (*location*) e a disponibilidade do transporte (*available*)
- **TransplantInfo:** Permite transmitir as informações do dador e do recetor envolvidos em determinada associação de transplante, com recurso às classes *OrganData* e *RecipientData*.

O diagrama de classes referente a este sistema pode ser encontrado na Figura 4.

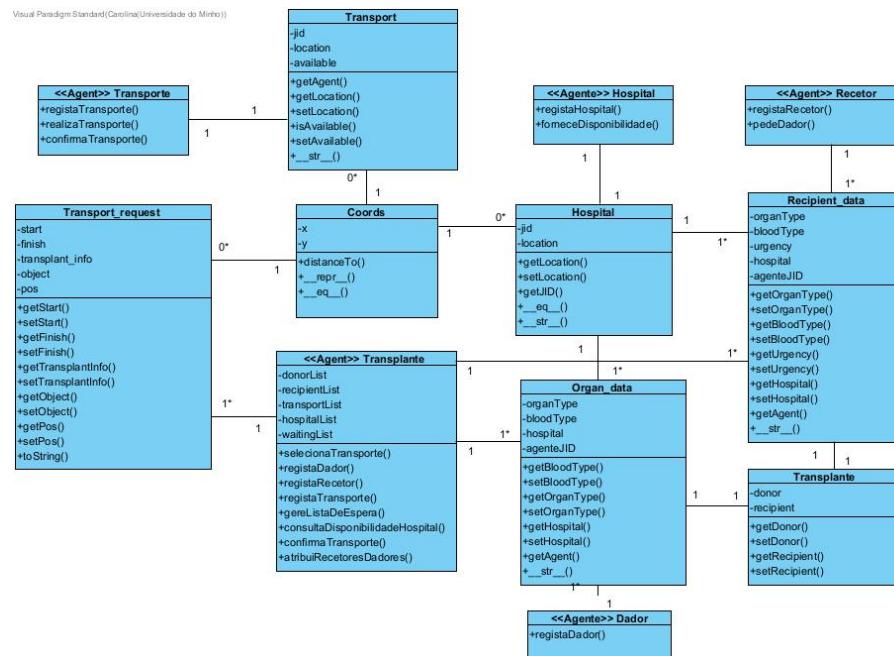


Figura 4: Diagrama de classes

5.2 Agentes

Os agentes são os principais responsáveis pela execução das operações dentro do sistema. Cada tipo de agente possui comportamentos específicos que são executados conforme a necessidade do

processo. No contexto deste sistema, temos agentes que representam os dadores, recetores, hospitais e transportes, e cada um deles interage com os outros de maneira coordenada para garantir o sucesso do processo de transplante de órgãos.

Foram definidos os seguintes agentes:

- **DonorAgent**: Este agente representa um dador de órgão, é responsável por iniciar a notificação da doação, com o *behaviour Donation*, que regista os dados do doador e informa o sistema sobre a disponibilidade de um novo órgão para transplante.
- **HospitalAgent**: O agente hospital gere a disponibilidade do hospital para receber pacientes com o *behaviour HospitalAvailable*.
- **RecipientAgent**: Este agente representa o recetor de órgãos, sendo responsável por gerir a solicitação de um órgão para transplante através do *behaviour RequestOrgan*, que administra a entrada na lista de espera para um órgão.
- **TransplantAgent**: O agente transplante, gere os órgãos disponíveis e o recetores, mantendo listas dos dadores e recetores. Este agente coordena a alocação de órgãos através de *behaviours* diferentes, o *ReceiveInfo* lida com a alocação imediata dos órgãos aos recetores compatíveis, o *ResendOrgan* gera a contingência de órgãos que não são compatíveis com os recetores disponíveis, redistribuindo-os conforme necessário. Assim, atua como um intermediário crucial para garantir a eficiência do processo de transplante, priorizando a compatibilidade e a urgência dos casos.
- **TransportAgent**: O agente de transporte é responsável por gerir o transporte de órgãos e, em situações pontuais de emergência, o transporte de pacientes. Tem dois *behaviours*: o *TransportSubscribe* que faz o registo dos transportes, e o *TransportOrgan* que realiza o transporte propriamente dito.

5.3 Behaviours

Os *behaviours* são classes que definem o comportamento dos agentes através de um conjunto de ações. A biblioteca SPADE oferece diversos tipos de *behaviours*, permitindo que os agentes executem diferentes tarefas de acordo com a sua funcionalidade. Entre os principais tipos de *behaviours*, é de referir o *OneShotBehaviour* e o *TimeOutBehaviour*, que são utilizados para tarefas executadas apenas uma vez, e o *PeriodicBehaviour* e o *CyclicBehaviour*, usados para tarefas repetitivas.

5.3.1 HospitalAvailable

O *behaviour HospitalAvailable* é responsável por verificar a disponibilidade dos hospitais para realizar transplantes de órgãos. Este *behaviour* é do tipo *CyclicBehaviour* e permanece à escuta de mensagens de outros agentes, como o *TransplantAgent* e o *RecipientAgent*. Quando recebe uma solicitação com a performative *request* de um desses agentes, o *HospitalAvailable* verifica a disponibilidade de recursos para realizar o transplante. Essa verificação é feita de forma aleatória. Caso o

hospital tenha recursos disponíveis, responde com a performative *confirm*, indicando que pode proceder com o transplante. Caso contrário, responde com a performative *refuse*, informando que não pode aceitar o órgão.

5.3.2 TransportSubscribe

Semelhante ao *behaviour* anterior, só que referente aos transportes, este *TransportSubscribe* permite ao *TransportAgent* informar o *TransplantAgent* sobre a sua localização e disponibilidade para realizar o transporte de órgãos. O agente transporte cria uma instância da classe Transporte, com uma localização aleatória e a disponibilidade e envia uma mensagem ao agente de transplantes com a performative *inform* com esses dados. Este comportamento garante que o agente de transplantes saiba sobre a disponibilidade de transporte para os órgãos serem movidos entre os doadores e os recetores.

5.3.3 Donation

Este *behaviour* é do tipo *OneShotBehaviour*, pelo que é executado uma única vez, quando o agente doador deseja notificar o agente de transplantes sobre a disponibilidade de um órgão para doação. O agente escolhe aleatoriamente o tipo de órgão e o tipo de sangue, além de selecionar um hospital para realizar o transplante. Assim, envia uma mensagem com a performative *Subscribe* ao *TransplantAgent*, notificando-o da disponibilidade e das características do órgão para doação.

5.3.4 RequestOrgan

Este *behaviour* é executado uma única vez, sendo do tipo *OneShotBehaviour*, e tem como objetivo fazer com o *recipientAgent* solicite um órgão para transplante. Ele escolhe aleatoriamente o tipo de órgão, o tipo de sangue, o hospital recetor e envia uma mensagem ao *TransplantAgent* com essas informações. A mensagem contém a performative *Request*, indicando que o *RecipientAgent* precisa de um órgão com as características específicas passadas na mensagem. A solicitação é feita ao sistema de transplantes, para encontrar uma correspondência para o pedido de órgão.

5.3.5 TransportOrgan

É responsável por gerir o transporte de órgãos e é executado de forma cíclica, sendo do tipo *CyclicBehaviour*. O *TransportAgent* recebe uma mensagem com a performative *Request* do *TransplantAgent*, que indica que um órgão precisa ser transportado do doado para o recetor. Após receber a solicitação, este agente, começa o processo de transporte, simulando-o com um atraso (`time.sleep(5)`) para representar o tempo de transporte. Quando o transporte é concluído, o agente de transporte envia uma mensagem de volta ao *TransplantAgent* com a performative *confirm*, indicando que o transporte foi bem-sucedido.

5.3.6 ReceiveInfo

Este *behaviour* é implementado pelo *TransplantAgent* e gera a receção e o processamento das mensagens. A principal responsabilidade deste comportamento é a gestão da informação sobre dado-

res, recetores, hospitais e transportes, além de realizar a alocação de órgãos entre dadores e recetores compatíveis, e solicitar a disponibilidade de transporte. Este *behaviour* recebe mensagens com diferentes *performatives* e executa ações específicas dependendo do tipo de mensagem. Assim, se a mensagem recebida tiver a *performative*:

- **Request:** Significa que a mensagem contém informações de um *RecipientAgent* (agente receptor). O *TransplantAgent* descodifica a mensagem, extrai os dados relevantes como a localização e os dados específicos do *recipient* e adiciona-o à sua lista de recetores - *recipientList*. Esta lista é fundamental para a gestão de potenciais recetores no sistema, permitindo que o agente identifique os candidatos quando surge um órgão disponível.
- **Inform:** Indica que há um novo *TransportAgent* disponível no sistema, pelo que extraí os dados do transporte como a localização e a disponibilidade do mesmo e adiciona-o à lista de transportes - *transportList*. Esta lista será utilizada posteriormente para gerir a logística do transporte de órgãos entre hospitais.
- **Subscribe:** É utilizada quando um *DonorAgent* ou outro *TransplantAgent* notifica sobre um órgão disponível. Se for de um *DonorAgent* regista os dados do órgão e adiciona o doador à sua lista de doadores - *donorList*. Se for de outro *TransplantAgent*, caso em que ocorre com a medida de contingência do *behaviour ResendOrgan*, verifica se o doador está registado, caso já esteja, ignora a mensagem para evitar duplicações. Após este registo e verificação, o *TransplantAgent* avalia todos os recetores compatíveis na *recipientList*. Estes são ordenados com base na urgência clínica e na proximidade ao hospital onde o órgão está localizado. O *TransplantAgent* então envia uma mensagem ao hospital do recetor com maior prioridade, solicitando a confirmação da sua disponibilidade para realizar o transplante.
- **Confirm com "hospital" incluído na mensagem:** Indica que o hospital está disponível para realizar o transplante. Quando isto acontece, o *TransplantAgent*:
 1. Regista a associação entre o órgão, o recetor e o hospital onde o transplante ocorrerá.
 2. Caso o doador e o recetor estejam em hospitais diferentes, seleciona um *TransportAgent* da *transportList* disponível para efetuar a logística do transporte do órgão. A escolha do agente de transporte é feita com base no cálculo da distância, sendo selecionado o transporte que estiver mais próximo do local onde o órgão do doador se encontra, garantindo maior eficiência no processo. Assim, o *TransplantAgent* envia uma mensagem de solicitação para o *TransportAgent* escolhido, detalhando as localizações do doador e do recetor, bem como outros dados relevantes. A mensagem contém as informações necessárias para o transporte do órgão, garantindo que o transporte seja feito de maneira eficiente e em tempo útil.
 3. Após o envio da mensagem ao *TransportAgent*, o *TransplantAgent* também envia uma mensagem de confirmação ao *RecipientAgent*, informando que o órgão está a caminho e solicitando que o paciente se prepare para a cirurgia. Esta confirmação garante que o recetor esteja ciente da chegada iminente do órgão e do transplante que será realizado. Se não houver transportes disponíveis no momento, o *TransplantAgent* coloca a solicitação na lista de espera e informa o *RecipientAgent* sobre o atraso, solicitando que o paciente se prepare para eventuais atrasos no processo de transporte.

- **Refuse:** Indica que o hospital não está disponível para realizar o transplante. O *TransplantAgent* reage de formas diferentes, dependendo do nível de urgência do recetor:

- **Urgência entre os níveis 1 a 4:** o órgão é atribuído ao próximo recetor compatível na lista, seguindo a ordem de prioridade estabelecida previamente.
- **Urgência máxima - nível 5:** Neste caso, dado que o recetor tem uma urgência muito elevada em receber o transplante, foi desenvolvida uma medida de contingência que permite redirecionar tanto o paciente quanto o órgão para outro hospital, caso o hospital inicial não tenha disponibilidade.

Assim, o *TransplantAgent* verifica outros hospitais representados por *HospitalAgents* na sua lista *hospitalList*, para determinar outros hospitais alternativos. Esta lista de hospitais alternativos é ordenada conforme a distância do hospital ao recetor. É então enviado o pedido de disponibilidade, através de uma mensagem do tipo *request*, a cada um desses hospitais na ordem definida. Se um hospital responder com a *performative inform*, significa que possui recursos para atender o recetor. É então selecionado esse hospital para proceder ao transplante, e o *TransplantAgent* informa o *RecipientAgent* sobre o hospital escolhido.

Posteriormente, o *TransplantAgent* tem de identificar um agente de transporte disponível na sua lista *transportList* para efetuar o transporte. A escolha é feita, mais uma vez, com base na proximidade do agente de transporte ao local do dador, priorizando o transporte mais próximo para minimizar o tempo de deslocação. Além disto, como o transporte do recetor também é necessário, o *TransplantAgent* procura um outro *TransportAgent* disponível para garantir o deslocamento do paciente até o hospital onde será realizado o transplante.

Caso o primeiro hospital alternativo não tenha disponibilidade, o *TransplantAgent* avança para o próximo hospital da lista e repete o processo, enviando novos pedidos até encontrar um hospital disponível. Se nenhum hospital na lista responder positivamente, o *TransplantAgent* sinaliza que a urgência máxima não pôde ser atendida no momento e mantém o recetor em estado prioritário na sua lista de recetores.

- **Confirm com "transport" incluído na mensagem:** O *TransportAgent* confirma que o transporte foi bem-sucedido. Se o objeto de transporte for o órgão, o *TransplantAgent* remove o órgão da lista de doadores e recetores, e marca o agente de transporte como disponível. Se o transporte envolver o paciente, nos casos em que foi necessário o deslocar, o *TransplantAgent* regista que o transporte do paciente foi bem-sucedido.

5.3.7 ResendOrgan

Este *behaviour* funciona como uma medida de contingência para garantir que, caso não haja pacientes compatíveis com um órgão em dado momento, a informação sobre este órgão disponível seja periodicamente reenviada (*PeriodicBehaviour*). Ao ser executado periodicamente, o *TransplantAgent* verifica se surgiram novos *Recipients* que atendem aos critérios de compatibilidade com o órgão. Se for o caso, a informação sobre a disponibilidade do órgão serão enviada novamente para garantir que ele seja alocado a um paciente compatível, aumentando a probabilidade de sucesso no processo de

transplante. Este processo de verificação contínua, é uma forma de garantir que o órgão não seja desperdiçado, mesmo que inicialmente não haja uma correspondência.

5.3.8 MonitoringWaitlist

Este *behaviour* funciona como uma medida de contingência para garantir que, caso não haja transportes disponíveis no momento da alocação, é procurado um transporte para o órgão alocado o mais rapidamente possível. Neste cenário, o hospital do recetor já confirmou ter condições para realizar o transplante, apenas há falta de transportes para levar o órgão doado até ao local. Deste modo, a informação do dador e do recetor é adicionada a uma lista de espera, pertencente ao agente Transplante. Ao ser executado periodicamente, o *MonitoringWaitlist* verifica se já há um transporte disponível para efetuar a entrega do órgão e em caso afirmativo, efetua o pedido ao transporte, verificando sempre qual é o transporte que se encontra mais próximo do hospital onde se encontra o órgão doado (no caso de haver vários transportes disponíveis neste momento). Por fim, remove a informação referente a esta alocação da lista de espera. Caso, no momento de execução deste *behaviour* ainda não haja um transporte disponível, a informação permanece na lista de espera. Este processo permite garantir que o órgão não é desperdiçado por indisponibilidades de transporte temporárias.

5.3.9 PatientReceives

Este *behaviour* cumpre apenas o propósito de receber as informações enviadas pelo Agente Transplane para o Agente Recetor, sendo um *behaviour* cíclico. As mensagens recebidas abordam o estado da alocação, nomeadamente uma mensagem de confirmação, caso tenha sido alocado um dador, e uma mensagem informativa, caso haja atrasos no transporte do órgão, caso o respetivo transporte do órgão doado tenha ido para lista de espera por indisponibilidade dos serviços de transporte.

6 Conclusão

Através da integração de diferentes agentes com comportamentos específicos, foi possível criar uma solução capaz de coordenar a doação e transporte de órgãos de uma maneira eficiente. Além disso, ao considerar fatores críticos como a compatibilidade, urgência e logística, o sistema tornou-se mais robusto, permitindo uma tomada de decisão mais informada e a aplicação de estratégias de contingência para lidar com falhas.

Entretanto, há várias melhorias que podem ser implementadas para otimizar a eficiência e a precisão das decisões, aprimorando a experiência de todos os envolvidos. Uma dessas melhorias seria a otimização da alocação de órgãos, com o refinamento dos algoritmos de compatibilidade e urgência, incluindo a consideração de fatores de histocompatibilidade, de forma a priorizar de maneira mais eficaz os receptores com maior necessidade. Além disso, uma abordagem mais robusta para priorização com base no nível de urgência poderia ser implementada, utilizando uma pontuação que leve em conta não apenas a compatibilidade, mas também o tempo de espera e a gravidade do caso, garantindo que os receptores mais críticos sejam priorizados de forma justa e eficiente. Outra melhoria importante seria a criação de relatórios dinâmicos sobre o desempenho do sistema, como tempos médios de transporte, taxa de sucesso de alocação de órgãos e taxa de falhas. A recolha e análise destes dados permitiriam ajustes contínuos no sistema, tornando-o mais ágil e eficiente, além de fornecer uma visão clara sobre os pontos de melhoria para otimização de processos e decisões.

Em conclusão, o sistema desenvolvido demonstrou a capacidade de coordenar de forma eficiente a doação, alocação e transporte de órgãos, considerando aspectos cruciais como compatibilidade, urgência e logística. Ao incorporar uma abordagem robusta para a tomada de decisões e estratégias de contingência, foi possível criar uma solução que contribui para otimizar o processo de alocação de órgãos e melhorar a eficiência do sistema como um todo. Embora haja sempre espaço para melhorias, o trabalho realizado oferece uma base sólida para futuros aprimoramentos e ampliações, assegurando que o sistema seja cada vez mais eficaz no suporte a decisões críticas e na gestão de recursos limitados.

Referências

- [1] Instituto Português do Sangue e da Transplantação. *Em 2023, Portugal registou resultados notáveis na atividade de doação e transplantação de órgãos, tecidos e células.* Acedido em 4 de janeiro de 2023. 2023. URL: <https://www.ipst.pt/index.php/pt/comunicacao/destaques-ipst-ip/49-ipst-newsletter/907-em-2023-portugal-registou-resultados-notaveis-na-atividade-de-doacao-e-transplantacao-de-orgaos-tecidos-e-celulas>.