



Universidade do Minho  
Escola de Engenharia

# Agentes e Sistemas Multiagentes

Mestrado Integrado em Engenharia Biomédica  
Informática Médica

Sistemas Inteligentes

2<sup>o</sup> semestre

2020/2021

**Autor:**

A83624 João Miguel da Silva Alves

**Docentes:**

Paulo Novais

Filipe Gonçalves

Braga

11 de abril 2021

## **Resumo**

A definição de agente é cada vez mais importante tanto no área da Inteligência Artificial (IA) como na de engenharia de computação. O objectivo deste relatório é fazer compreender o leitor da importância, nomeadamente as vantagens e as desvantagens da utilização destes agentes nas mais variadas situações. Para tal, é relevante entender a classificação e a semântica dos agentes e Sistemas Multiagentes, bem como o seu design e construção. Ao longo do relatório é também referido o relacionamento destes sistemas com as várias áreas de IA, bem como entrevistas e conferências fundamentais sobre esta simbiose. Além disso, refere-se uma visão geral dos vários domínios onde são aplicados estes sistemas, tal como as tecnologias ou software que são utilizados para a sua implementação, dando ênfase a tecnologias e ideias no domínio da engenharia biomédica. Por fim, ocorre uma descrição de duas ideias relativamente a possíveis manipulações e implementações de Sistemas Inteligentes e menciona-se as diferentes questões éticas que dizem respeito à sua utilização no presente e futuro.

**Keywords:** Agente, Inteligência Artificial, Sistemas Multiagentes, tecnologias, engenharia biomédica, questões éticas.

# Índice

<b>1</b>	<b>Introdução</b>	<b>6</b>
<b>2</b>	<b>Agentes</b>	<b>7</b>
2.1	Agentes Inteligentes . . . . .	7
2.2	Agentes e ambiente . . . . .	9
2.3	Agentes versus Objectos . . . . .	10
2.4	Agentes e sistemas especialistas . . . . .	11
2.5	Agentes como Sistemas Intencionais . . . . .	11
2.6	Agentes de raciocínio dedutivo . . . . .	12
2.7	Agentes com raciocínio prático . . . . .	13
2.8	Arquiteturas de agentes . . . . .	14
<b>3</b>	<b>Sistemas Multiagentes</b>	<b>18</b>
3.1	Conceito e estado da arte de Sistemas Multiagentes . . . . .	18
3.2	Comunicação entre Agentes . . . . .	20
3.3	Plataformas de Comunicação entre Agentes de Software . . . . .	21
3.4	Linguagens de Comunicação entre Agentes de Software . . . . .	22
3.5	Ferramentas para a Construção de Agentes . . . . .	26
3.6	Vantagens e desvantagens da utilização de sistemas Multiagentes e seu software . . . . .	28
<b>4</b>	<b>Metodologias de software</b>	<b>30</b>
4.1	SMA centralizado . . . . .	30
4.2	SMA descentralizado . . . . .	31
4.3	SMA Aberto . . . . .	32
4.4	SMA Fechado . . . . .	33
<b>5</b>	<b>Sistemas Multiagentes e IA</b>	<b>35</b>
5.1	Inteligência Artificial Distribuída . . . . .	35
5.2	Conferências e Palestras sobre IAD e SMA . . . . .	37
<b>6</b>	<b>Análise crítica de aplicações/projetos existentes e plataformas disponíveis</b>	<b>38</b>
<b>7</b>	<b>Aplicabilidade no domínio da Engenharia Biomédica</b>	<b>41</b>
7.1	Reforço No Diagnóstico . . . . .	41
7.2	Alertas E Notificações . . . . .	42

7.3	Modelos Algorítmicos na Medicina Preventiva . . . . .	43
<b>8</b>	<b>Ideias para projetos futuros sobre Sistemas Multiagentes</b>	<b>44</b>
8.1	Agentes para monitorizar as filas de espera nas instituições de saúde . . . . .	44
8.2	Agentes para monitorizar as condições de um utilizador . . . . .	45
<b>9</b>	<b>Sistemas Multiagentes no futuro e questões éticas</b>	<b>47</b>
<b>10</b>	<b>Conclusão</b>	<b>48</b>
	<b>Referências</b>	<b>49</b>

## Índice de Figuras

1	Agente e seus sensores e atuadores . . . . .	7
2	Agente e relação com o ambiente . . . . .	9
3	Agentes intencionais e especialistas . . . . .	12
4	Agentes de raciocínio dedutivo . . . . .	13
5	Agentes com raciocínio prático . . . . .	14
6	Agentes e sua classificação . . . . .	14
7	Agentes reativos . . . . .	15
8	Agentes deliberativos . . . . .	16
9	Arquitetura híbrida . . . . .	16
10	Arquitetura BDI . . . . .	17
11	Tipos de coordenação de um SMA . . . . .	19
12	Sistemas Multiagente . . . . .	20
13	Estrutura de uma mensagem em KQML . . . . .	23
14	Arquitetura da linguagem ACL . . . . .	24
15	Alguns parâmetros das mensagens ACL . . . . .	25
16	Exemplo da linguagem XML . . . . .	25
17	Arquitetura Jade . . . . .	26
18	SMA centralizado . . . . .	31
19	SMA descentralizado . . . . .	32
20	Inteligência artificial distribuída . . . . .	36
21	ISMSI 2021 — 5th International Conference on Intelligent Systems . . . . .	37
22	RoboViz . . . . .	39
23	RoboCanes . . . . .	39
24	MoCap . . . . .	40
25	Imagens de mamografias retiradas de [1] . . . . .	41
26	Sistema AIDA . . . . .	42
27	Imagens de MRI adquiridas de [2] . . . . .	43
28	Class diagram do projeto . . . . .	44
29	Activity Diagram do projeto . . . . .	45
30	Class diagram do projeto . . . . .	46
31	Activity Diagram do projeto . . . . .	46

# 1 Introdução

Na atualidade estamos rodeados de informação que pode ser muito importante para se tomar decisões de acordo com estímulos ou deveres de cada um. Desta forma, a informação é o resultado do processamento e manipulação dos dados que se encontram à nossa volta, resultando, assim, na manipulação quantitativa ou qualitativa de um sistema, seja pessoa, objeto ou animal.

Desde os jogos de computador a sociedades humanas, muitos são os fenómenos naturais e/ou artificiais que podem ser representados como Sistemas Multiagentes. Com o decorrer do tempo, todos estes sistemas provaram ser uma ferramenta realmente poderosa para modelar e compreender fenómenos em áreas como economia e comércio, saúde, indústria e ciências sociais. Contudo, apesar de os agentes inteligentes existirem há vários anos, o seu impacto significativo na sociedade está a dar os primeiros mas importantes passos.

No que diz respeito à tomada de decisão, a utilização de agentes ou Sistemas Multiagentes é crucial no auxílio às pessoas. Este fenómeno pode ser classificado como um processo cognitivo, no qual uma pessoa escolhe e executa uma ação baseada em diferentes cenários, ambientes e fatores. Para se ter a perceção de todos os cenários possíveis numa tomada de decisão é necessário que se tenha a possibilidade de especular sobre todos os casos, de forma a que se possa medir os prós e contras consoante a ação que se realiza.

Os Sistemas Inteligentes não são apenas técnicas de software com o único propósito de ajudar os humanos na tomada de decisão, visto que são capazes de muito mais. Os vários tipos e definições de sistemas de agentes e multiagentes, bem como as suas vantagens e desvantagens vão sendo descritas ao longo do relatório. É de salientar que o grande objectivo da utilização destes sistemas não é substituir o humano nas suas tarefas, mas sim auxiliá-lo nas mesmas, de forma a permitir um aumento de eficiência e melhoria nos resultados finais.

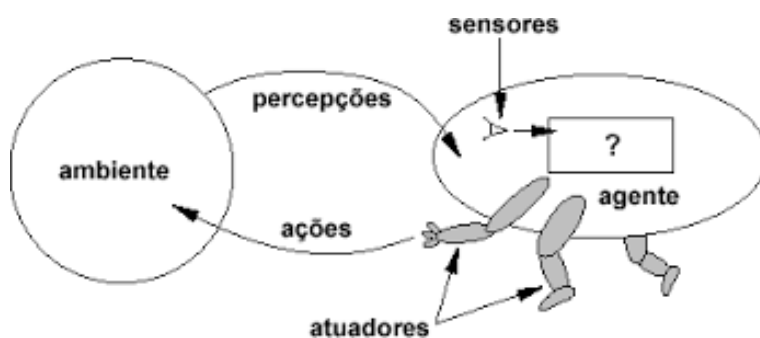
O relacionamento de sistemas inteligentes com as diferentes áreas de IA é cada vez mais notório e irá permitir atingir resultados nunca esperados nesta área. No presente relatório salienta-se o propósito desta relação e objetivos que se pretendem ser atingidos no domínio das diferentes áreas do saber, fundamentalmente na área da saúde.

A sociedade deverá interpretar a implementação de sistemas inteligentes não como um fim, mas um meio para atingir esse fim, uma melhoria na qualidade de vida para todos os que vivem no mundo.

## 2 Agentes

Um agente assume-se como uma figura metafórica utilizada em inúmeras áreas do conhecimento humano, que vão desde a Psicologia, a Economia e a Sociologia, até às Ciências da Computação. Foram apresentadas, pela primeira vez, entidades designadas como objectos auto-contidos, interactivos e concorrentes, que comunicavam por troca de mensagens e que demonstravam possuir um certo tipo de comportamento. Designaram-se esses objectos por actores. O avanço tecnológico, em matéria de computação e de comunicação, permitiu o aparecimento da Inteligência Artificial Distribuída, passível de se subdividir, em termos da resolução de problemas, em Sistemas de Processamento Distribuído e nos Sistemas Multiagente [3].

O conceito de agente tem sido difícil de concretizar, uma vez que as tarefas e os objetivos para os quais os agentes são usados diferem. Dessa forma, existem inúmeras definições do que é um agente, no entanto todas elas prendem-se com inúmeros e distintos contextos de utilização e de desenvolvimento de aplicações em que esses agentes são utilizados. Por um lado, de acordo com a literatura, um agente é algo que é capaz de agir, capaz de produzir alguma coisa. Por outro lado, um agente é também algo que é capaz de obter conhecimento e informação a partir de sensores que respondem a estímulos e atuam no ambiente de onde obtiveram esse conhecimento através de atuadores [3], **Figura 1**. Por esse motivo, um agente é um componente persistente e ativo, o qual raciocina, atua e comunica com outros agentes [3].



**Figura 1:** Agente e seus sensores e atuadores

### 2.1 Agentes Inteligentes

A noção de agente ou Sistema Multiagente pode ser considerada como fraca ou forte, dependendo das propriedades intrínsecas ao agente ou ao sistema. Particularmente, um agente ou Sistema Multiagente é considerado fraco quando não apresenta um conjunto mínimo de

atributos ou se apresentar um certo conjunto de atitudes [3] como, por exemplo:

- **Autonomia:** os agentes operam sem a intervenção de outros e têm controlo sobre as suas acções e o seu estado de conhecimento interno;
- **Reactividade:** os agentes têm percepção do que ocorre no seu universo de discurso e respondem adequadamente a mudanças ocorridas nesse ambiente;
- **Pro-actividade:** os agentes são capazes de tomar uma iniciativa própria, conduzindo as suas próprias acções segundo um comportamento que é dirigido por objectivos;
- **Sociabilidade:** os agentes interactivam com outros agentes, comunicando-se entre si, competindo ou cooperando na resolução de problemas.

Os atributos anteriormente mencionados representam a noção fraca de agente e que está intimamente ligada à Computação Distribuída e IA. Neste domínio, o agente é considerado como um processo de automação, que é implementado para a resolução de problemas em ambientes distribuídos.

A outra classificação de agente ou Sistemas Multiagente é a classificação forte, uma vez que se passa a ter entidades passíveis de desenvolver a sua própria consciência apresentando um conjunto de mais-valias como a perceptibilidade, a sentimentalidade e o emocionismo [3]. O conjunto mínimo de atributos para se considerar um agente como forte são os seguintes:

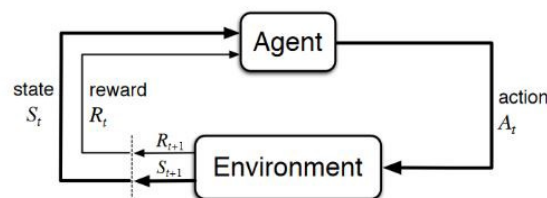
- **Mobilidade:** um agente diz-se móvel quando se movimenta através da rede executando as tarefas de que foi programado e cumprindo objectivos necessários;
- **Intencionalidade:** capacidade que o agente apresenta para a definição de objectivos, tal como de estratégias para os atingir;
- **Aprendizagem:** capacidade que o agente apresenta para adquirir conhecimento. A actualização da base de conhecimento é feita através da assimilação de padrões de comportamento ou de preferências manifestadas pelo próprio agente.
- **Competência:** um agente é competente quando conduz com sucesso e eficácia as tarefas que é programado de realizar. A competência está normalmente relacionada com a confiança depositada no agente, por eventuais terceiros.
- **Veracidade:** um agente não fornece, de forma propositada, informação falsa.



- **Racionalidade:** um agente racional não aceita executar tarefas que lhe pareçam impossíveis de realizar, contraditórias com os seus princípios (e.g., ética), ou quando não são compensados em termos do risco, custo e esforço envolvido.
- **Emotividade:** certas características próprias do ser humano têm vindo a migrar e a constituir-se como parte constituinte de agentes e Sistemas Multiagente.

## 2.2 Agentes e ambiente

É de extrema importância salientar que o ambiente apresenta enorme relação para com o agente. Dessa forma, torna-se necessário saber que tipos de classificação de ambientes existem de modo a compreender a maneira como o ambiente interage com o agente, **Figura 2**.



**Figura 2:** Agente e relação com o ambiente

Russell e Norvig sugerem a seguinte classificação de propriedades do ambiente [4]:

- **Acessível versus inacessível:** Um ambiente acessível é aquele em que o agente pode obter informações completas, precisas e atualizadas sobre o estado do ambiente, sendo que a maioria dos ambientes do mundo real (incluindo, por exemplo, o mundo físico quotidiano e a Internet) são inacessíveis nesta definição;
- **Determinístico versus não determinístico:** Um ambiente determinístico é aquele em que qualquer ação tem um único efeito garantido, ou seja, não há incerteza sobre o estado que resultará da realização de uma ação;
- **Estático versus dinâmico:** Um ambiente estático é aquele que pode ser considerado inalterado, exceto pelo desempenho de ações por parte do agente. Por outro lado, um ambiente dinâmico é aquele que possui outros processos e que, por isso, está constantemente a mudar e que está para além do controle do agente. O mundo físico é um ambiente altamente dinâmico, tal como a Internet;
- **Discreto versus contínuo:** Um ambiente é discreto se houver um fixo ou finito número de ações nele. Caso contrário será contínuo.

Quanto mais acessível for um ambiente, mais simples será construir agentes que operaram efetivamente dentro dele. Um 'bom' agente é aquele que toma as decisões 'certas' [4].

A classe geral de ambientes mais complexa são aqueles que são inacessíveis, não determinísticos, dinâmicos e contínuos. Ambientes que têm essas propriedades são frequentemente chamados de abertos [4]. As propriedades ambientais têm um papel na determinação da complexidade do agente e seu processo de design, contudo não são de forma alguma os únicos fatores importantes. Além das propriedades ambientais, a forma como o agente interage com o ambiente é fundamental para a natureza do agente.

## 2.3 Agentes versus Objectos

A distinção entre agentes e objectos é rara e, de certa forma, existe uma falta de rigor na sua definição. Há uma tendência de pensar nos objetos como 'atores' e dotá-los com intenções e habilidades humanas [3]. Contudo objetos são definidos como entidades computacionais que encapsulam algum estado e são, portanto, capazes de realizar ações ou métodos neste estado e se comunicar por mensagens entre si. Embora existam semelhanças óbvias, também existem diferenças significativas entre agentes e objetos.

A principal diferença entre um entidade e outra depara-se na autonomia. Apesar de os objetos terem completo controlo sobre o seu estado de conhecimento, não o têm relativamente ao seu estado de funcionamento. Por sua vez, os agentes possuem essas duas propriedades, sendo, por isso, autónomos e podem perfeitamente recusar qualquer pedido de comunicação ou assistência.

Outra diferença muito importante baseia-se na comunicação. Programadores familiarizados com linguagens orientadas a objetos, como Java, C++ poderão não conseguir ver algo de inovador na ideia de agentes. No entanto, os objetos comunicam-se através da indicação dos métodos que podem ser invocados a partir da classe desses objetos, tornando estes procedimentos intrínsecos dos objectos em que se desencadeiam. Nos agentes, a mensagem e a linguagem a utilizar são definidas ao nível do colectivo ou comunidade de agentes, e não individualmente.

Em termos de comportamento, os agentes são flexíveis, cultivam a autonomia, percebem o seu ambiente e agem de acordo com os seus objectivos [3].

De uma forma geral, um objeto ou a programação orientada aos objetos consiste num dos vários paradigmas da programação, enquanto que a utilização de agentes ultrapassa o conceito de programação orientada aos objetos e passa a usar esse paradigma para representar entidades com propriedades e atributos que foram anteriormente mencionados.

## 2.4 Agentes e sistemas especialistas

Um sistema especialista é aquele que é capaz de resolver problemas ou fornecer conselhos nalgum domínio rico em conhecimento. Eles obtêm informações não por meio de sensores, mas por meio de um utilizador agindo como intermediário. Ao contrário dos agentes, eles não agem em nenhum ambiente, mas sim dão feedback ou conselhos a um terceiro. Além disso, os sistemas especialistas geralmente não são capazes de cooperar com outros agentes. As principais diferenças entre agentes e sistemas especialistas são as seguintes:

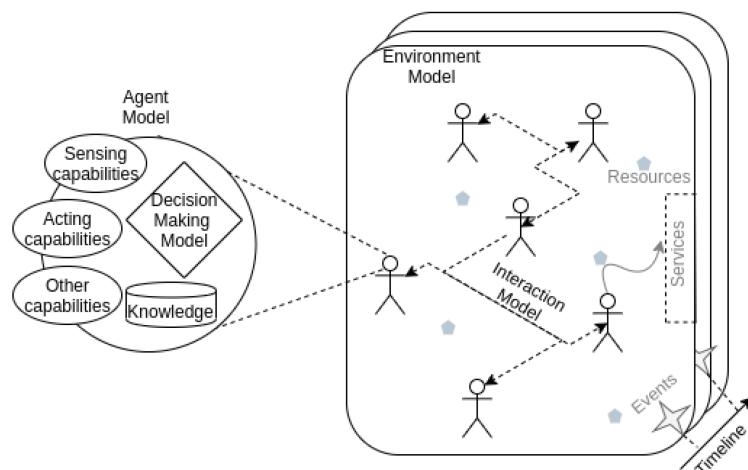
- Sistemas especialistas 'clássicos' não estão conectados a nenhum ambiente no qual atuam, mas agem por meio de um usuário que funciona como um 'intermediário';
- Os sistemas especialistas geralmente não são capazes de comportamento reativo e proativo;
- Sistemas especialistas geralmente não são equipados com habilidade social, no sentido de cooperação, coordenação e negociação.

Apesar destas diferenças, alguns sistemas especialistas, particularmente aqueles que realizam tarefas de controle em tempo real, parecem-se muito com agentes.

## 2.5 Agentes como Sistemas Intencionais

Uma abordagem comum adotada na discussão de sistemas de agentes é a posição intencional. Com esta abordagem, é possível incutir nos agentes estados mentais: crenças, desejos, esperança, etc... Essas declarações fazem uso de uma psicologia popular, pela qual o comportamento humano é previsto e explicado por meio da atribuição de atitudes, como acreditar e querer, esperar, temer, etc... Um agente pode, assim, ser caracterizado como um sistema intencional, o qual utiliza essas propriedades para guiar as suas atitudes no processo de resolução de problemas [3].

Para muitos investigadores, a ideia de programar sistemas de computador em termos de noções psicológicas, como crença, desejo e intenção são uma componente chave de sistemas baseados em agentes, **Figura 3**.



**Figura 3:** Agentes intencionais e especialistas

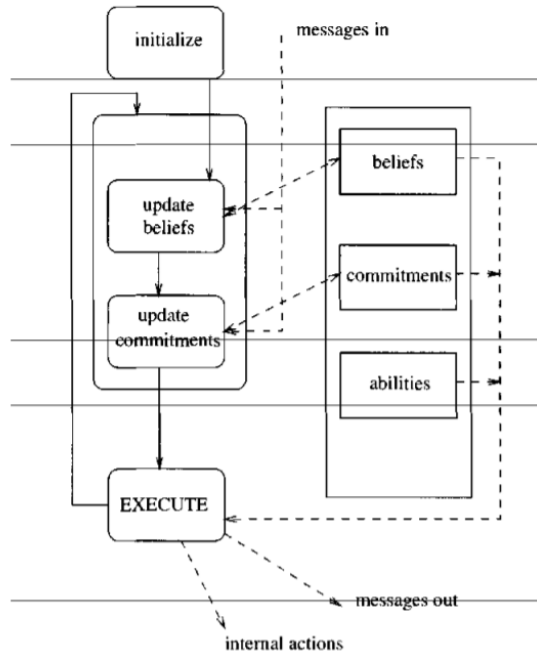
## 2.6 Agentes de raciocínio dedutivo

Agentes de raciocínio dedutivo utilizam representações simbólicas, ou seja, fórmulas lógicas, e recorrem à manipulação sintática dessas mesmas formas, conseguindo, assim, a dedução ou prova de teoremas. Por exemplo, existem inúmeras formas de implementar um sistema de controlo para um robô, contudo uma maneira é fornecer uma descrição ou representação do ambiente em que deve operar, **Figura 4**.

Nestes agentes, o estado interno é considerado uma base de dados de fórmulas da lógica clássica. Por exemplo, uma base de dados do agente pode conter fórmulas lógicas, as quais pode ser usadas para deduzir outras fórmulas. O objetivo principal é a partir de axiomas, os quais são implementados no sistema do agente como verdades absolutas, sendo, portanto, o seu ambiente de representação de conhecimento, conseguir originar novo conhecimento que deve ser retido pelo agente. Desta forma, é possível aumentar a percepção do que o rodeia.

Yoav Shoham propôs um 'novo paradigma de programação, baseado numa visão da computação 'que ele chama de programação orientada a agentes [4]. A motivação por trás da proposta é que os humanos usam conceitos como um mecanismo de abstração para representar propriedades de sistemas complexos da mesma forma que usamos essas noções para descrever e explicar o comportamento humano. Por isso, pode ser útil use estes conceitos para programar máquinas. A ideia de programar sistemas de computador em termos de estados mentais foi articulado em Shoham [4]. A primeira implementação do paradigma de programação orientada a agentes foi a Linguagem de programação AGENTO. Neste idioma, um agente é especificado em termos de um conjunto de capacidades (coisas que o agente pode fazer), um conjunto de crenças iniciais, um conjunto de compromissos iniciais e um conjunto de regras

de compromisso.



**Figura 4:** Agentes de raciocínio dedutivo

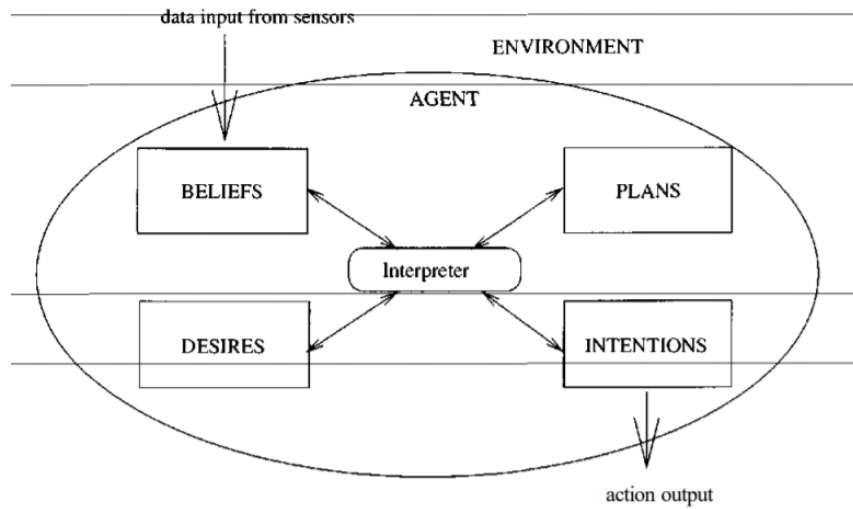
## 2.7 Agentes com raciocínio prático

O modelo particular de tomada de decisão é conhecido como raciocínio prático. Raciocínio prático é o raciocínio direcionado a ações, ou seja, o processo de descobrir o que fazer a seguir.

Ao contrário do tipo de agentes anteriores, os quais promovem o raciocínio teórico, estes pretendem representar conhecimento já utilizado para descobrir o que fazer a seguir. Por este motivo, existem diferenças entre os dois tipos de raciocínio. O raciocínio teórico é direcionado para as crenças. O raciocínio prático humano consiste em pelo menos duas atividades distintas. A primeira delas envolve decidir que estado de coisas se quer alcançar e o segundo processo envolve decidir como se quer alcançar esses mesmos estados de coisas, **Figura 5**.

Intenções desempenham os seguintes papéis importantes em raciocínio prático:

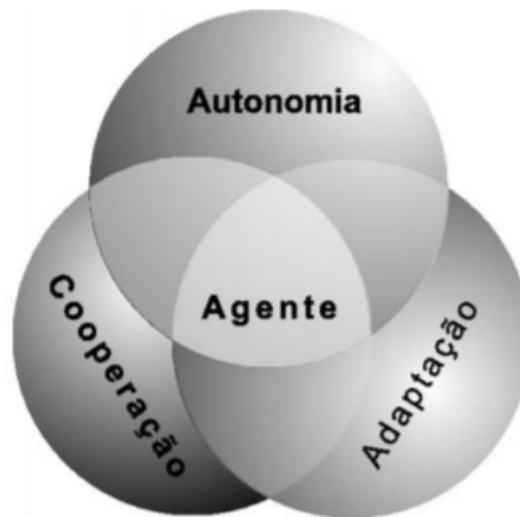
- As intenções impulsionam o raciocínio para decidir como queremos alcançar o estados de coisas;
- As intenções restringem a deliberação futura;
- As intenções influenciam as crenças nas quais se baseia o raciocínio prático futuro.



**Figura 5:** Agentes com raciocínio prático

## 2.8 Arquiteturas de agentes

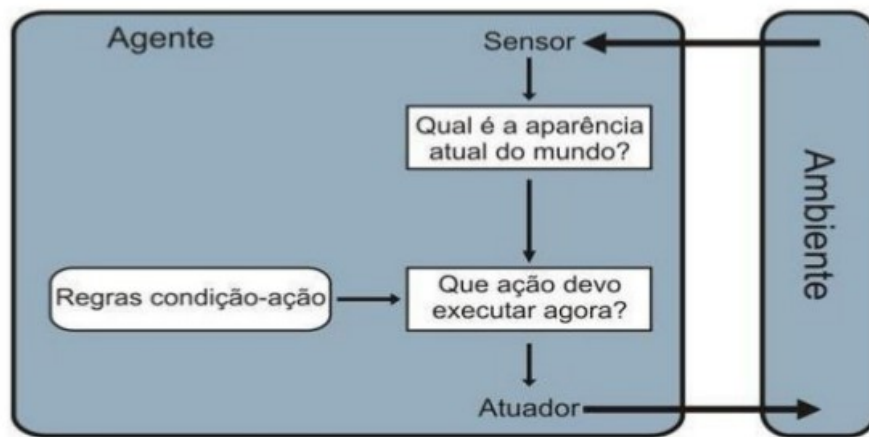
Os agentes podem também ser classificados, tendo em conta se possuem ou não um conjunto mínimo de atributos, tais como a autonomia, a cooperação e a aprendizagem/adaptação. Dessa forma, são classificados como agentes colaborativos, agentes de interface, agentes de aprendizagem ou agentes inteligentes [3], **Figura 6**.



**Figura 6:** Agentes e sua classificação

### Arquitetura reativa

Um agente do tipo reativo não possui qualquer representação simbólica do universo de discurso, nem apresenta formas de raciocínio complexas [3]. Por esse motivo, consiste num autômato inserido no meio ambiente que o rodeia. Num agente deste tipo, a percepção sobre o mundo à sua volta ocorre por meio de sensores, isto é, apenas recebe informação do tipo atômico e reage aos estímulos que são capturados por esses mesmos sensores. Este sistema de agente pode ser visto como o resultado da invocação de regras do tipo condição/acção, **Figura 7**.



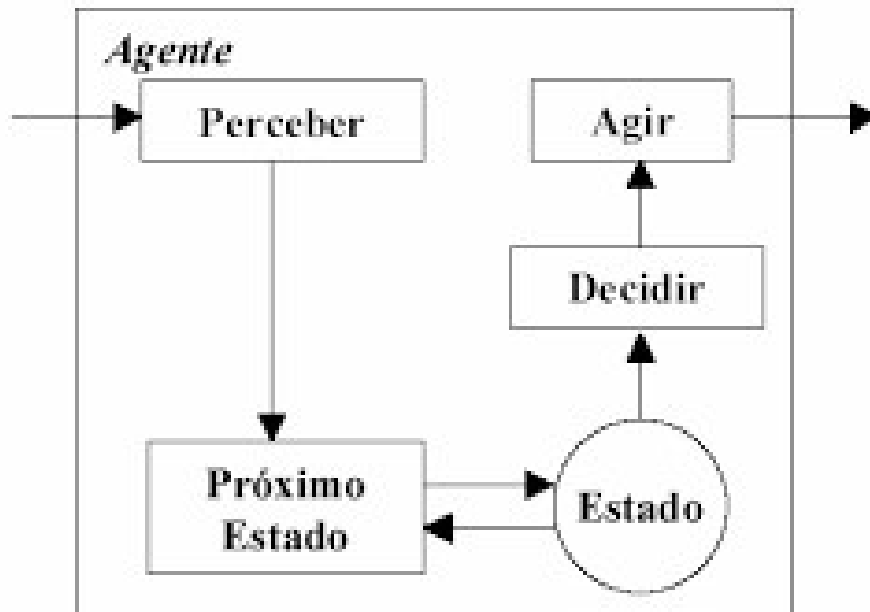
**Figura 7:** Agentes reativos

### Arquitetura deliberativa

Um agente deliberativo ao contrário de um agente reativo possui representação simbólica do universo de discurso e o seu estado de conhecimento pode ser modificado por alguma forma de raciocínio lógico-matemático [3]. Apresenta também formas de raciocínio complexas, que lhe permitem resolver problemas em que há precedência na acção, **Figura 8**.

A ideia de agente deliberativo baseado em puro raciocínio lógico-matemático é extremamente chamativo. No entanto, dois importantes problemas têm-se vindo a colocar:

- Um problema de transposição e/ou representação: como traduzir o mundo real em termos de um programa em lógica ou teoria?
- Um problema de raciocínio: como levar os agentes a raciocinar?



**Figura 8:** Agentes deliberativos

### Arquitetura híbrida

Os agentes que utilizam a arquitetura reativa são na sua maioria das vezes simples de desenvolver, uma vez que apenas reagem a estímulos através de ações implementadas pelo programador. No entanto, pelo facto de não apresentarem uma representação simbólica do ambiente que os rodeia tornam-se pouco autónomos e difíceis de se adaptarem [3]. Por outro lado, os agentes deliberativos conseguem-no fazer, mas a um custo elevado, visto que são incapazes de reagir de uma forma rápida a estímulos vindos do exterior. Por este motivo, a implementação de uma arquitetura híbrida será a mais vantajosa, já que interliga as vantagens das duas arquiteturas anteriores, **Figura 9**.



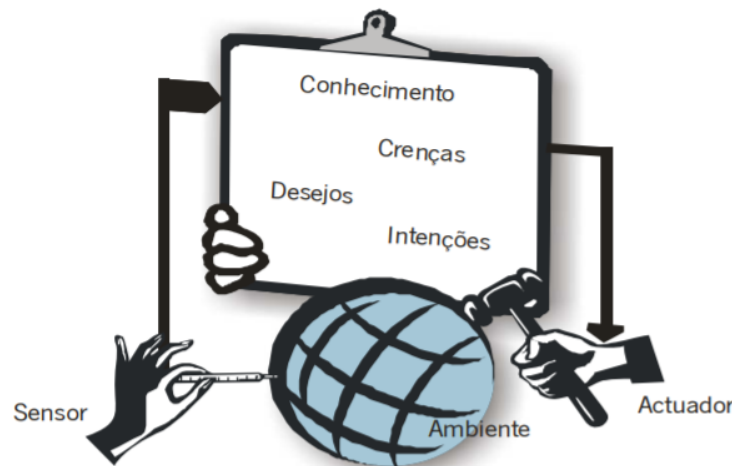
**Figura 9:** Arquitetura híbrida



## Arquitetura BDI

Uma arquitectura para agentes de software que suporte a concretização das Crenças, Desejos e Intenções (uma arquitectura BDI – Beliefs, Desires, Intentions) de um agente, é claramente uma arquitectura deliberativa [4], **Figura 10**. Nesta implementação pretende-se que o agente se adequa à descrição do estado interno de conhecimento com base nos seus estados mentais, estados estes de que o agente fará uso para determinar o seu curso de acção.

- Crenças - uma opinião ou convicção firmemente cimentada;
- Desejos - O desejo é sinónimo de motivação;
- Intenções - Um conjunto de intenções é o resultado de uma deliberação.



**Figura 10:** Arquitetura BDI

### Exemplo de um agente reativo:

Num hospital é necessário arquivar inúmeros ficheiros de diferentes pacientes com diferentes medições e prescrições.

Um agente reativo poderia ser implementado de forma a que quando um ficheiro de um determinado paciente fosse arquivado na base de dados do hospital, esse mesmo agente analisaria os valores por exemplo de diabetes presentes. Se fossem superiores a um valor limite previamente estabelecido, então seria notificado como importante ou urgente para que na próxima consulta ou exame desse paciente, o médico soubesse de antemão o que estaria de errado com o paciente de uma forma rápida e eficaz. Estamos, portanto, a falar de um agente que sob determinados estímulos (valores superiores a um valor limite fixo) reage e efetua uma acção.

## 3 Sistemas Multiagentes

### 3.1 Conceito e estado da arte de Sistemas Multiagentes

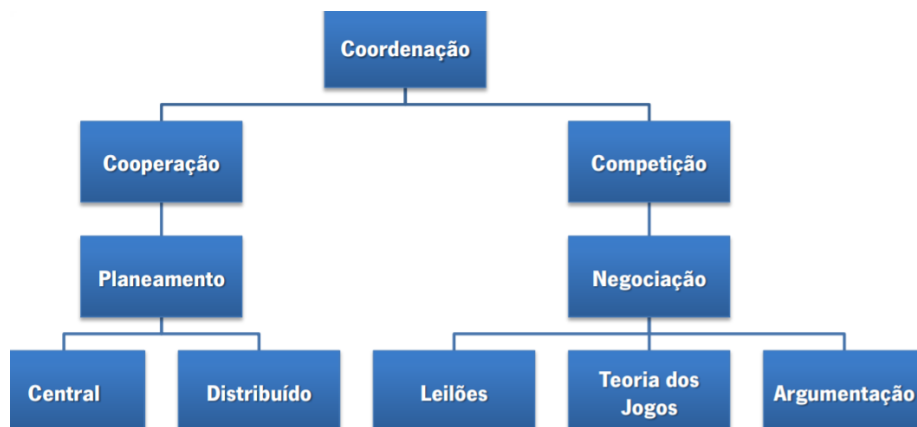
A ideia de um Sistema Multiagente é muito simples. Um agente é um sistema de computador que é capaz de agir independentemente do seu proprietário. Noutras palavras, um agente pode descobrir por si mesmo o que precisa de fazer a fim de satisfazer os seus objetivos de design, ao invés de ter que ser dito explicitamente o que fazer a qualquer momento. Um Sistema Multiagente é aquele que consiste numa série de agentes, que interagem uns com os outros, normalmente trocando mensagens através de alguma infraestrutura de rede informática, **Figura 12**.

A criação de Sistemas Multiagentes advém da área da Inteligência Artificial Distribuída (IAD). A Inteligência Artificial Distribuída (IAD), também conhecida como Inteligência Artificial Descentralizada, com um enfoque diferente da Inteligência Artificial (IA) tradicional, tenta dividir um problema em problemas pequenos e mais simples [3]. Desta forma, torna-se possível a sua utilização em áreas como: comércio eletrónico, networking, programação e Sistemas de Multiagentes. Contudo, construir um SMA requer muito mais do que a compreensão de protocolos de comunicação e paradigmas computacionais associados à programação distribuída. É preciso dominar tecnologias que têm a ver com as linguagens de comunicação entre agentes, arquiteturas de agentes, resolução de problemas em ambientes distribuídos, concepção de protocolos, assim como ser perito em teoria das organizações [3].

Os Sistemas Multiagentes, sendo um conjunto de agentes individuais que tentam trabalhar em grupo de forma harmoniosa, podem ser coordenados de duas formas distintas: coordenação cooperativa e coordenação competitiva, **Figura 11**.

Na coordenação cooperativa existe um processo de tomada de decisão em que as partes envolvidas negociam para que possam alcançar um ou mais objetivos [3]. Neste caso pode-se ter um sistema central em que a decisão passa por um agente responsável pelas decisões finais ou um sistema distribuído, em que cada agente possui a sua própria responsabilidade na tomada de decisão de planeamento.

Na coordenação competitiva existe um processo de decisão em que as partes envolvidas competem tendo em conta um único objetivo [3]. Como exemplos tem-se: leilões, a teoria dos jogos e a argumentação, onde cada agente compete de forma a conseguir vencer outro durante um diálogo mediante determinados argumentos.



**Figura 11:** Tipos de coordenação de um SMA

Um Sistema Multiagente compreende um conjunto de entidades que cooperam de forma a solucionar um dado problema; o que, normalmente, estará para lá das suas capacidades individuais.

A sua implementação só poderá ser fazível se respeitar as seguintes camadas:

- A plataforma de comunicação;
- A linguagem de comunicação;
- A ontologia.

No que diz respeito à plataforma de comunicação, existem várias plataformas de software que foram desenvolvidas ao longo dos anos, que permitem que diferentes agentes possam comunicar entre si. Correspondem a um meio físico de transmissão de informação entre os vários agentes. A ligação entre o remetente e o endereço pode ser do tipo ponto a ponto, em que o agente remetente envia uma mensagem ao destinatário, identificando-o, ou difusa, em que os potenciais destinatários de uma mensagem constituem o colectivo do sistema.

A linguagem de comunicação é a forma pela qual um agente consegue transmitir uma mensagem para outro agente, devendo, por isso, ser: uma linguagem comum, na forma de interpretar o conhecimento trocado e ter a possibilidade e capacidade de trocar esse conhecimento. As linguagens poderão ser de dois tipos: o tipo procedimental, as quais permitem a troca de programas ou procedimentos, que são executados directa e eficientemente e do tipo declarativo, que permitem a troca de definições e regras, as quais são processadas de forma indirecta. Mais à frente no relatório são mencionadas e explicadas as diferentes linguagens usadas na comunicação entre agentes, nas subsecções "Plataformas de Comunicação entre Agentes de Software" e "Linguagens de Comunicação entre Agentes de Software".

Por fim, as ontologias são também necessárias, uma vez que não basta que exista um formato comum e uma linguagem de comunicação comum. Os agentes precisam de concordar com um modelo conceptual da realidade, que permita capturar o conhecimento próprio dessa mesma realidade e que a permita compreender, i.e., acordar numa ontologia comum [3]. A ontologia fornece por outras palavras o vocabulário de representação para o domínio em questão, e um conjunto de definições e axiomas que restringem o significado dos termos nesse vocabulário, de forma a permitir uma interpretação consistente e única no universo de discurso.



**Figura 12:** Sistemas Multiagente

### 3.2 Comunicação entre Agentes

Os seres humanos desenvolveram uma estrutura de sinais conhecidos como linguagem, a qual permite a comunicação e a interação entre humanos. Como já referido anteriormente, os agentes para se comunicarem também usam uma linguagem própria que pode ser implementada usando diferentes metodologias de software. Num Sistema Multiagente, tal como para os humanos, a comunicação é a fundação para as interações e organização social, uma vez que, sem ela, os agentes são apenas componentes individualizados [3].

Existem muitos modelos de comunicação, contudo todos eles consistem em enviar informações de um remetente para um destinatário. Existem três aspectos para se definir formalmente comunicação: a sintaxe, que corresponde à forma como os símbolos da comunicação estão estruturados, a semântica, que pretende mostrar o que os símbolos significam e a ontologia, que

é como os símbolos são interpretados.

Há dois tipos de mensagens: perguntas e declarações, sendo que qualquer agente, sendo ativo ou passivo, deve ser capaz de aceitar informações. No caso mais simples, a informação é comunicada para um agente externo por meio de uma mensagem de afirmação. Para assumir o papel de ativo ou passivo, o agente deve, adicionalmente, ter a capacidade de responder a uma pergunta, ou seja, ele deve saber aceitar uma pergunta de um outro agente e respondê-la por meio de uma afirmação [3]. Para que ocorra esta troca de mensagens é necessário que protocolos sejam respeitados. Um protocolo é especificado por uma estrutura de dados, nomeadamente:

- Remetente;
- Destinatário;
- Linguagem;
- Funções de codificação e decodificação;
- Ações para serem tomadas pelo(s) destinatário(s).

### 3.3 Plataformas de Comunicação entre Agentes de Software

A plataforma de comunicação é o meio físico para a transmissão da informação entre os agentes.

#### **Plataforma CORBA**

A plataforma CORBA (Common Object Request Broker Architecture) é a arquitetura padrão criada pelo Object Management Group para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos. CORBA permite a colaboração entre sistemas em diferentes sistemas operacionais, linguagens de programação e hardware. O CORBA usa um modelo orientado a objetos. A plataforma CORBA é um exemplo do paradigma de objetos distribuídos [3].

Para construir um sistema que usa ou implementa uma interface de objeto distribuída baseada em CORBA, um programador deve obter ou escrever o código que define a interface orientada ao objeto para a lógica que o sistema usará ou implementará. Normalmente, uma implementação de ORB inclui uma ferramenta chamada de compilador IDL que traduz a interface IDL para o idioma de destino para uso naquela parte do sistema. Um compilador tradicional então compila o código gerado para criar os arquivos de objeto que serão usados na aplicação. Além de fornecer aos usuários uma linguagem e uma especificação de chamada de

procedimento remoto (RPC) neutra para a plataforma, o CORBA define serviços necessários, como transações e segurança, eventos, tempo e outros modelos de interface específicos de domínio.

### **Plataforma DCOM/OLE**

A plataforma DCOM/OLE (Distributed Component Object Model/Object Linking Embedding) da Microsoft privilegia a interoperabilidade entre objectos desenvolvidos segundo diferentes linguagens de programação, tendo em vista o sistema operativo Windows [3]. Possibilita um determinado nível de programação que permite a um computador executar programas pela rede num computador diferente, como se o programa estivesse sendo executado localmente.

### **Plataforma JATLITE**

JATLite (Java Agent Template, Lite) é um pacote de classes e programas Java que permitem aos utilizadores criar rapidamente novos sistemas de agentes de software que se comunicam pela Internet para realizar uma computação distribuída.

Além do código para a criação de agentes, JATLite fornece uma infraestrutura de agente robusta, na qual os agentes se registam com um agente Message Router (AMR), usando um nome e uma senha, para poder trocar mensagens num buffer e transferir arquivos entre outros agentes na Internet, sendo que alguns podem ser miniaplicativos Java) [5]. A comunicação pode ser assíncrona e há suporte para agentes intermitentes. Não há nenhum requisito para instalação de software especial para agentes de host e nenhum host especial é assumido para qualquer agente.

JATLite facilita a construção de agentes que enviam e recebem mensagens usando a linguagem de comunicação KQML. No entanto, os programadores podem construir sistemas em JATLite usando outras linguagens, como as comunicações FIPA ACL que são construídas em padrões abertos da Internet, como TCP / IP, SMTP e FTP. [5].

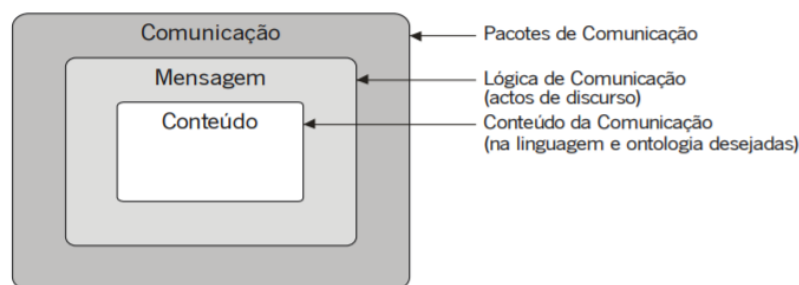
## **3.4 Linguagens de Comunicação entre Agentes de Software**

Muitas são as linguagens que têm sido apresentadas de forma a solucionar o problema da comunicação entre agentes, nomeadamente [3]:

- KIF (Knowledge Interchange Format);
- KQML (Knowledge Query Manipulation Language);
- FIPA ACL (Foundation for Intelligent Physical Agents - Agent Communication Language);
- Ontolingua;
- RDF (Resource Description Format);
- XML (Extended Markup Language);
- BRML (Basic Rule Markup Language);
- DAML (DARPA Agent Markup Language).

### Linguagem KQML

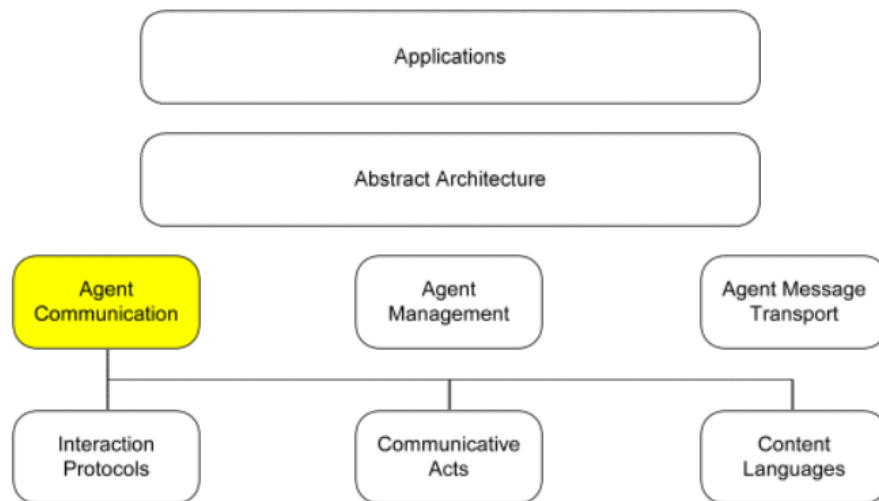
O KQML é um protocolo para troca de informações e conhecimento. As performatives de KQML são modelos de performatives de atos da fala. A semântica do KQML é de domínio independente, enquanto a semântica da mensagem é definida pelos campos content (a própria mensagem), language (a linguagem na qual a mensagem é expressa), e ontology (o vocabulário de “palavras” da mensagem), **Figura 13** [3]. Com isso, o KQML agrupa a mensagem numa estrutura que pode ser entendida por qualquer agente. Para entender a própria mensagem, o receptor deve entender a linguagem e ter acesso à ontologia. Os termos content, language e ontology delimitam a semântica da mensagem. Outros argumentos, incluindo sender, receiver, reply-with e in-reply-to, são parâmetros da mensagem enviada. O KQML assume uma comunicação assíncrona, ou seja, o campo reply-with de um emissor e o campo in-reply-to do agente respondente unem a mensagem de saída com a resposta esperada [4].



**Figura 13:** Estrutura de uma mensagem em KQML

## Linguagem ACL e normas FIPA

As especificações da FIPA Agent Communication lidam com mensagens da linguagem de comunicação do agente (ACL), **Figura 14**, protocolos de interação de troca de mensagens, atos comunicativos baseados na teoria do ato de fala e representações da linguagem do conteúdo, [6].



**Figura 14:** Arquitetura da linguagem ACL

Nos sistemas de agentes FIPA, os agentes comunicam entre si através de mensagens. Os 3 aspectos fundamentais da comunicação baseada em mensagens são: a estrutura das mensagens (norma 61), a representação das mensagens (norma 37), o transporte de mensagens (norma 67) [3].

O único elemento que é obrigatório em todas as mensagens ACL é a performative, embora a maioria das mensagens também tenham os elementos sender, receiver e content, **Figura 15**.

Cada agente tem uma ou mais descrições de transporte (transport-descriptions), que são usadas pelos outros agentes para lhe enviarem mensagens (transport-messages). Cada descrição de transporte indica uma forma possível de transporte da mensagem, como por exemplo IIOP, SMTP, ou HTTP. Um transporte é um mecanismo para transferir mensagens, sendo que o conjunto de descrições de transporte é guardado no tuplo <agent-locator> da entrada respectiva no serviço de diretório de agentes;



Elemento	Categoria de Elemento
Performative	Tipo do ato comunicativo
Sender	Participante na comunicação
Receiver	Participante na comunicação
reply-to	Participante na comunicação
Content	Conteúdo da mensagem
Language	Descrição do Conteúdo
Encoding	Descrição do Conteúdo
Ontology	Descrição do Conteúdo
Protocol	Controlo da conversação
conversation-id	Controlo da conversação
reply-with	Controlo da conversação
in-reply-with	Controlo da conversação
reply-by	Controlo da conversação

**Figura 15:** Alguns parâmetros das mensagens ACL

### Linguagem XML

Extensible Markup Language (XML) é uma linguagem de marcação que define um conjunto de regras para a codificação de documentos num formato que pode ser lido por humanos e por máquina [3]. É, neste momento, a linguagem mais promissora para o armazenamento e a difusão de informação na Web.

A linguagem é amplamente usada para a representação de estruturas de dados arbitrárias, como aquelas usadas em serviços da web, permitindo, assim, que se possa descrever virtualmente qualquer tipo de informação (e.g., entidades, objectos, agentes, estruturas de dados complexos), **Figura 16**.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<tell>
  <sender> agent1 </sender>
  <receiver> agent2 </receiver>
  <ontology> sales </ontology>
  <language> prolog </language>

  <content> sell(productx,100) </content>
  <label> label1</label>
</tell>
```

**Figura 16:** Exemplo da linguagem XML

### 3.5 Ferramentas para a Construção de Agentes

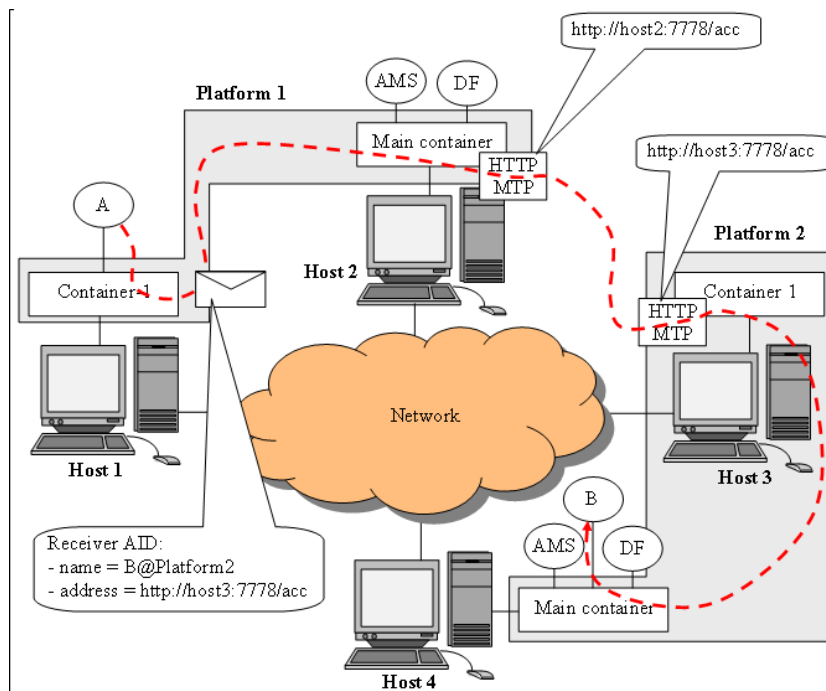
Para que os agentes possam ser devidamente implementados e interligados num sistema são necessárias diferentes ferramentas que permitem a sua construção.

#### Plataforma JADE

A plataforma JADE consiste em containers que podem estar espalhados entre diferentes hosts com um container principal num host executando o serviço RMI. Todos os agentes numa plataforma comunicam-se usando o protocolo RMI. RMI é o mecanismo de comunicação interna a uma plataforma, **Figura 17** [7].

Os agentes JADE também podem-se comunicar entre plataformas separadas, ou seja, o JADE fornece mecanismos para comunicação entre plataformas. Esses mecanismos são baseados em CORBA e usam o protocolo IIOP em vez de RMI. Tal arranjo permite que os agentes JADE conversem com agentes em conformidade com as normas FIPA em qualquer plataforma que suporte os padrões CORBA e FIPA. Esta é a ideia do FIPA: tornar interoperáveis diversos sistemas de agentes.

Além disso, existem outros protocolos para comunicação entre plataformas, sendo o mais usado o protocolo HTTP.



**Figura 17:** Arquitetura Jade

## **ZEUS**

A ferramenta ZEUS, tal como JADE, permite a construção de agentes. Foi inicialmente implementado em JAVA, já que com esta linguagem de programação, consegue-se a execução concorrente (multi-threading), orientação à Internet e corresponde a uma linguagem que pode correr em diferentes sistemas.

Nesta ferramenta cada agente é composto por 3 camadas: uma camada de definição do agente, que é responsável por enunciar e implementar as capacidades particulares de um agente, uma camada organizacional, que define as relações estabelecidas entre conjuntos de agentes e uma camada de coordenação, a qual é responsável por fornecer aos agentes, as técnicas para se conseguirem coordenar e negociar entre si.

Esta plataforma fornece inúmeras ferramentas para implementação de agentes como: Ontology Editor, Task Description Editor, Organisation Editor, Agent Definition Editor, Coordination Editor, Fact/Variable Editor, Constraint Editor e Code Generator, uma API para a execução de tarefas com avaliação de custo, comunicação KQML, algoritmos de planeamento e escalonamento, serviços de páginas amarelas e visualizadores [3].

## **FIPA-OS**

O FIPA-OS cooresponde a um conjunto de ferramentas open-source para o desenvolvimento de agentes, os quais se baseiam no padrão da Foundation for Intelligent Physical Agents (FIPA) [3].

Existem 4 componentes obrigatórios nesta ferramenta: Agent Shell, Task Manager, Conversation Manager e Message Transport Service (MTS). A Agent Shell apresenta um ambiente simples para a implementação de agentes baseados em classes. O Task Manager é responsável por repartir uma tarefa em elementos individuais, conseguindo, assim, obter a capacidade de receber e enviar mensagens independentemente do local onde o código deve ser executado. O Conversation Manager permite a existência de um protocolo específico, que poderá ser FIPA ou outro definido pelo programador. O MTS fornece uma das mais importantes senão a mais importante característica dos agentes inteligentes, que é a comunicação, possibilitando a troca de mensagens entre agentes.

### 3.6 Vantagens e desvantagens da utilização de sistemas Multiagentes e seu software

Na atualidade existem alguns investigadores que se encontram céticos sobre os Sistemas Multiagentes, juntamente com alguns na comunidade internacional de pesquisa em ciências da computação.

Existem alguns na comunidade de sistemas distribuídos que se questionam se os Sistemas Multiagentes são suficientemente diferentes dos sistemas distribuídos 'padrão' para merecer um estudo separado.

Alguns cientistas sociais estão interessados em Sistemas Multiagentes, porque eles fornecem uma ferramenta experimental para modelar sociedades humanas. Além disso, uma abordagem óbvia para o design de smaSs - que são sociedades artificiais - é olhar como uma função particular que funciona nas sociedades humanas e tenta construir o sistema multiagente da mesma maneira. Embora possamos extrair percepções e analogias úteis das sociedades humanas, não se sabe que se possa construir sociedades artificiais exatamente da mesma maneira. Por esse motivo, é notoriamente difícil modelar com precisão o comportamento das sociedades humanas, simplesmente porque eles dependem de muitos parâmetros diferentes. Os Sistemas Multiagentes fornecem uma ferramenta nova e poderosa para modelagem e compreensão das sociedades, enquanto as ciências sociais representam um rico repositório de conceitos para a compreensão e construção de SMAs, contudo são disciplinas bastante distintas.

Sistemas Multiagentes permitem, [8]:

- Um SMA distribui recursos e capacidades computacionais numa rede de agentes interconectados;
- Um SMA permite a interconexão e interoperação de vários sistemas existentes;
- Um SMA modela problemas em termos de componentes-agentes de interação autónomos, o que se está a provar ser uma forma mais natural de representar a alocação de tarefas, planeamento de equipas, preferências do utilizador, ambientes abertos, etc..;
- Um SMA recupera, filtra e coordena globalmente informações de fontes distribuídas;
- Um SMA melhora o desempenho geral do sistema, especificamente nas dimensões de eficiência computacional, confiabilidade, extensibilidade, robustez, capacidade de manutenção, capacidade de resposta, flexibilidade e reutilização.
- Arquiteturas para agentes inteligentes, modelagem de comportamento inteligente;

- Formas ótimas de organização e cooperação;
- Coordenação e colaboração.

Contudo, os Sistemas Multiagentes também apresentam desvantagens para além das mencionadas anteriormente, nomeadamente:

- Previsibilidade, compreensibilidade e controlo limitados;
- Sistema sujeito a acidentes e erros;

## 4 Metodologias de software

Nos Sistemas Multiagentes é necessário ter em conta diferentes metodologias de software, que dizem respeito à forma como se pretende implementar um sistema desse tipo. Isto é, os mecanismos standard e a respetiva organização de agentes aplicados neste tipo de sistemas.

No que diz respeito à implementação de Sistemas Multiagentes, estes diferem das técnicas computacionais tradicionais que são abordadas para por exemplo, programação funcional ou programação orientada a objetos. Por um lado, na programação funcional, a implementação é realizada com o recurso a funções e relacionamento entre elas, nas quais recebem um determinado input e retornam um determinado output consoante o respetivo processamento. Ora, um SMA difere deste tipo de metodologia, uma vez que neste sistema, diferentes agentes realizam uma determinada tarefa que pode ser vista como uma função ou um conjunto de funções, mas não é por um agente ter falhado que todo o sistema falha, ao contrário do que acontece com a programação funcional.

Por outro lado, existe também a noção de programação orientada a objetos, na qual existem classes e pode-se representar diferentes instâncias dessas mesma classe, que se designam por objetos. Um SMA pode-se basear nesta implementação, contudo também apresenta diferenças comparativamente à técnica tradicional de programação orientada a objetos. Isto acontece, pois os agentes possuem propriedades como autonomia e, por isso, podem perfeitamente recusar qualquer pedido de comunicação ou assistência. Os objetos comunicam-se através da indicação dos métodos que podem ser invocados a partir da classe desses objetos, tornando estes procedimentos intrínsecos dos objectos em que se desencadeiam. Nos agentes, a mensagem e a linguagem a utilizar são definidas ao nível do colectivo ou comunidade de agentes, e não individualmente. Portanto, um objeto ou a programação orientada aos objetos consiste num dos vários paradigmas da programação, enquanto que a utilização de agentes ultrapassa o conceito de programação orientada aos objetos [3].

Quatro abordagens podem ser realizadas: um SMA centralizado ou um SMA descentralizado e um SMA aberto ou SMA fechado.

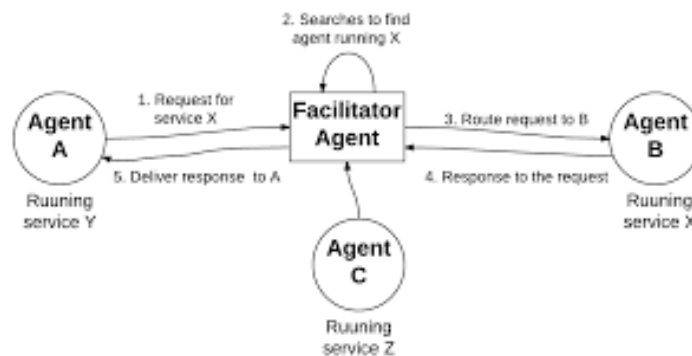
### 4.1 SMA centralizado

Num Sistema Multiagente centralizado há sempre um agente que acaba por tomar todas as decisões. Independentemente da quantidade de agentes individuais que possam existir no sistema, existe um agente responsável por tomar as decisões finais, permitindo, assim a comunicação indireta dos vários agentes, **Figura 18**.

Um exemplo de um Sistema Multiagente centralizado pode ser, por exemplo, uma central de comboios, em que há uma central, agente responsável pelas decisões, que efetua a marcação dos pedidos dos clientes nas viagens de comboio pretendidas e o próprio comboio onde essa viagem será feita. Portanto, temos os comboios e os clientes que podem ser tratados como agentes e a central que é o agente que permite a comunicação entre os dois.

As vantagens deste tipo de SMA é que permite uma melhor organização entre agentes, assegurando uma comunicação mais eficiente entre os mesmos. Menos erros são tomados e melhores serão os resultados finais, uma vez que existe uma melhor atribuição do papel a cada agente.

Por outro lado, um SMA centralizado traz algumas desvantagens, já que quando ocorrer algum problema ou bug no agente responsável pelas decisões, então todo o sistema é comprometido. Além disso, como tudo tem de passar pelo agente moderador, o processo poderá ser mais eficiente mas poderá ocorrer também de uma forma mais lenta.



**Figura 18:** SMA centralizado

## 4.2 SMA descentralizado

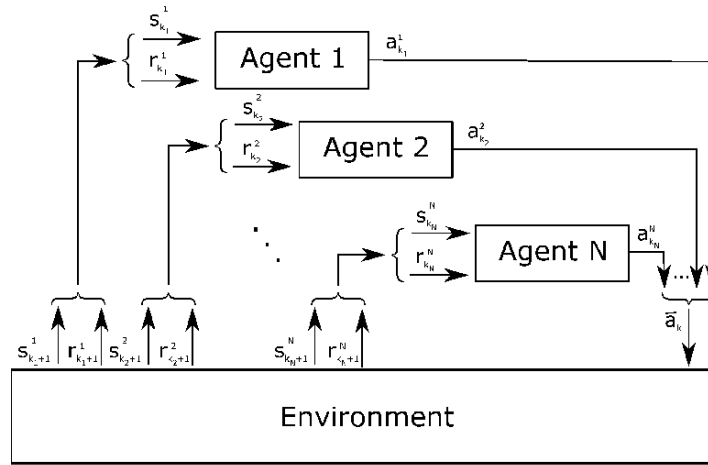
Num sistema descentralizado, ao contrário do anterior, cada agente apresenta a sua própria responsabilidade, tomando decisões de acordo com a sua interpretação do SMA em que se encontra. Isto é, não existe um agente moderador por onde todas as decisões deverão passar. Cada agente efetua a ação pelo qual foi programado e comunica diretamente com outros agentes, **Figura 19**.

Um exemplo de um SMA descentralizado poderá ser quando se tem 3 agentes que correspondem a três jogadores de futebol num videojogo. Cada um dos agentes comunica entre si e tomam decisões de acordo com o ambiente que o rodeia e de acordo com o que os outros agentes lhe dizem.

No que diz respeito a este tipo de sistema existem vantagens associadas, nomeadamente

o facto de permitir uma comunicação direta e rápida entre agentes, já que não é necessário esperar pela decisão do agente moderador. Além disso, permite também uma implementação mais rápida e não existe um ponto único que compromete o sistema por completo, visto que ao falhar um agente, os restantes continuam operacionais.

Por outro lado, também acarreta algumas desvantagens como, por exemplo: a organização entre os agentes poderá não ser a melhor, tal como a sua comunicação. Isto acontece, pois ao existir um agente moderador, a ligação entre os agentes do mesmo sistema pode ser mais confiável, uma vez que não se encontram todos a comunicarem-se ao mesmo tempo. Apesar da implementação ser talvez mais rápida a curto prazo, mais bugs poderão ocorrer mais tarde, comprometendo o projeto a longo prazo.



**Figura 19:** SMA descentralizado

### 4.3 SMA Aberto

Para um sistema ser considerado interoperável, é extremamente importante que trabalhe com padrões abertos, conseguindo, assim, comunicar de forma transparente com outro sistema [3].

Um Sistema Multiagente é, tipicamente, um sistema aberto, complexo e heterogêneo, no qual se realizam atividades coordenadas. A coordenação de atividades neste tipo de ambientes exige linguagens e sistemas de comunicação apropriados: a sintaxe padronizada, ou seja, normas de comunicação ao nível da estrutura das mensagens e da gestão dos canais de comunicação e a Semântica padronizada, isto é normas sobre o conteúdo das mensagens/ontologias partilhadas [3].

Um SMA Aberto não tem uma arquitetura prévia, apenas contém agentes no seu ambiente e, portanto, os agentes não têm, necessariamente, consciência da existência dos outros. Neste



SMA é exigido um mecanismo para identificar e localizar outros agentes e, por esse motivo, os agentes podem ser não-cooperativos [3].

Para a implementação deste tipo de SMA é necessário ter em conta as suas vantagens, bem como desvantagens de acordo com a aplicação pretendida.

**Vantagens:**

- Agentes ou grupos de agentes são concebidos separadamente;
- Flexíveis e tolerantes a falhas;
- Desenvolvimento incremental do sistema;
- Manutenção facilitada;
- Sociedade aberta e dinâmica.

**Desvantagens:**

- O comportamento geral do sistema não é previsível;
- Os protocolos, a linguagem, as ontologias, podem variar entre agentes;
- Podem surgir comportamentos maliciosos, difíceis de evitar;
- Não se adaptam facilmente às situações concretas.

#### **4.4 SMA Fechado**

Num Sistema Multiagente Fechado, ao contrário do anterior, existe uma arquitetura de desenho estático, com componentes e funcionalidades já pré-definidos [3].

Neste tipo de metodologia, as propriedades do próprio sistema são conhecidas antecipadamente como: a linguagem comum entre agentes e, portanto, cada agente pode ser desenvolvido como um expert. Além disso, os agentes são normalmente cooperativos e podem trabalhar em prol do desenvolvimento do sistema.

**Vantagens:**

- Carga e perícia distribuídas;
- Simplicidade e previsibilidade;

- Os componentes são conhecidos;
- Os agentes normalmente ostentam comportamentos cooperativos;
- Os agentes partilham a arquitetura e o software.

**Desvantagens:**

- Custos de manutenção podem ser elevados;
- Pode ser menos tolerante a falhas;
- Difícil a interoperabilidade com outros sistemas.

## 5 Sistemas Multiagentes e IA

A Inteligência Artificial começou a ser desenvolvida há alguns anos atrás e continua a ser um tema altamente relevante e utilizado nas diferentes áreas do saber. Atualmente, existem sistemas específicos dotados de "inteligência" que são capazes de competir com humanos em diversas áreas. Apesar do recurso e manipulação da IA, é também um termo que provoca diferentes interpretações entre pessoas, à custa de divergências entre os vários investigadores.

Diversos autores tentaram estabelecer um conceito que poderia expressar o que é a inteligência, mas qualquer um destes conceitos é apenas uma mera caricatura da realidade ou apenas expressa algumas características e manifestações da inteligência [9].

IA é também definida como um ramo da ciência da computação que permite o estudo de como os computadores conseguem realizar coisas, as quais poderão ser no futuro superiores às coisas que os humanos realizam [10].

Apenas recentemente, com o surgimento do computador moderno, é que a inteligência artificial ganhou meios para se estabelecer como ciência integral, com problemáticas e metodologias próprias. Desde então, seu desenvolvimento tem extrapolado os clássicos programas de xadrez ou de conversão e envolvido áreas como visão computacional, análise e síntese da voz, lógica difusa, redes neuronais artificiais e muitas outras [9].

Entre os diferentes teóricos ou investigadores que estudam o que é possível efetuar com a IA existe uma discussão em aberto sobre onde se consideram duas propostas básicas: IA "forte" e IA "fraca". A investigação em Inteligência Artificial Forte aborda a criação da forma de inteligência baseada em computador que consiga raciocinar e resolver problemas; uma forma de IA forte é classificada como auto-consciente. A inteligência artificial fraca centra a sua investigação na criação de inteligência artificial que não é capaz de verdadeiramente raciocinar e resolver problemas. Uma tal máquina com esta característica de inteligência agiria como se fosse inteligente, mas não tem autoconsciência ou noção de si. Um exemplo que se tornou no alicerce do tema de IA é o teste de Turing, em que se fazia perguntas a uma pessoa e a um computador escondidos, no qual era impossível a alguém distinguir qual interlocutor é a máquina e qual é a pessoa.

### 5.1 Inteligência Artificial Distribuída

A Inteligência Artificial Distribuída (IAD), com um enfoque diferente da Inteligência Artificial tradicional, tenta dividir um problema em problemas pequenos e mais simples. Eric Bonabeau, Marco Dorigo e Guy Theraulaz discutem sobre a metáfora dos insetos sociais, ou

seja, um conjunto de agentes simples, flexíveis e robustos, interagindo num ambiente próprio e entre si [4]. Um exemplo relevante para esta definição é o conjunto de formigas que interagem umas com as outras para saber qual o caminho mais curto de um ponto A para um ponto B.

Uma definição bastante esclarecedora de IAD é a de [3]: “Diz-se que um sistema é de Inteligência Artificial Distribuída se seu funcionamento depende de um determinado conjunto de partes (não de um único elemento) para resolver de modo cooperativo um determinado problema.

Uma das principais motivações para o uso de IAD é precisamente pelo facto de os seres vivos se agruparem de forma a encontrarem uma forma realizar uma tarefa em conjunto ou até mesmo como forma de sobrevivência. Seres humanos formam grupos, organizações, sociedades, nações, distribuídos e de forma hierárquica, tal como as abelhas que formam uma sociedade organizada e com especializações. Outra motivação também pode ser a busca de soluções para problemas complexos que podem ser decompostos em problemas simples.

Com o recurso à Inteligência Artificial Distribuída, surge a necessidade da utilização de agentes, que é precisamente o tema do presente relatório. Esta estuda o conhecimento e os métodos de raciocínio que podem ser necessários ou úteis para que estes agentes computacionais participem de uma sociedade de agentes.

A IAD, **Figura 20**, encontra-se dividida em dois grupos [3]:

- Solução Distribuída de Problemas (SDP): agentes são desenvolvidos para cooperar, dividir as tarefas e comunicar-se de maneira confiável;
- Sistemas Multiagentes (SMA): estudo das pressuposições básicas sobre agentes que garantam a possibilidade de ação cooperativa em sociedade, ou seja, o foco é o agente.



**Figura 20:** Inteligência artificial distribuída

## 5.2 Conferências e Palestras sobre IAD e SMA

Inteligência Artificial Distribuída e, por sua vez, Sistemas Multiagentes têm vindo a tornar-se uma forte componente de estudo e discussão no mundo da ciência, em especial na ciência da computação. Dessa forma, ocorreram e continuam a existir conferências onde especialistas de todo o mundo juntam-se para dialogar sobre este tema e como este tem tido um forte impacto nas suas investigações e/ou pesquisas, **Figura 21**.

Exemplos de conferências [11]:

- ISMSI 2021 — 5th International Conference on Intelligent Systems, Metaheuristics Swarm Intelligence 10 Apr 2021 - 11 Apr 2021 • Victoria, Seychelles
- The 21st European Conference on Evolutionary Computation in Combinatorial Optimisation;
- AAMAS 2021 — 20th International Conference on Autonomous Agents and Multiagent Systems 03 May 2021 - 07 May 2021 • Virtual, United Kingdom;
- ICECMSN 2021 — 2nd Scopus-Indexed Springer International Conference On Evolutionary Computing And Mobile Sustainable Networks;
- ADAITC 2021 — 2021 IEEE Autonomous Driving AI Test Challenge 23 Aug 2021 - 26 Aug 2021 • San Jose, United States;
- ICMLAS 2021 — International Conference on Machine Learning and Autonomous Systems 24 Sep 2021 - 25 Sep 2021 • Palkulam, Kanyakumari District - 629401. TamilNadu, India.



**Figura 21:** ISMSI 2021 — 5th International Conference on Intelligent Systems

## 6 Análise crítica de aplicações/projetos existentes e plataformas disponíveis

Os sistemas informáticos cada vez mais estão a analisar quantidades enormes de dados e tomam decisões nas vidas dos seres humanos tendo por base o conhecimento retido por esses mesmos dados. A grande questão reside no facto de a modelação e a computação estarem-se a tornar cada vez mais complexas e será crucial desenvolver software suficientemente robusto para lidar com estas problemáticas. Por esta razão, os Sistemas Multiagentes são necessários para resolver problemas, por exemplo, na área industrial como o controlo de processos e o controlo/simulação de sistemas de controlo de Tráfego Aéreo ou urbano com a implementação de carros autónomos, no ensino como é o caso das plataformas e-learning. Além disso, é de extrema importância o seu uso na área da medicina, especialmente para questões de diagnóstico de doenças, monitorização de pacientes, cuidados de saúde, gestão hospitalar, entre outros, [8].

Isabel Fernández, especialista de inteligência artificial da Accenture afirma que "como sociedade, não precisamos de perceber a inteligência artificial, mas precisamos de confiar nela". É de salientar que nova tecnologia poderá trazer assistentes virtuais inteligentes, carros sem condutor e muitos avanços na saúde. Contudo, à custa da inovação, muitos empregos poderão desaparecer. Isabel Fernández salienta que "já aconteceu no passado e conseguimos resolvê-lo" [12].

A utilização de sistemas Multiagentes tem vindo a ter impactos positivos na vida quotidiana. É de destacar que na Península Ibérica, mais concretamente em Madrid, Espanha, existe um sistema responsável pela temperatura dentro das estações do metro de Madrid que é um Sistema de Multiagentes puro, o qual realiza previsões sobre o número de pessoas, a temperatura no exterior e a temperatura interior, que depende do número de carruagens que estão a chegar à estação [12]. Este sistema também avalia o preço de mercado da electricidade. Através destes modelos, o SMA faz a gestão do ar condicionado em tempo real e escolhe automaticamente a temperatura, dentro de um intervalo muito pequeno, para tentar minimizar os custos de energia. Com este recurso é possível poupar 26% os custos de electricidade. Outro exemplo do impacto de SMAs mais precisamente na medicina é na situação de quando alguém vai ao médico, existe um sistema que permite analisar em tempo real as notas que ele está a tirar e que lhe dá avisos sobre possíveis diagnósticos. No entanto, fica sempre a cargo do médico seguir as recomendações do sistema. Este último ponto é importantíssimo, uma vez que mostra que a utilização de SMAs não tem como objectivo substituir um ser humano, mas

auxiliá-lo, sendo que o poder e a vontade de querer essa ajuda reside no humano.

Para além da medicina e indústria, os SMAs podem também ser usados em videojogos. Vários projetos desenvolvidos por investigadores na universidade de Flórida como [13]:

- **RoboViz** é um programa de software projetado para avaliar e desenvolver comportamentos de agentes num Sistema Multiagente, que é uma liga de futebol simulada designada por RoboCup 3D. RoboViz é um monitor interativo que consegue renderizar informações do agente numa perspetiva tridimensional. Além disso, o RoboViz fornece um desenho programável para agentes no qual se podem comunicar numa rede, **Figura 22**;



**Figura 22:** RoboViz

- Outra aplicação é o **Motor móvel RoboCanes**. Um novo motor de caminhada adaptativa para robôs bípedes. É uma caminhada em malha fechada que se adapta às diferenças no comportamento físico do robô otimizando desta forma os parâmetros do modelo enquanto caminha e executa outras tarefas. Desta maneira uma quantidade significativa de erros nas previsões do modelo pode ser reduzida, **Figura 23**;



**Figura 23:** RoboCanes

- **MoCap.** O laboratório da Universidade de Flórida, USA é um laboratório de captura de movimento que se baseia em marcadores passivos com material retrorreflexivo. Existem 10 câmaras de 4 megapixels (120 Hz) e 4 plataformas de força para medir as forças no solo. Também existem instalações para capturar expressões faciais e dispositivos eletromagnéticos para capturar voltagem em 16 grupos de músculos para estudos biomédicos. Para se conseguir elaborar este sistema foi necessário fazer experiências com movimentos de desporto complexos, por exemplo os movimentos realizados no futebol. Os dados 3D são então depois transferidos para um jogador de futebol animado em 3D num videojogo, **Figura 24**.



**Figura 24:** MoCap



## 7 Aplicabilidade no domínio da Engenharia Biomédica

Sistemas Multiagentes possuem uma forte presença na área médica, já que tentam construir um ser inteligente ou intelectual semelhante ao humano, ou seja, pretende-se que seja capaz de aprender e realizar intervenções práticas.

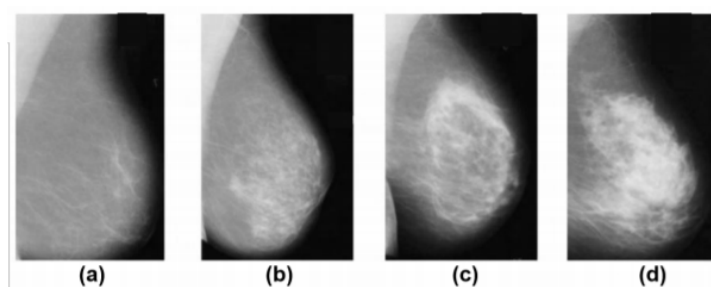
Aplicações de SMAs no domínio da saúde:

### 7.1 Reforço No Diagnóstico

Um sistema computacional pode analisar a foto de uma mancha na pele, comparando-a com a sua base de dados e dessa forma estabelecer as probabilidades de ser um melanoma. A partir disso, o planeamento de tratamentos pode ser considerado como outro ponto forte dos Sistemas Multiagentes que, ao se basear nas informações específicas de um paciente pode permitir o tratamento de milhares de casos semelhantes [14].

Aplicações similares estão sendo desenvolvidas para muitas outras doenças, nomeadamente no cancro da mama, que é precisamente a forma mais mortal de cancro entre mulheres. O padrão na prática clínica para apoio ao diagnóstico do cancro de mama por imagem ainda é a mamografia por raios-x, **Figura 25**. No entanto, essa técnica tem como desvantagens, além do custo, o uso de radiações ionizantes, que por sua vez podem ainda estar relacionados a fatores causadores de cancro, e a impossibilidade de utilização em pacientes mais jovens. Sistemas Inteligentes, baseados na identificação de padrões em imagens com lesões de mama, aliados à combinação entre descritores de forma e de textura e à decomposição em série de Wavelets têm se mostrado bastante eficientes na automatização da análise mamográfica. Esta utilização permite, assim, melhorar tanto o diagnóstico como a fase de tratamento posterior possibilitando melhores exames e prescrições ao paciente [1].

Figura 13 – Mamogramas dos diferentes tecidos mamários: (a) adiposo (tipo I), (b) fibroglandular (tipo II), (c) heterogeneamente denso (tipo III) e (d) extremamente denso (tipo IV).



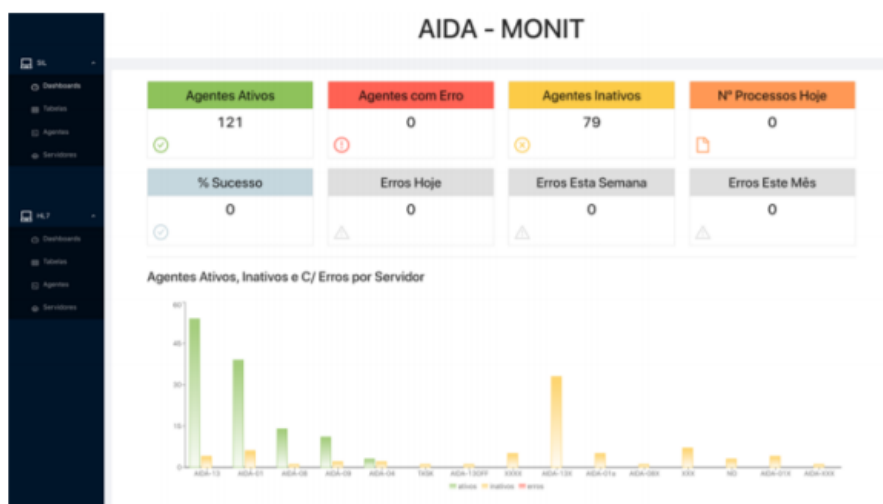
**Figura 25:** Imagens de mamografias retiradas de [1]

## 7.2 Alertas E Notificações

Primeiramente, há recursos baseados em SMAs que podem alertar os médicos, em tempo real e por meio do envio de alertas ou mensagens automatizadas, acerca de quaisquer alterações importantes no quadro clínico de seus pacientes.

Para fortalecer e tornar ainda mais prático o relacionamento entre pacientes e médicos, é possível, também, que os hospitais e clínicas tenham sistemas para a análise de exames e prescrição de medicamentos. A esta conexão designa-se por interoperabilidade, que pode ser com alguma facilidade implementada por Sistemas de Multiagentes [14].

A plataforma AIDA foi desenvolvida pelo grupo de Inteligência Artificial da Universidade do Minho e está de acordo com os princípios e funcionalidades intrínsecas de agentes e sistemas de agentes, **Figura 26**. AIDA é um sistema complexo que consiste em especialistas e agentes inteligentes diretos, que buscam a integração, disseminação e arquivo de grandes volumes de dados de fontes heterogêneas (por exemplo, computadores e dispositivos médicos) e a uniformidade dos sistemas clínicos. AIDA usa uma arquitetura multiagente do tipo orientada a serviços (SOA) para garantir a interoperabilidade entre vários sistemas de informação. Para conseguir este feito, no sistema os diferentes agentes que poderão estar implementados em diferentes equipamentos conseguem enviar mensagens uns aos outros de forma a efetuar as ações disponibilizadas pelo sistema AIDA, permitindo que os hospitais e clínicas tenham sistemas mais eficazes para a análise de exames e prescrição de medicamentos [15].



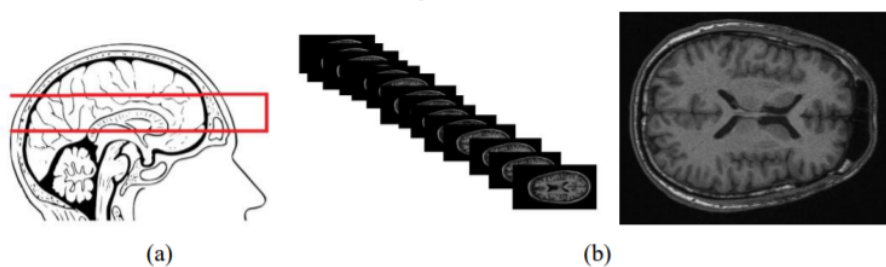
**Figura 26:** Sistema AIDA

### 7.3 Modelos Algorítmicos na Medicina Preventiva

As implementações de SMAs na medicina, partindo de padrões de referência e dados de pacientes, podem criar modelos algorítmicos destinados a qualificar informações, prever a evolução de pacientes e suas necessidades. A utilização de modelos algorítmicos antecipa as probabilidades de que algo aconteça e analisa as suas possíveis complicações, permitindo intervenções adequadas. De fato, a análise preditiva aplicada às informações de saúde tem um imenso potencial para salvar vidas, [14].

Uma das aplicações será a previsão da Doença de Alzheimer (DA). O seu interesse está focado, cada vez mais, no seu diagnóstico preciso a fim de iniciar de forma mais precoce os tratamentos possíveis na tentativa de retardar a progressão da doença. Para que tal aconteça é necessário a utilização técnicas de Machine Learning que se têm mostrado bastante adequadas para construir ferramentas automatizadas de apoio ao diagnóstico por imagem de MRI (Magnetic Resonance Imaging), **Figura 27**, além de ter um baixo custo computacional para o treino. Esta técnica envolve o recurso a redes neurais que vão ser treinadas através de um determinado conjunto de datasets, melhorando diferentes hiper-parâmetros para que no período de teste, aquilo que o modelo prevê seja o mais próximo possível do que foi reservado para teste. Desta forma poderemos ter agentes, neste caso software capaz de aprender e tomar decisões por vontade própria.

Figura 08 - Exemplo das 15 fatias selecionadas e de um corte transversal, oriundas de um volume de imagem, escolhido de forma aleatória, da base de dados MIRIAD. Juntamente com um exemplo de corte transversal aproximado.



**Figura 27:** Imagens de MRI adquiridas de [2]

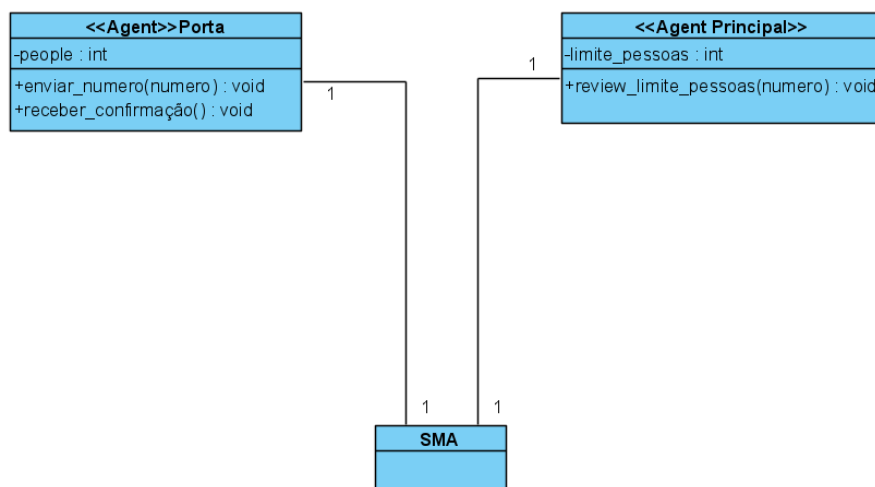
## 8 Ideias para projetos futuros sobre Sistemas Multiagentes

### 8.1 Agentes para monitorizar as filas de espera nas instituições de saúde

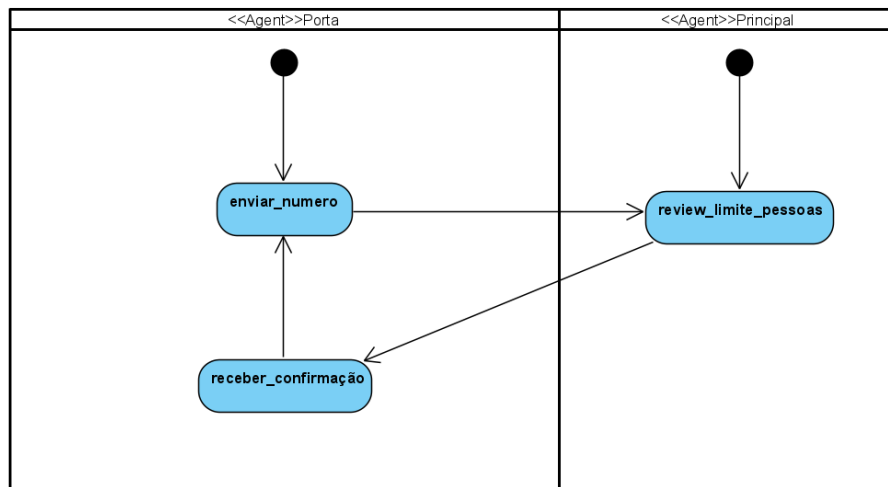
Atualmente, os cuidados de saúde têm um papel fundamental na vida da população em geral. Todo o cuidado com a saúde é pouco e, por esse motivo, as instituições como hospitais e clínicas encontram-se cheias de profissionais de saúde (médicos e enfermeiros) e de funcionários necessários para um bom e eficiente funcionamento. No entanto, a utilização de Sistemas Multiagentes pode auxiliar a monitorizar o tempo de espera na zona da receção dos hospitais e/ou clínicas tornando, desta maneira, o funcionamento ainda mais eficaz.

Desta forma, foi considerado um sistema Multiagente, cuja elaboração e reflexão está ilustrada nas **Figuras 28 e 29**, as quais foram realizadas com recurso ao software "visual paradigm".

No sistema referido existem 2 agentes: o agente "Porta" e o agente "Principal". O agente "Porta" corresponde a um agente implementado na zona da porta principal da instituição que através de sensores recebe a informação de que uma ou mais pessoas entraram. Por essa razão, apresenta uma variável que vai sendo incrementada consoante o número de pessoas que entram na instituição e que vai decrementando com a situação contrária. Sempre que a variável é incrementada, este agente envia uma mensagem do tipo "proposal" ao agente Principal, que corresponde ao sistema geral do hospital. O agente "Principal", ao receber a mensagem, verifica se o número que recebeu, o qual corresponde ao número de pessoas que se encontram na instituição, é ou não superior a um valor que é o valor máximo de pessoas a partir do qual é necessário informar os funcionários da receção que deverão acelerar o processo.



**Figura 28:** Class diagram do projeto

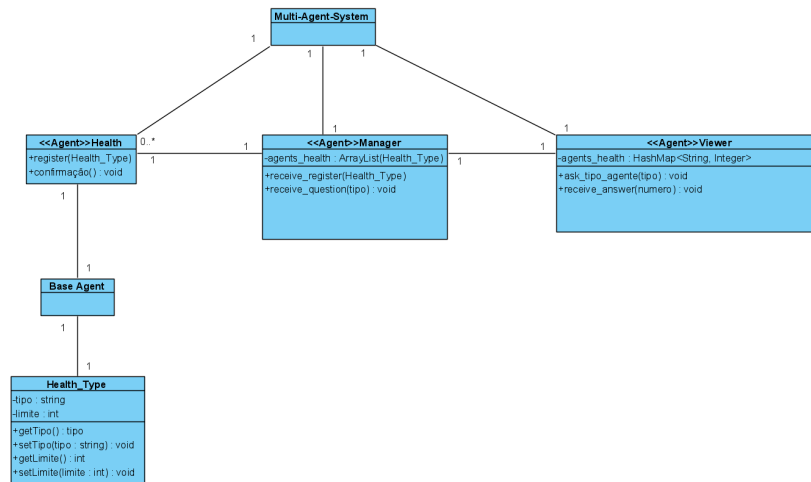


**Figura 29:** Activity Diagram do projeto

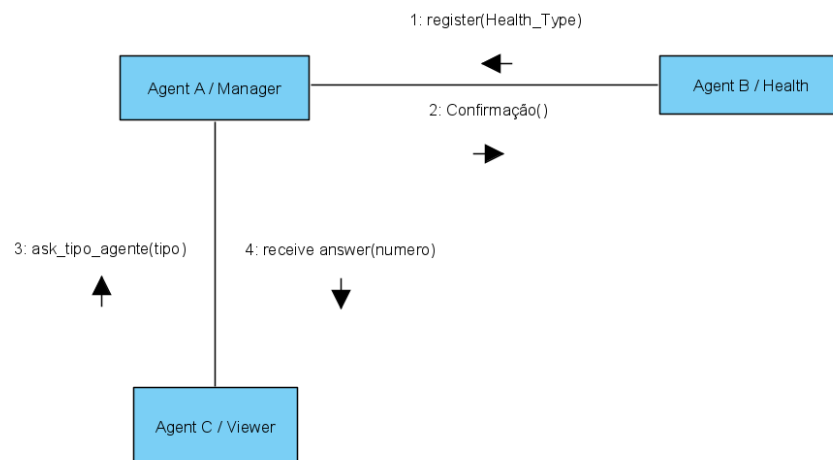
## 8.2 Agentes para monitorizar as condições de um utilizador

Além da aplicação anterior é também possível implementar uma outra que possa agora monitorizar as condições de saúde de um determinado utilizador, nomeadamente a sua pressão arterial, o seu nível de oxigénio no sangue, a sua temperatura corporal, etc...

Com este objetivo, foi, então, pensado num Sistema Multiagente que pudesse solucionar o problema. Neste sistema temos 3 agentes: o agente "Health", o agente "Manager" e o agente "Viewer". Além disto, também se tem uma classe que contém duas variáveis. A primeira corresponde ao tipo de condição de saúde que poderá ser um das mencionadas anteriormente e um número associado ao limite máximo de umas dessas condições de saúde. Como se ilustra nas **Figuras 30 e 31**, o agente "Health" regista-se primeiro enviando uma mensagem do tipo "subscribe" ao agente "Manager", dizendo qual é o tipo de condição a qual é responsável e o número máximo limite associado. De seguida, o agente "Manager" responde, de forma a efetuar a respetiva confirmação. Depois, surge o agente "Viewer" que, através de sensores colocados num dispositivo que possa medir os valores associados às condições de saúde de um paciente no presente, envia uma mensagem do tipo request ao agente "Manager" a pedir qual o número máximo limite associado ao tipo de condição que acabou de receber dados. Por fim, o agente "Manager" responde enviando esse número e o agente "Viewer" verifica se esse valor é maior ou não do que aquele que recebeu dos sensores. Se for, então mostra que não há perigo, caso contrário mostra que há perigo.



**Figura 30:** Class diagram do projeto



**Figura 31:** Activity Diagram do projeto

É de salientar que estes agentes, tanto na situação 1 como na 2, são agentes reativos, uma vez que recebem dados através de sensores e reagem consoante esses dados sejam maiores ou menores do que aqueles considerados como limites. Além disso, na primeira situação temos um SMA descentralizado, já que não existe um agente no qual todas as decisões deverão passar. Por outro lado, na segunda situação tem-se um SMA centralizado pela existência do agent "Manager" que toma as decisões e realiza a comunicação indireta entre os outros dois agentes.

## 9 Sistemas Multiagentes no futuro e questões éticas

Sistemas Multiagentes não têm os seus dias contados muito pelo contrário. O desenvolvimento nesta área começou logo a seguir à Segunda Guerra Mundial à custa do matemático inglês Alan Turing com o artigo "Computing Machinery and Intelligence". Desde então, SMAs tentam reproduzir o pensamento e intelectualidade do ser humano.

Existe uma ideia errada do que o futuro reserva relativamente aos SMAs e torna-se cada vez mais importante compreender que a existência dos SMAs reside no facto de puderem auxiliar o humano e não substituí-lo por completo. Ideias erradas são, por exemplo, pensar que SMAs ou Inteligência Artificial pode funcionar como o cérebro humano. Tal ideia é errada pelo menos por enquanto, uma vez que apesar de conseguirem realizar determinadas tarefas ou resolver problemas de uma forma mais rápida e eficiente, ainda estão longe de simular por completo o cérebro humano, sendo, por isso capazes de serem melhorados no futuro.

Além disso, uma outra ideia é que SMAs poderão eliminar todos os cargos de trabalho. Uma ideia que apesar de incorreta, apresenta alguma verdade, visto que muitos dos trabalhos desenvolvidos neste momento poderão ser automatizados. No entanto, existem muitos outros em que características como simpatia, emoção e empatia são cruciais e que SMAs ainda não têm a capacidade de possuir. Por esse motivo, muitos trabalhos continuarão a ser como são.

De salientar também a ideia que SMAs e IA poderão mudar por completo o mundo em muito pouco tempo. Nos dias de hoje, existem, de facto, muitas tarefas repetitivas que podem ser realizadas de forma automatizada, pelo simples facto que "robôs" não possuem características como cansaço ou despesas a não ser de manutenção. Contudo este tipo de tecnologia está numa fase de investigação em massa mas que é gradual, sendo que só se poderá pensar numa mudança drástica ou num impacto significativo para a sociedade num futuro não muito recente.

Por fim, os Sistemas Multiagentes foram e continuarão a ser utilizados em grande maioria na área da investigação, produzindo resultados com o objetivo de melhorar determinados aspetos da vida social das pessoas, noemadamente na área da indústria, como na condução autónoma ou na utilização para melhoria nos sistemas de distribuição como, por exemplo, os drones da Amazon. Também na área da saúde permitindo o desenvolvimento de algoritmos no âmbito da IA, Machine Learning, para encontrar padrões que dêem informação relevante sobre a classificação ou previsão de determinadas doenças, na área dos videojogos para melhorar a experiência de jogo, colocando jogadores online a se defrontarem com outros do mesmo rank, etc. [16].

## 10 Conclusão

Os Sistemas Multiagentes, agentes e a sua respetiva definição encontram-se longe de ser consensual. Contudo, tal pode ser visto como uma vantagem ou desvantagem, dependendo do problema que se pretende resolver recorrendo a estes sistemas.

Agentes apresentam diferentes definições como inúmeras características. Dependendo da problemática, diferentes características podem ser usadas em detrimento de outras por serem mais adequadas à resolução do problema em questão.

Diferentes agentes podem comunicar-se entre si formando o que se designa por Sistemas Multiagentes, sendo que essa comunicação poderá ser feita por várias vias. As mensagens trocadas uns com os outros seguem normas que devem ser respeitadas de forma a se conseguir uma comunicação eficiente sem perda de informação. Existem diferentes metodologias de software referidas ao longo do artigo, que permitem este tipo de comunicação e implementação de SMAs.

Os SMAs podem ser utilizados em diversas áreas de aplicação, as quais são mencionadas no decorrer do artigo. Os Sistemas Multiagentes possuem vantagens e desvantagens que deverão ser tomadas em conta no momento da sua implementação.

A elaboração de Sistemas Multiagentes poderá variar de problema para problema, no entanto pela existência de software próprio para este efeito, torna toda a fase de processamento e desenvolvimento mais rápida e eficaz para um determinado investigador da área.

Por fim, o relacionamento de SMAs com Inteligência Artificial será de uma enorme importância nos próximos anos e que vai, sem espaço para dúvidas, marcar a história da Humanidade para sempre.



## Referências

- [1] MAÍRA ARAÚJO DE SANTANA, “Sistemas inteligentes para apoio ao diagnóstico do câncer de mama usando imagens mamográficas e termográficas,” M.S. thesis, UNIVERSIDADE FEDERAL DE PERNAMBUCO CENTRO DE TECNOLOGIA E GEOCIÊNCIAS DEPARTAMENTO DE ENGENHARIA BIOMÉDICA, Recife, Brasil, 2020.
- [2] GABRIELA DOS SANTOS LUCAS E SILVA, “Sistemas inteligentes para diagnóstico não invasivo da doença de alzheimer usando imagens mri,” M.S. thesis, UNIVERSIDADE FEDERAL DE PERNAMBUCO CENTRO DE TECNOLOGIA E GEOCIÊNCIAS ESCOLA DE ENGENHARIA DE PERNAMBUCO, Recife, Brasil, 2020.
- [3] “Agentes inteligentes,” class notes for Intelligent Systems, Universidade do Minho Departamento de Informática, Braga, Portugal, Setembro de 2006.
- [4] M. Wooldridge, “An introduction to multiagent systems,” University of Liverpool, UK. [Online]. [Online]. Available: <http://dl.icdst.org/pdfs/files/417f5e65f9fbd30413f6b983df8f4175.pdf>
- [5] Heecheol Jeon Charles Petrie Mark R. Cutkosky. "jatlite: A java agent infrastructure with message routing,"mar. 25, 2021. [Online]. Available: <http://www-cdr.stanford.edu/ProcessLink/papers/jat/jat.html>
- [6] Foundation for Intelligent Physical Agents. "agent communication language specifications,"mar. 25, 2021. [Online]. Available: <http://www.fipa.org/repository/aclspecs.html>
- [7] University Of Montreal. "tutorial 3. jade inter platform communication,"mar. 25, 2021. [Online]. Available: <http://www.iro.umontreal.ca/~dift6802/jade/doc/tutorials/JADEAdmin/JadePlatformTutorial.html>
- [8] Jing Xie Chen-Ching Liu. "multi-agent systems and their applications,"mar. 26, 2021. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/22348972.2017.1348890>
- [9] “Inteligência artificial distribuída,” *Wikipedia*. Accessed Mar. 22, 2021. [Online]. [Online]. Available: [https://pt.wikipedia.org/wiki/Inteligência\\_artificial\\_distribuída](https://pt.wikipedia.org/wiki/Inteligência_artificial_distribuída)
- [10] “Inteligência artificial,” *Wikipedia*. Accessed Mar. 24, 2021. [Online]. [Online]. Available: [https://pt.wikipedia.org/wiki/Inteligência\\_artificial](https://pt.wikipedia.org/wiki/Inteligência_artificial)

- [11] COMS. "conferences and meetings on software agents and intelligent agents in artificial intelligence," mar. 25, 2021. [Online]. Available: <https://www.conference-service.com/conferences/agents.html>
- [12] J. P. Pereira, "Inteligência artificial," PUBLICO, 2 de Novembro, 2018. [Online]. [Online]. Available: <https://www.publico.pt/2018/11/02/tecnologia/noticia/maioria-construir-bem-sistemas-inteligentes-1849482>
- [13] Florida International University. "multi-agent systems," mar. 26, 2021. [Online]. Available: <https://www.cs.miami.edu/home/visser/multi-agent-systems.html>
- [14] T. E. em Telerradiologia, "Inteligência artificial: 5 maiores aplicações na medicina," Web, 24 de Fevereiro, 2021. [Online]. [Online]. Available: <https://teleimagem.com.br/inteligencia-artificial-5-maiores-aplicacoes-da-inteligencia-artificial-na-medicina/>
- [15] Regina Sousa , Diana Ferreira , António Abelha, José Machado, "Step towards monitoring intelligent agents in healthcare information systems," *ALGORITMI Research Center, School of Engineering, University of Minho, Gualtar Campus, 4710 - 057 Braga, Portugal*, vol. 1, no. 1, March 2021. [Online]. doi: [http://repositorium.sdum.uminho.pt/bitstream/1822/65910/1/Submission15<sub>worldCIST20</sub>.pdf](http://repositorium.sdum.uminho.pt/bitstream/1822/65910/1/Submission15_worldCIST20.pdf).
- [16] K. Highnam, "The future of multi-agent systems research," Web, Mar. 24, 2020. [Online]. Available: <https://medium.com/@kwhighnam/the-future-of-multi-agent-systems-research-5aef163f1d69>