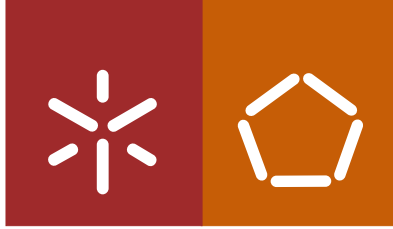


Universidade do Minho
Escola de Engenharia

Luciana Almeida Cardoso

Desenvolvimento de uma Plataforma baseada em Agentes para a Interoperabilidade



Universidade do Minho

Escola de Engenharia

Luciana Almeida Cardoso

Desenvolvimento de uma Plataforma baseada em Agentes para a Interoperabilidade

Dissertação de Mestrado em Informática Médica
Mestrado Integrado em Engenharia Biomédica

Trabalho efetuado sob orientação do
Professor Doutor António Carlos da Silva Abelha
e coorientação do
Mestre Carlos Filipe da Silva Portela

outubro de 2013

Declaração

Nome: Luciana Almeida Cardoso

Endereço eletrónico: lucianacardoso05@gmail.com

Cartão de Cidadão: 13752637

Título da Dissertação: Desenvolvimento de uma Plataforma baseada em Agentes para a Interoperabilidade

Orientador: Professor Doutor António Carlos da Silva Abelha

Coorientador: Mestre Carlos Filipe da Silva Portela

Ano de conclusão: 2013

Designação do Mestrado: Mestrado Integrado em Engenharia Biomédica

Área de Especialização: Ramo de Informática Médica

DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA
A REPRODUÇÃO DE QUALQUER PARTE DESTA DISSERTAÇÃO.

Universidade do Minho, __ / __ / __

Assinatura: _____

Agradecimentos

Gostaria de agradecer em primeiro lugar ao meu orientador, professor doutor António Abelha e ao meu coorientador, mestre Filipe Portela, pela disponibilidade prestada e pela ajuda na elaboração deste projeto.

Não menos importante, quero prestar agradecimento ao professor doutor José Machado pelo tempo e pela paciência disponibilizada. Agradecer todas as suas sugestões que se revelaram bastante importantes para este trabalho, e especialmente, pela sua boa disposição natural.

A inserção de um sistema num ambiente real é um etapa importante no desenvolvimento de *software*, dessa forma gostaria de agradecer também ao Centro Hospitalar do Porto pela oportunidade concedida, a possibilidade de testar e implementar o sistema desenvolvido nas suas instalações com recurso aos seus sistemas informáticos.

Gostaria também de agradecer aos meus amigos de longa data e àqueles que fiz durante o meu percurso académico, que me acolheram numa cidade desconhecida, que me fizeram rir nos momentos mais felizes e me aconchegaram nos mais tristes, um grande obrigada por fazerem parte da minha vida, da minha família, do meu coração.

Quero agradecer também à minha família, aos meus pais que sempre me apoiaram nas minhas decisões mais importantes, e que fizeram tudo para me proporcionar uma boa educação e formação cívica. Em especial quero agradecer à minha irmã com quem tenho partilhado toda a minha vida.

Por fim, e de um modo muito especial quero agradecer ao Fernando por toda a paciência que tem para me aturar e essencialmente pelos bons e maus momentos passados ao longo destes anos. Obrigada por tudo.

Resumo

Nas últimas décadas e com os avanços tecnológicos, a quantidade de dados e de sistemas de informação existentes nas unidades de saúde têm vindo a aumentar exponencialmente. A acessibilidade e disponibilidade da informação clínica dos pacientes é uma necessidade constante neste tipo de ambiente. A *Agência para a Integração, Difusão e Arquivo de Informação Médica (AIDA)* foi então desenvolvida com o intuito de contornar esta necessidade, e encontra-se implementada no *Centro Hospitalar do Porto (CHP)* há alguns anos, revelando um elevado sucesso, garantindo a interoperabilidade entre os sistemas de informação hospitalares lá existentes. A *AIDA* é um sistema baseado em agente que tem demonstrado ao longo do tempo a necessidade de um maior controlo sobre essas entidades e sobre a atividade que executam.

A construção e implementação de uma plataforma que integre todos os agentes e que controle a sua atividade é o principal objetivo deste projeto. Dessa forma foi criada a *Biomedical Multiagent Platform for Interoperability (BMaPI)* que facilita ao utilizador a criação de novos agentes localmente ou remotamente, permite ativar e desativar determinados serviços, simplifica a programação e reprogramação da atividade do agente e ainda possibilita ao utilizador monitorizar as atividades do agente em tempo real.

A *BMaPI* implementada no *CHP* revelou fornecer ao utilizador e consequentemente ao hospital, as funcionalidades e os dados necessários para que seja feita uma monitorização e gestão completa das atividades de cada agente e desse modo contribuir para um ambiente mais interoperável. Foi considerado ainda, um sistema eficiente, na medida em que não compromete os recursos da máquina onde foi implementado. Para além desses contributos a *BMaPI* quando integrada na *AIDA* eleva a sua funcionalidade e usabilidade global, pois fornece novas funcionalidades que revelam ser importantes para o ambiente, para o auxílio ao utilizador e constitui um sistema simples e de fácil compreensão.

Abstract

In the last decades, with technological advances, the amount of data and the information systems in health units has been increasing exponentially. The accessibility and availability of patient clinical information are a constant need in this type of environment. The AIDA was then developed in order to circumvent this need and it was implemented at the CHP a few years ago, revealing a highly successful, ensuring interoperability among CHP's healthcare information systems. AIDA is an agent based system that has demonstrated over time the need for greater control over these entities and on the activity they perform.

The construction and implementation of a platform that integrates all the agents and control their activity is the main objective of this project. Thus was created the BMaPI that facilitates the user to create new agents locally or remotely, it allows to enable and disable certain services, it simplifies the scheduling and rescheduling of agent activity and it enables the user to monitor the activities agent in real time.

The BMaPI implemented in CHP revealed provide the user the functionalities and the necessary data for it to make a complete monitoring and management of the activities of each agent and thereby contribute to a more interoperable environment. It was still considered an efficient system, since it does not compromise the resources of the machine where it was implemented. In addition to these contributions, the BMaPI when integrated in AIDA increases its functionality and overall usability, because it provides new features that prove to be important for the environment, for the aid to the user and represents a simple system and easy to understand.

Conteúdo

Resumo	v
<i>Abstract</i>	vii
Acrónimos	xx
1 Introdução	1
1.1 Contextualização e Enquadramento	1
1.2 Motivação	4
1.3 Objetivos	5
1.4 Estrutura	6
2 Estado da Arte	11
2.1 Sistemas de Informação Hospitalar	11
2.2 <i>Electronic Health Records</i>	13
2.3 Interoperabilidade	17
2.3.1 <i>Standards</i> para a Interoperabilidade	20
2.3.2 Princípios da Interoperabilidade	25
2.4 Agentes Inteligentes para a Interoperabilidade	29
2.5 Sistemas baseados em Agentes para a Interoperabilidade	31
3 Materiais e Métodos	33
3.1 Agentes Inteligentes	34
3.2 .NET	42
3.2.1 <i>System.Data.OracleClient</i>	44
3.2.2 <i>System.Net</i>	44

3.2.3	<i>System.Threading</i>	45
3.2.4	<i>System.Xml</i>	45
3.2.5	<i>System.Diagnostics</i>	45
3.2.6	<i>System.Timers</i>	45
3.3	<i>Pentaho Community</i>	46
4	Plataforma baseada em Agentes para a Interoperabilidade	47
4.1	Plataforma AIDA	47
4.2	Arquitetura e Implementação	53
4.2.1	Descrição dos componentes	56
4.2.2	Implementação do Protótipo em Ambiente Real	70
5	Apresentação e Discussão dos Resultados	73
5.1	Monitorização da Atividade dos Agentes	73
5.1.1	Duração média das execuções do agente	77
5.1.2	Número de execuções do agente	79
5.2	Performance Geral	81
6	Conclusões	87
6.1	Contributos	87
6.2	Trabalho Futuro	90
	Bibliografia	98
	Apêndices	98
A	Glossário	99
B	Mensagens ACL	101
B.1	<i>Performatives</i> e o seu significado	101
C	Monitorização da atividade dos Agentes	103
C.1	Agente 35	103
C.2	Agente 36	107

D Publicações	111
D.1 Tracking People and Equipment Simulation inside Healthcare Units	111
D.2 Intelligent Information System to Tracking Patients in Inten- sive Care Units	113
D.3 Interoperability in Healthcare	114
D.4 Intelligent Systems for Monitoring and Preventing Interopera- bility Procedures	115

Lista de Figuras

Lista de Figuras	xv
2.1 Método POMR.	16
2.2 Modelo LCIM.	26
3.1 Propriedades dos Agentes e a sua interação com o ambiente. . .	36
3.2 Exemplo de mensagem ACL.	38
3.3 Exemplo de transferência de dados utilizado <i>sockets</i> TCP/IP. . .	40
3.4 Arquitetura cliente servidor recorrendo a <i>sockets</i>	42
3.5 Camadas da Plataforma .NET.	43
4.1 Arquitetura da AIDA	49
4.2 Tipos de agentes que constituem a Plataforma AIDA	50
4.3 Arquitetura da BMaPI.	55
4.4 Interface do programa Servidor Geral.	57
4.5 Interface do programa Servidor Geral para a criação de novos agentes.	58
4.6 Processo automático de Inicialização da BMaPI.	60
4.7 Processo de criação de um agente.	62
4.8 Processo de execução da atividade de um agente.	63
4.9 Página inicial do Controlador <i>web</i>	65
4.10 Separador das Propriedades do agente selecionado.	66
4.11 Separador para Agendamento de atividades.	67
4.12 Separador para a Monitorização das atividades.	68
4.13 Separador para a Monitorização das atividades.	69

5.1	Página inicial do Controlador <i>web</i> no período analisado.	74
5.2	Página de programação da atividade do agente 609.	75
5.3	Página de propriedades do agente 609.	76
5.4	Análise diária do agente 609 para o dia 11 de setembro de 2013.	78
5.5	Duração média das atividades executadas por dia da semana de 10 a 15 de setembro.	79
5.6	Número de vezes que o agente executou a sua atividade em cada hora do dia 11 de setembro.	80
5.7	Número de vezes que o agente executou a sua atividade em cada dia da semana de 10 a 15 de setembro.	81
5.8	Comparação da percentagem de CPU livre.	82
5.9	Comparação da percentagem de memória RAM livre.	83
C.1	Página de programação da atividade do agente 35.	103
C.2	Página de propriedades do agente 35.	104
C.3	Duração média das atividades executadas por dia da semana de 10 a 15 de setembro.	104
C.4	Duração média das atividades executadas por hora para o dia 11 de setembro de 2013.	105
C.5	Duração média das atividades executadas por hora para o dia 13 de setembro de 2013.	105
C.6	Número de vezes que o agente executou a sua atividade por hora para o dia 11 de setembro de 2013.	106
C.7	Número de vezes que o agente executou a sua atividade em cada dia da semana de 10 a 15 de setembro.	106
C.8	Página de programação da atividade do agente 36.	107
C.9	Página de propriedades do agente 36.	108
C.10	Duração média das atividades executadas por dia da semana de 10 a 15 de setembro.	108
C.11	Duração média das atividades executadas por hora para o dia 11 de setembro de 2013.	109
C.12	Duração média das atividades executadas por hora para o dia 13 de setembro de 2013.	109

C.13 Número de vezes que o agente executou a sua atividade por
hora para o dia 11 de setembro de 2013. 110

C.14 Número de vezes que o agente executou a sua atividade em
cada dia da semana de 10 a 15 de setembro. 110

Lista de Tabelas

Lista de Tabelas	xvii
2.1 Implicações do modelo LCIM.	27
2.2 Níveis de Interoperabilidade.	28
3.1 Parâmetros das mensagens FIPA-ACL.	37
4.1 Vantagens e Desvantagens da Plataforma AIDA implementada no CHP.	52
B.1 Tipos de atos comunicativos FIPA - Parte 1.	101
B.2 Tipos de atos comunicativos FIPA - Parte 2.	102

Acrónimos

- ACL** *Agent Communication Language*. xiii, xvii, 5, 7, 8, 33, 37, 38, 54, 88
- AIDA** Agência para a Integração, Difusão e Arquivo de Informação Médica. v, vii, xiii, xvii, 1, 3–8, 47–54, 59, 70, 73, 81, 84, 85, 87–90
- AmI** *Ambient Intelligence*. 4
- AMS** *Agent Management System*. 56, 61–64
- BMaPI** *Biomedical Multiagent Platform for Interoperability*. v, vii, xiii, 1, 5–8, 47, 53–65, 67, 70, 71, 73–75, 81–84, 88–91
- CDA** *Clinical Document Architecture*. 21, 23, 24
- CDE** *Community Dashboard Editor*. 7, 33, 46, 64, 70
- CHP** Centro Hospitalar do Porto. v, vii, xvii, 5, 8, 47, 48, 51, 52, 54, 70, 73, 74, 85, 88
- CLR** *Common Language Runtime*. 44
- CPU** *Central Processing Unit*. xiv, 81–83, 89, 90
- DCMI** *Dublin Core Metadata Initiative*. 26, 28
- DICOM** *Digital Imaging and Communications in Medicine*. 21, 25
- EHR** *Electronic Health Records*. 6, 13–15, 17, 25
- FIPA** *The Foundation for Intelligent Physical Agents*. xvii, 8, 31, 32, 37, 38, 54, 101, 102
- HL7** *Health Level Seven*. 3, 20–24, 29
- HTML** *HyperText Markup Language*. 19, 46, 99
- IP** *Internet Protocol*. xiii, 7, 39–41, 44, 55, 58, 59, 61, 66, 88

- JADE** *Java Agent DEvelopment Framework*. 32
- MAS** *Multi-Agent Systems*. 4, 5, 7, 29–33, 36, 48, 50, 53, 87
- MCDT** *Meios Complementares de Diagnóstico e Terapêutica*. 50, 70
- PCE** *Processo Clínico Eletrónico*. 48
- POMR** *Problem Oriented Medical Record*. xiii, 16
- RAM** *Random Access Memory*. xiv, 81, 83, 89, 90
- RIM** *Reference Information Model*. 23, 24
- SAM** *Sistema de Apoio ao Médico*. 14, 49
- SAPE** *Sistema de Apoio às Práticas de Enfermagem*. 14, 49, 70, 74, 77
- SI** *Sistemas de Informação*. 2, 3, 12, 13, 17, 18, 22, 29–31
- SIH** *Sistemas de Informação Hospitalar*. 1–4, 6, 8, 11–13, 20–23, 30, 51, 52, 84, 87, 89
- SNOMED-CT** *Systematised Nomenclature of Medicine - Clinical Terminology*. 20, 24, 25
- SOA** *Service Oriented Architectures*. 4, 48
- SOAP** *Subjective, Objective Assessment and Plan*. 16
- TCP** *Transmission Control Protocol*. xiii, 7, 39, 40, 44, 55, 59, 88
- TI** *Tecnologias de Informação*. 11–13
- UDP** *User Datagram Protocol*. 40
- XML** *Extensible Markup Language*. 19, 24, 44, 45, 62, 63

Capítulo 1

Introdução

A presente dissertação descreve todo o trabalho desenvolvido na criação e implementação da *Biomedical Multiagent Platform for Interoperability (BMaPI)*. O projeto surge no âmbito da dissertação de mestrado em Informática Médica do Mestrado Integrado em Engenharia Biomédica da Universidade do Minho.

Neste primeiro capítulo é feita uma pequena contextualização e enquadramento do trabalho, focando os *Sistemas de Informação Hospitalar (SIH)*, a necessidade de comunicação e interoperabilidade entre estes sistemas e uma breve apresentação da *Agência para a Integração, Difusão e Arquivo de Informação Médica (AIDA)*. As motivações e os objetivos para este trabalho surgem também neste capítulo bem como a estrutura geral do documento.

1.1 Contextualização e Enquadramento

A saúde é uma das temáticas mais importantes na sociedade humana, uma vez que influencia diretamente a qualidade de vida dos cidadãos. É assim que surge a Informática Médica, uma área que alia duas ciências (a ciência da computação e a ciência da saúde) de forma a modernizar e melhorar a qualidade da prestação de serviços de saúde. Tendo em conta os aspetos tecnológicos e os aspetos económicos, a otimização da prestação de cuidados de saúde passa pela informatização dos processos, ou seja pela conversão

dos processos tradicionais baseados em papel em processos eletrônicos, recorrendo à utilização de **Sistemas de Informação (SI)**. Os **SIH** têm sido um assunto cada vez mais atrativo no âmbito da investigação científica, uma vez que providenciam aos profissionais de saúde um melhor contacto com os equipamentos, reduzindo assim o número e a incidência de erros médicos. Contribuem também para uma redução de custos e podem providenciar um meio para melhorar a gestão hospitalar [1–5].

Devido à sua especificidade, os **SIH** formam um ambiente complexo constituído por sistemas heterogêneos, distribuídos e ubíquos, que comunicam de diferentes formas, integram equipamentos médicos específicos e são desenvolvidos por entidades diferentes, com diferentes objetivos. Com o avanço da tecnologia, todos os dias são desenvolvidos novos sistemas, com o objetivo de ajudar os profissionais de saúde no seu trabalho diário. Contudo esses sistemas são geralmente construídos de uma forma isolada causando um determinado impacto no ambiente no qual são inseridos. Esse impacto revela-se especialmente no que diz respeito à informação que produzem, formando ao longo do tempo ilhas isoladas de dados, limitando dessa forma o fluxo de informação. Deste modo, emerge cada vez mais a necessidade dos **SIH** estarem em constante comunicação [1, 4–7].

A possibilidade e a necessidade de comunicar é uma das principais características dos seres humanos. Também os **SIH** necessitam de comunicar e cooperar de forma a melhorar o seu desempenho global, a sua utilidade, a qualidade dos diagnósticos, mas essencialmente, para melhorar a qualidade do tratamento dos pacientes. A cooperação e troca de dados e de informação é de facto uma das características mais relevantes, é a essência para a otimização dos recursos existentes e para uma melhoria do processo de tomada de decisão através da consolidação, verificação e disseminação de informação [1, 6, 8].

Surge a necessidade de introduzir neste ambiente a perceção de integração e de interoperação. A integração tem como principal objetivo adquirir e reunir informações de diferentes sistemas reforçando ou fortalecendo as suas potencialidades, enquanto que a interoperação concentra-se na comunicação contínua e troca de informação entre sistemas cooperativos. É então apresen-

tado o conceito de interoperabilidade, que define a capacidade para sistemas independentes, trocarem informações significativas e realizarem ações pertencentes a outros sistemas, de forma a operar em conjunto para um objetivo mútuo. O principal objetivo da interoperabilidade, integrada num ambiente hospitalar, é conectar os diversos sistemas e garantir que a informação é partilhada e distribuída para os profissionais de saúde, quando e onde eles necessitam dela. Interoperabilidade deixa assim de ser uma opção tecnológica passando a ser um requisito fundamental para uma prestação de cuidados de saúde eficaz, garantindo assim a saúde e o bem estar de milhões de pacientes em todo o mundo [1, 8, 9].

Nas últimas décadas a interoperabilidade e as respetivas implicações para a prestação de cuidados de saúde, têm sido submetidas a diversos estudos. Surgiu então a *Health Level Seven (HL7) International* [10], uma organização sem fins lucrativos, cujo principal objetivo é providenciar um enquadramento global e respetivos *standards* para a troca, integração, partilha e pesquisa de informação que suporta as prática médicas em formato eletrónico, e também a gestão, distribuição e avaliação dos serviços de saúde. A *HL7* providência *standards* para a interoperabilidade com múltiplos objetivos como: a melhoria da prestação de cuidados, a otimização do fluxo de trabalho diário, a redução da ambiguidade e o aumento da troca de conhecimento entre todos os intervenientes [1, 10].

Devido à complexidade de cada *SIH*, a possibilidade de criação de um *SI* global surge como algo complexo e incompleto. No entanto, a necessidade de recolher a informação importante para ser partilhada entre os diversos serviços e a comunicação de todos os dados relevantes do paciente, são ações de elevado interesse tanto para a instituição de saúde como para o paciente. De forma a agregar e consolidar toda a informação relevante, um processo de interoperação ou integração sólido e eficiente tem de ser desenvolvido. Este processo tem de ter em conta a escalabilidade, flexibilidade, portabilidade e segurança quando aplicado neste ambiente [1, 8].

A *AIDA* [11] surge então para fazer parte do processo de integração de diferentes fontes de informação, através do uso de diferentes protocolos e metodologias para resolver problemas e representar o conhecimento. Recorre a

modelos computacionais, tecnologias e ferramentas, que permitem a introdução de tecnologias como *Ambient Intelligence* (AmI) ou práticas baseadas em computação ubíqua nas unidades de saúde. A AIDA é então uma plataforma desenvolvida para permitir/facilitar a disseminação e integração de informação gerada em ambiente hospitalar. Incorpora diversas capacidades de integração, fazendo o uso principalmente de tecnologias como *Service Oriented Architectures* (SOA) e *Multi-Agent Systems* (MAS) para implementar a interoperabilidade num ambiente específico e distribuído, de acordo com os *standards* requeridos pelos prestadores de serviços que constituem uma instituição hospitalar [1, 3, 8].

Esta plataforma, desenhada para garantir a interoperabilidade entre os SIH, é caracterizada por aplicações eletrônicas que providenciam trabalhadores inteligentes, aqui entendidos como agentes de *software*, que têm um comportamento pró-ativo e são responsáveis por tarefas como a comunicação entre diferentes subsistemas, o envio e a receção de informação (por exemplo, relatórios clínicos ou médicos, imagens, recolha de dados, prescrições), a gestão da informação e a resposta a pedidos, utilizando os recursos necessários para as realizarem corretamente e em tempo útil. O principal objetivo é, como o próprio nome indica, integrar, difundir e arquivar grandes conjuntos de dados provenientes de diversas fontes (isto é, departamentos, serviços, unidades, computadores, equipamentos médicos, etc.). No entanto esta plataforma também providencia ferramentas para implementar e facilitar a comunicação com os seres humanos através de serviços baseados em *web* [1, 3, 7].

1.2 Motivação

Existem já diversas metodologias e arquiteturas através das quais é possível implementar a interoperabilidade entre os SIH. Essas metodologias são baseadas em arquiteturas de comunicação comuns e *standards*. A tecnologia MAS tem vindo a destacar-se na área da interoperabilidade, inclusivamente na interoperabilidade na saúde, abordando as preocupações anteriormente mencionadas. Um MAS é simplesmente um conjunto de agentes que comuni-

cam entre si, a comunicação é a base desta tecnologia e os agentes necessitam de partilhar uma linguagem de comunicação comum (*Agent Communication Language (ACL)* [12]) para ser possível a sua comunicação. A habilidade social confere a um agente a capacidade de interagir no MAS e cooperar de forma a cumprir os seus objetivos, contudo a autonomia e a pró-atividade são também características dos agentes e permitem-lhes planear e executar tarefas definidas para atingir determinados objetivos [13–15].

Com o aumento da quantidade e da complexidade dos dados numa instituição de saúde, o número de agentes criados para garantir a interoperabilidade tem vindo a aumentar. Surgiu então a necessidade de criar uma plataforma para controlar esses agentes. Uma ferramenta de auxílio ao administrador do sistema na criação do agente, na programação da sua atividade e especialmente na monitorização da sua atividade.

A motivação desta dissertação vem exatamente nesse sentido, sendo então numa primeira fase, projetado um estudo à AIDA, à sua forma, bem como ao MAS e aos agentes que a constituem, de forma a perceber o seu funcionamento e os seus componentes. Numa segunda fase projetou-se a criação da BMaPI, uma plataforma que permita a comunicação entre os agentes, independentemente da sua localização (máquina onde executam a sua atividade), com o objetivo de fornecer ao administrador do sistema uma forma simples para controlar e gerir a atividade destas entidades.

1.3 Objetivos

Para responder às características referidas anteriormente propõe-se, não só a criação da BMaPI em si, mas também a sua implementação num ambiente real de forma a validar as suas características e funcionalidades. Nesse sentido, a BMaPI foi desenvolvida tendo em conta as especificações do Centro Hospitalar do Porto (CHP) e da plataforma AIDA lá implementada.

Os objetivos desta dissertação encontram-se organizados na forma de questões que serão respondidas ao longo do documento.

Questão 1 - Após a análise da AIDA, quais são as suas vantagens e

limitações?

Questão 2 - Como garantir um maior controlo sobre os agentes que constituem a plataforma [AIDA](#)?

Questão 3 - Quais são as principais funcionalidades que a [BMaPI](#) deve garantir?

Questão 4 - Quais são as tecnologias a utilizar para o desenvolvimento da [BMaPI](#), nomeadamente para a comunicação entre os agentes?

Questão 5 - De que forma é que a solução apresentada melhora o desempenho global da [AIDA](#)?

Questão 6 - Quais são os aspetos que podem ser melhorados na [BMaPI](#)?

1.4 Estrutura

Este documento encontra-se organizado em seis capítulos: introdução, estado da arte, materiais e métodos, plataforma baseada em agentes para a interoperabilidade, apresentação e discussão dos resultados e conclusões. Este primeiro capítulo apresenta uma breve contextualização e enquadramento do trabalho, as motivações, os objetivos e a estrutura do documento.

O **Capítulo 2, Estado da Arte** apresenta conceitos teóricos importantes para este estudo, nomeadamente os [SIH](#) as implicações inerentes à sua implementação em ambiente real e a sua necessidade de comunicar de forma a atingirem a sua performance global. É descrito o sistema de registos clínicos eletrónicos, *Electronic Health Records* ([EHR](#)) como um dos principais [SIH](#). De seguida é exposto o conceito de interoperabilidade, a sua necessidade extrema nas unidades de saúde, as suas características e os diferentes tipos de interoperabilidade. Mais detalhadamente são apresentados os princípios da interoperabilidade e os *standards* para a interoperabilidade na saúde. Ainda

relativamente à interoperabilidade é explorada a utilização de agentes na sua implementação, nomeadamente os MAS. E por fim são apresentadas algumas soluções para a interoperabilidade baseadas em agentes, tanto em ambiente hospitalar como em ambientes distintos.

Segue-se o **Capítulo 3, Materiais e Métodos**, onde são expostas as tecnologias e ferramentas utilizadas para o desenvolvimento deste projeto. É explorada em primeiro lugar a tecnologia orientada a agentes e as suas características. Ainda relativamente aos agentes é apresentada a forma utilizada para estabelecer a comunicação entre estas entidades, através da utilização de mensagens ACL e de *sockets* com o recurso a protocolos *Transmission Control Protocol (TCP)/Internet Protocol (IP)* e com a arquitetura Cliente/-Servidor para o transporte dessas mensagens. A *framework .NET* é também neste capítulo descrita, as suas principais características são salientadas e são também apresentadas as classes e *Namespace*s utilizados no desenvolvimento do sistema. Para terminar é apresentada a ferramenta *Pentaho Community* e mais especificamente o seu editor *Community Dashboard Editor (CDE)* utilizado para a construção de gráficos incluídos num dos componentes da BMaPI.

O **Capítulo 4, Plataforma baseada em agentes para a Interoperabilidade** como o próprio nome indica apresenta detalhadamente a BMaPI. Antes da apresentação da BMaPI, é apresentada a plataforma AIDA, ambiente onde irá ser inserida a plataforma desenvolvida. As suas principais características são destacadas seguindo-se da apresentação da arquitetura e dos componentes da AIDA. Por fim, faz-se um levantamento das principais vantagens e limitações da plataforma e é então apresentada a BMaPI como a solução para uma das limitações existentes, de forma a melhorar a performance global da AIDA. São descritas as principais funcionalidades e a arquitetura da BMaPI, seguindo-se de uma descrição detalhada de cada um dos seus componentes, as respetivas funcionalidades e o seu funcionamento do ponto de vista do utilizador e do ponto de vista técnico e tecnológico. O capítulo termina com a descrição da implementação da BMaPI na AIDA do

CHP.

O **Capítulo 5, Apresentação e Discussão dos resultados**, expõe os resultados obtidos através da implementação da **BMaPI** no **CHP**, durante o período de 10 a 16 de setembro de 2013. É feita uma análise detalhada dos resultados, explicando possíveis acontecimentos para os resultados obtidos. O capítulo é encerrado com uma análise à performance global da **BMaPI**, nomeadamente a avaliação do seu comportamento na máquina onde foi implementada, sendo posteriormente apresentada uma breve discussão das capacidades da **BMaPI** e da sua importância na plataforma **AIDA** e no ambiente geral onde foi implementado.

O **Capítulo 6, Conclusões**, apresenta as principais conclusões inerentes a este projeto e ainda faz um resumo das principais contribuições que a **BMaPI** oferece não só na sua execução individual mas também através da sua implementação num ambiente real. De uma forma geral e resumida são expostos os objetivos propostos e a forma como foram alcançados. Por fim surge uma secção de trabalho futuro que apresenta quais as medidas e de que forma é que podem ser implementadas para melhorar a **BMaPI** ou para adicionar características que revelaram ser importantes para a **BMaPI**.

O **Apêndice B**, no contexto das **Mensagens ACL** apresenta o conjunto completo dos tipos de atos comunicativos criados pela *The Foundation for Intelligent Physical Agents (FIPA)* e o seu significado/utilização.

O **Apêndice C**, denominado **Monitorização da atividade dos agentes**, apresenta os resultados relativos aos agentes 35 e 36, também eles inseridos na **BMaPI** no período de testes, apresentado no Capítulo 5.

O **Apêndice D, Publicações**, apresenta as publicações científicas realizadas no âmbito deste projeto. Numa primeira fase do projeto foram efetuadas publicações sobre os **SIH** desenvolvidos pelo mesmo grupo de investigação, alguns com implementação e integração na plataforma **AIDA**, numa

fase posterior foi escrito um capítulo de um livro sobre a interoperabilidade na área da saúde, por fim foi ainda escrito um último artigo relacionado com projeto apresentado neste documento.

Capítulo 2

Estado da Arte

Neste capítulo são apresentados de forma detalhada os conceitos teóricos necessários para este estudo. Sendo a interoperabilidade dos **SIH** a temática central deste trabalho, em primeiro lugar é apresentado o conceito de **SIH** e a sua influência no ambiente hospitalar. O conceito de interoperabilidade, as suas características, os seus princípios e os *standards* para a sua implementação, são detalhadamente expostos neste capítulo. Por fim destaca-se a tecnologia baseada em agentes, como um meio para a implementação da interoperabilidade, e são brevemente apresentados alguns sistemas existentes com esse objetivo.

2.1 Sistemas de Informação Hospitalar

Atualmente, as **Tecnologias de Informação (TI)** têm adquirido um papel importante no fluxo de trabalho, informação e conhecimento quando inseridas nas instituições de saúde, influenciando assim o seu funcionamento interno. A unicidade de cada serviço prestado e a tecnologia implementada, requer **SIH** particularmente adaptados que necessitam de trocar informação e garantir tanto a qualidade da informação como capacidades técnicas eficazes [16].

Investimentos consideráveis têm sido feitos pela maior parte das instituições de saúde, de forma a desenvolver **SIH** adequados com capacidades de integração e interoperabilidade; liberdade no desenvolvimento de novos **SIH**

e desativação dos **SIH** obsoletos; consolidação e disseminação da informação médica; melhoria da integridade e disponibilidade dos serviços prestados; utilização de terminologias médicas claras e *standards*; adaptabilidade de forma a suportar mudanças nos fluxos de trabalho ou nos modelos de representação do conhecimento; e disponibilidade da informação e do conhecimento quando e onde for necessária [16–18].

O aumento do desenvolvimento dos **SIH** está relacionado com a preocupação universal e omnipresente da melhoria dos serviços prestados pelas unidades de saúde, e com a redução dos custos associados a estes serviços, através da otimização dos recursos existentes [18]. Um **SIH** constitui a principal ferramenta neste ambiente, deve ser partilhado por todas as soluções existentes nas diversas áreas da unidade de saúde. Por outras palavras, um **SIH** deve ser definido como um **SI** apto para o processamento e troca de dados, informação e conhecimento num ambiente hospitalar. Especificamente um **SIH** deve fazer uso dos sistemas existentes para recolher, processar, relatar e usar a informação e o conhecimento que existe num ambiente hospitalar, para suportar decisões e políticas de gestão, programas de saúde, ensino, pesquisa e especialmente práticas médicas na instituição [19]. Deve destinar-se não só ao tratamento da informação relativa ao paciente, mas também para facilitar a extração de indicadores clínicos ou de gestão, que permitem aos profissionais tomar decisões baseadas em informação com qualidade, e melhorar a planificação de programas direcionados para a melhoria da prestação de serviços de saúde.

Doravante, um **SIH** deve conectar horizontalmente qualquer serviço e aplicação existente numa unidade de saúde, necessitando de interoperação entre esses serviços distintos e integração daquilo que contêm. Um **SIH** deve ser considerado como uma estrutura combinada de todos os serviços existentes e aplicações, que incorporam toda a informação disponível num determinado momento numa unidade de saúde [17].

A saúde e os serviços a ela associados estão a ser melhorados e mudados constantemente, não só a nível clínico mas também no uso das **TI**. A documentação baseada em papel tem sido gradualmente abandonada, enquanto que os computadores e as novas tecnologias tem sido incluídos no funciona-

mento diários dos diversos serviços nas unidades hospitalares. Por esta razão, a capacidade das TI para permitir acesso fácil a todas as informações de um paciente e apresentar o conhecimento médico apropriado como suporte para o processo de tomada de decisão, constitui a preocupação predominante de um SIH [18].

Cada vez mais, os profissionais de saúde tomam como base para um correto diagnóstico, a informação relativa ao paciente. Com os avanços económicos e tecnológicos, o conjunto de informações disponíveis sobre o paciente encontra-se distribuído por diferentes SI e têm alcançado elevadíssimas dimensões, muitas vezes com diferente estrutura, formato e até informação codificada, tornando assim a função de pesquisa por parte do profissional de saúde um processo difícil e bastante moroso. Surge então como solução, nos anos 90, os registos clínicos eletrónicos (EHR) uma aplicação central que cobre horizontalmente a instituição de saúde virtual e possibilita uma análise transversal dos registos clínicos dos vários serviços, unidades e até registos sobre outros tratamentos anteriormente realizados, permitindo que os profissionais de saúde pesquisem toda a informação que necessitam e até comparem toda a história clínica do paciente, para que o seu tratamento/diagnóstico tenha a melhor qualidade possível.

2.2 *Electronic Health Records*

Na década de 1990 nos Estados Unidos da América surgiu o conceito de EHR sempre virado para o apoio aos cuidados de saúde e para manter e melhorar a qualidade dos serviços. Atualmente o EHR é já utilizado na maior parte das instituições de saúde a nível mundial e assume-se como um SIH de excelência que substitui os registos tradicionais feitos em papel. O conceito de EHR é descrito como uma coleção abrangente, multi-institucional e longitudinal de informação clínica, administrativa e financeira [20, 21].

Existem várias definições para EHR, mas em geral, pode ser definido como o registo informatizado de dados clínicos dos pacientes, ou seja todas as informações fornecidas por dados demográficos, dados relativos à história clínica, dados laboratoriais, entre outros. Estes dados são introduzidos num

sistema eletrónico, que permite a captura, manutenção, transmissão e armazenamento de informação clínica, que é essencial não só para a monitorização do estado de saúde de cada paciente, mas também para fins como a gestão de custos. Em Portugal o primeiro avanço do registo eletrónico surgiu com um sistema chamado SONHO, este sistema está focado unicamente na gestão administrativa. Atualmente, existem já sistemas de gestão da informação clínica como o Sistema de Apoio ao Médico (SAM) e o Sistema de Apoio às Práticas de Enfermagem (SAPE).

Um EHR constitui um conjunto de documentos *standard*, ordenado e conciso, dirigido ao registo de ações e procedimentos médicos, um conjunto de informações compiladas por médicos e outros profissionais de saúde que relatam toda as informações sobre a saúde dos pacientes. O principal objetivo é substituir os documentos físicos em eletrónicos aumentando o processamento dos dados e diminuindo o tempo e os custos [20, 22].

A fim de facilitar e melhorar o atendimento nos serviços de saúde, o EHR serve como suporte à decisão ou unicamente como fonte de informação, reduz a perda de dados ou a frequência de registo de dados, reduz a ocorrência de erros médicos, gere os recursos e melhora os procedimentos de avaliação e gestão de custos. A principal diferença em relação aos registos efetuados em papel está relacionada com a interação entre as várias fontes de informação heterogéneas [20, 22].

Além dos requisitos organizacionais, funcionais, técnicos e científicos, é necessário ter em conta as necessidades éticas e legais bem como a qualidade dos dados, segurança da informação, controlo de acessos e privacidade. Deve ser rápido, fiável e seguro, a disponibilidade do sistema deve ser elevada, isto é deve estar disponível 24 horas por dia. A sua estrutura deve permitir uma boa integração com a sua evolvente, promovendo a omnipresença de registos entre diferentes especialidades e serviços. Essa omnipresença permitirá o acesso a mecanismos de alerta, monitorização e apoio à decisão [20, 22].

O EHR permite a construção de documentos e relatórios personalizados para fins específicos, torna mais fácil a configuração da interface do utilizador, permitindo o uso de diferentes *layouts* de inserção e visualização de informação, garante a legibilidade dos dados, permite o processamento e re-

colha automática de dados contínuos, podendo até alertar quando estes se encontram fora de determinados parâmetros [20, 22].

Em suma um **EHR** deve conter [23]:

- Informação clínica passada e atual de um determinado indivíduo, resultante dos cuidados de saúde prestados;
- Observações e aspetos clínicos observados por um médico, enfermeiro ou outro profissional de saúde durante um atendimento em modo de ambulatório;
- Resultados de testes obtidos durante a prestação de cuidados;
- Informação resultante do contacto médico-paciente fora do ambiente clínico, tais como informações que podem ser transmitidas por telefone ou outros meios;
- Informações de carácter administrativo pertinentes para a tomada de decisão clínica;
- Dados relativos à população tais como registo de doenças e iniciativas para detetar surtos;
- Informações para facilitar e otimizar a gestão e coordenação do cuidado de um indivíduo que fica sujeito a múltiplos serviços e consequentemente diferentes profissionais de saúde;
- Informação que pode intervir na situação clínica ou utilizada para interpretar os dados e apoiar as decisões clínicas;
- Informações sobre evidências médicas, estudos científicos ou situações ambientais;
- Informações provenientes de dispositivos remotos de monitorização que capturam dados em tempo real sobre os sinais vitais, dados de laboratório, entre outros.

O processo de recolha dos dados baseia-se no método *Problem Oriented Medical Record* (POMR) assumindo que o registo é uma produção de documentos clínicos, um formato de registo clínico composto por uma lista de problemas; uma base de dados que inclui o histórico do paciente; planos de diagnóstico, terapêuticos e educacionais; e as notas de progressão diárias baseadas no plano *Subjective, Objective Assessment and Plan* (SOAP). A lista de problemas serve como um índice para o leitor onde cada problema é seguido até à respetiva resolução. Este sistema realça o reconhecimento das cinco fases diferentes do processo de decisão, ou seja, a recolha de dados, a formulação dos problemas, a elaboração de um plano de gestão, a análise da situação e caso necessário a sua revisão [3, 20, 24].

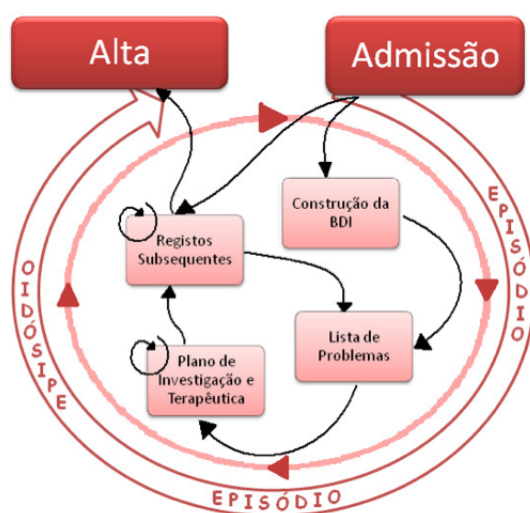


Figura 2.1: Método POMR (Adaptado de [3])

Analisando a Figura 2.1, verifica-se que todos os episódios começam com uma ficha de admissão e terminam com a alta. Um episódio constitui assim o conjunto de todas as operações para o paciente desde o início até ao fim do tratamento. É de salientar que os registos podem ser atualizados durante o episódio e ao longo da evolução do paciente.

O processamento de dados é feito tendo em consideração a organização, estrutura, sistematização e codificação, usando técnicas no domínio da representação do conhecimento e raciocínio. Por outro lado, o uso de sistemas de

codificação e ontologias permitem um registo mais espontâneo, automático, eficiente, adaptativo e inteligente [20].

Independentemente do objetivo principal se focar na melhoria da qualidade dos cuidados de saúde, novos desafios tem surgido. No conjunto desses desafios emerge a agregação e consolidação da informação num processo sólido e eficiente de interoperabilidade e integração. Contudo quando um processo deste tipo é aplicado ao EHR é necessário ter em conta a escalabilidade, flexibilidade, portabilidade e segurança uma vez que a informação é bastante complexa e sensível podendo influenciar de forma grave o diagnóstico do paciente quando mal registada. Para evitar esta complexidade ética e moral deve ser realizada uma profunda validação da informação trocada e integrada neste processo. O desenvolvimento de uma plataforma com alto nível de interoperabilidade é cada vez mais a essência de um ambiente hospitalar [1, 8, 25].

2.3 Interoperabilidade

Nas últimas décadas, a principal preocupação das organizações em geral é introduzir tecnologia nos seus ambientes, também nas unidades de saúde são desenvolvidos novos SI para uma melhor gestão da qualidade e da prestação de serviços de saúde, nomeadamente ferramentas para o agendamento, faturação e registo de informação médica. Apesar de aumentarem a qualidade dos serviços de saúde, esses sistemas são desenvolvidos de uma forma isolada, falhando na capacidade de interagir efetivamente [4].

É então de fácil perceção que hoje em dia, a tecnologia implementada na medicina e na saúde esteja a passar por uma situação complicada, na medida em que foram introduzidos no fluxo diário de trabalho, diferentes SI com um determinado grau de independência. Essa independência constrói um ambiente heterogéneo onde se torna cada vez mais difícil a gestão dos dados e da informação que deve estar acessível quando e onde for necessária. No cenário atual, a troca de dados e de informação é a essência da otimização dos recursos existentes e da melhoria do processo de tomada de decisão através da consolidação, verificação e disseminação da informação. Para re-

resolver os problemas da independência dos SI, é necessária a comunicação e a cooperação de forma a melhorar o seu desempenho global, a sua utilidade, a qualidade dos diagnósticos e principalmente para melhorar a qualidade da prestação de cuidados de saúde ao paciente [1, 26, 27].

São por isso apresentados os conceitos de integração e interoperação em diferentes níveis conceptuais e com objetivos distintos. Ambos são importantes para a cooperação e o fluxo de informação nas organizações de saúde, contudo são baseados em princípios distintos. De uma forma simples, a integração visa reunir e adquirir informação de diversos sistemas para um outro sistema que solicitou essa mesma informação, enquanto que a interoperação centra o seu objetivo na comunicação contínua e troca de informação entre sistemas cooperativos [1, 8].

A interoperabilidade é definida então pela capacidade que um sistema ou um produto tem, para trabalhar com outros sistemas ou produtos, sem que haja um esforço especial por parte do utilizador. De outro ponto de vista e de uma forma mais completa pode também ser definida como a capacidade que um sistema independente tem para trocar informação significativa e iniciar ações pertencentes a outros sistemas de forma a trabalharem em conjunto para atingir um benefício mútuo [8].

O principal objetivo da interoperabilidade, na área da saúde é conectar aplicações e dados, de forma a poderem ser partilhados por todo o ambiente e distribuídos para os profissionais de saúde, quando e onde for necessária. Tem vindo a revelar-se um requisito fundamental para a prestação de cuidados de saúde e para o bem estar de milhões de pacientes em todo o mundo [8, 9].

Ainda assim, garantir a interoperabilidade num ambiente hospitalar é uma tarefa difícil, especialmente por causa da complexidade e especificidade da informação médica e também devido a problemas sociopolíticos e éticos. Contudo estima-se que a criação de um sistema *standard* para a interoperabilidade nos Estados Unidos renderia ao estado um lucro de mais de 75 bilhões de dólares americanos por ano [9].

Para além da complexidade inerente à sua implementação existem também vários tipos de interoperabilidade. A maior parte dos autores considera apenas a interoperabilidade sintática e a interoperabilidade semântica,

contudo *C. N. Mead et al* [28] distingue a semântica humano-humano e a semântica máquina-máquina:

- **Sintática** - Garante a troca da estrutura dos dados, mas não garante que o significado seja interpretado de forma idêntica por todas as partes. As páginas *web* construídas com *HyperText Markup Language* (HTML) ou *Extensible Markup Language* (XML) são bons exemplos da interoperabilidade sintática máquina-máquina uma vez que a sua estrutura própria pode ser lida por qualquer máquina com um *web browser*, já o significado da página para uma determinada máquina pode variar substancialmente, contudo isso geralmente não é considerado um problema, porque a semântica de uma página é feita para ser interpretada por humanos.
- **Semântica Humana** - Garante que o significado dos dados não é ambíguo quando trocados entre humanos. Documentos como as notas de progresso, referências, consultas e outros dependem da especificidade do vocabulário do médico e das suas práticas comuns para garantir a interoperabilidade semântica ao nível médico-médico.
- **Semântica Computacional** - Requer que o significado dos dados não seja ambíguo na troca de dados entre máquinas. Não significa necessariamente que todas as máquinas necessitam de processar os dados recebidos da mesma forma, mas cada máquina tomará o seu processo de decisão com base no mesmo significado.

Em particular, o estabelecimento da interoperabilidade semântica entre as fontes de informação heterogêneas e distribuídas tem sido um problema crítico que tem atraído uma atenção significativa por parte de investigadores. Quando se tenta entender a semântica dos dados, é necessário identificar o contexto em que estes se encontram inseridos. Diferentes unidades funcionais da mesma organização podem interpretar e utilizar as informações de formas distintas, é então necessário superar os conflitos semânticos decorrentes dessas diferenças de significados implícitos, perspectivas e suposições, criando assim informação semanticamente compatível com o meio ambiente [29].

Considerando este problema, toda a informação a ser transferida tem de ser normalizada, de forma a evitar diferentes estruturas e interpretações erradas. Assim sendo, o uso de *standards* garante uma melhor comunicação entre os profissionais de saúde e a interoperabilidade entre os sistemas, permitindo alguma automatização dos registos hospitalares.

Os standards referentes aos registos hospitalares são categorizados para três fins diferentes [7]:

- *Standards* para representar a informação clínica;
- *Standards* para a comunicação;
- *Standards* de imagem.

2.3.1 *Standards* para a Interoperabilidade

Existem diversos significados e motivações para a utilização de *standards*, tais como : a uniformização da produção, a compatibilidade de tecnologias, a objetividade na medição, entre outros. Também a compatibilidade na forma de interoperabilidade de *software* tem sido enfatizado nos últimos anos, devido ao rápido avanço da tecnologia e à conectividade global [30].

A complexidade que advém da própria natureza dos dados de saúde, confere aos SIH dificuldades, na integração dos mesmos e problemas de interoperabilidade. Uma forma de resolver tais problemas é recorrendo ao uso de um modelo comum de dados. Contudo o grau de integração dos SIH depende da flexibilidade do modelo empregue. Tem sido feito um esforço intensivo para desenvolver *standards* adaptados e otimizados para a área da saúde, o resultado deu origem a vários modelos contudo alguns deles não eram muito consensuais em avaliações comparativas, destacando-se assim o HL7 como sendo considerado o *standard* mais adaptável à interoperabilidade na saúde [8, 30, 31].

Tal como já foi referido os *standards* na saúde são considerados a principal fonte para garantir a interoperabilidade entre os SIH. Estes incluem *standards* relacionados com as mensagens (HL7), as terminologias (*Systematised*

Nomenclature of Medicine - Clinical Terminology (SNOMED-CT)), as informações clínicas e os registos do paciente (*openEHR* e *HL7 Clinical Document Architecture* (CDA)) e as imagens (*Digital Imaging and Communications in Medicine* (DICOM)) [31]. Cada um destes será brevemente descrito dando especial interesse ao HL7, na medida em que é o *standard* responsável pela troca eletrónica de dados entre SIH.

Health Level Seven (HL7)

Fundada em 1987, a organização HL7 [10] surgiu para lidar com a crescente diversidade de mensagens desenvolvidas no sector dos seguros de saúde nos Estados Unidos da América. O protocolo HL7 é um conjunto de formatos *standard* que especificam as interfaces para a troca eletrónica de dados entre aplicações informáticas heterogéneas em ambientes hospitalares. O HL7 começou por ser o principal protocolo de comunicação orientado à interoperabilidade sintática na saúde, ao nível aplicacional. As versões iniciais do protocolo, definiam as mensagens a ser trocadas por aplicações de saúde fracamente ligadas através da classificação dos diferentes tipos de mensagens envolvidas neste ambiente e com a agregação de segmentos *standard* [8,22,32].

O objetivo do HL7 está então centrado na sintaxe da informação que é trocada, no entanto a estrutura e o *design* deste *standard* definem que os dados devem ser transferidos através de uma mensagem, habilitando e potencializando a aplicação do HL7 numa arquitetura cliente-servidor. Sendo assim a implementação mais comum do HL7 é baseada na comunicação entre diferentes clientes e servidores *socket*, onde o cliente envia uma mensagem HL7 estruturada ao servidor, que após o processamento envia uma mensagem HL7 *standard* como *acknowledgement* [32].

Tal como foi referido anteriormente, as versões iniciais do HL7 garantiam exclusivamente aspetos sintáticos, e de acordo com os modelos gerais de interoperabilidade é um dos níveis mais baixos deste processo. A atual versão do HL7 - 3, surge com o objetivo de possibilitar o alcance do HL7 para a interoperabilidade semântica, incluindo o uso adequado das informações trocadas. A principal característica desta versão é especificar o conteúdo da informação

das mensagens, através de um modelo de informação que esclarece as definições e garante que elas são usadas de forma consistente. Esta versão contém relações e meta-dados num nível abstrato que elevam o nível de integração, nomeadamente através da interoperabilidade semântica e da validação das informações trocadas. Os meta-dados e arquétipos definidos pelo HL7 permitem a produção e organização dos dados clínicos em segmentos e campos claramente definidos e ligados, que podem ser validados [22, 32].

Embora o desenvolvimento da versão 3 apresente diversas melhorias em relação à versão anterior, esta ainda é a mais usada em SI e equipamentos de saúde. As mensagens nesta versão são definidas e identificadas pelo seu segmento controlo. No segmento de controlo várias regras são aplicadas a todas as mensagens, sendo então definidos [32]:

- Segmento da Mensagem
- Tipo da Mensagem
- Eventos *trigger*

O princípio fundamental é o fluxo de informação entre os sistemas heterogêneos que compõem os SIH. A maioria dos eventos no ambiente hospitalar funcionam como *trigger* para o início da divulgação da informação. O HL7 foi desenvolvido com o propósito do acontecimento chamado *trigger* provocar a troca de mensagens entre um par de aplicações. Quando ocorre um evento num sistema compatível com HL7, uma mensagem HL7 é preparada recolhendo todos os dados necessários a partir da aplicação subjacente e é enviada para o sistema requerente [33].

Um evento *trigger* pode ser proveniente de uma das seguintes fontes [32]:

- Baseado num pedido de utilizador (neste documento, também referido como ambiente) - Por exemplo, o evento *trigger* solicita a um sistema para enviar todos os dados acumulados num sistema de recolha, a cada 12 horas, é considerado ambiental. Da mesma forma, quando um utilizador pressiona um botão numa determinada interface, também pode ser considerado ambiental.

- Transição de estado - o evento resulta de uma transição de estado. O *trigger* para o cancelamento de um agendamento, por exemplo, pode ser considerado um *trigger* baseado numa transição de estado.
- Baseado na interação - baseado numa receção de uma outra interação. Por exemplo, a resposta a uma consulta (que é uma interação) é um evento *trigger* baseado na interação.

Deste ponto de vista o fluxo de informação entre um SIH e os elementos numa instituição de saúde é feito através deste eventos, ou por outras palavras, todo o SIH é regido por esses eventos. É o papel deste *standard* regular e definir todos os eventos bem como as suas implicações e as informações necessárias para os procedimentos subjacentes [32].

O facto de a maior parte das comunicações serem atualmente realizadas com uma norma sintática, como o HL7 versão 2, resulta na perda de modularidade e numa conexão inerente mais forte, do que o que seria de esperar entre todos sistemas. Além disso, a complexidade e especificidade da interoperação entre os sistemas, restringe uma avaliação *standard* e estendida da informação da mensagem antes de a partilhar pelos sistemas que compõem o SIH [32].

Apesar de outros padrões e tecnologias tais como HL7 versão 3 ou *openEHR* permitirem um raciocínio e validação mais semântica, a sua implementação em ambiente real está longe, considerando as dificuldades existentes devido à antiguidade dos sistemas, entre outras. Além disso, a eficiência do HL7 versão 2, e a sua otimização ao longo dos anos transforma a migração destes padrões e tecnologias muito mais complexa [32].

HL7 CDA e *openEHR*

A atual versão do HL7 (3) baseia-se num modelo de dados orientado a objetos, chamando modelo de referência de informação (*Reference Information Model* (RIM)), um meio de especificar o conteúdo da informação da mensagem, através de um modelo de informação que esclarece as definições e garante que elas são usadas de forma consistente. O HL7 CDA é uma

estrutura genérica derivada do RIM para a comunicação de documentos clínicos. Por vezes tem sido considerado como o equivalente ao HL7 de uma arquitetura de registo, embora tenha sido concebido como um mecanismo de transferência de um único documento [22, 33].

Os documentos CDA são codificados em XML e estão organizados em três níveis, em que cada nível iterativamente adiciona mais noções aos documentos clínicos, embora o índice clínico se mantenha constante em todos os níveis [33].

O nível um centra-se no conteúdo dos documentos narrativos com contexto de alto nível, tais como datas, horários, lugares e organizações estruturais. Destina-se principalmente a representar documentos de estilo narrativo e algumas informações básicas de cabeçalho de forma estruturada. É constituído por duas partes, o cabeçalho e o corpo do CDA, baseados nos tipos de dados HL7. O cabeçalho do documento é derivado do RIM e inequivocamente define a semântica de cada entrada do documento. O corpo contém o conteúdo do documento clínico e pode ser texto não estruturado, ou composto por estruturas como secções, parágrafos, listas e tabelas através de marcação estruturada [22, 33].

O nível dois permite restringir tanto a estrutura como o conteúdo de um documento através de um molde, aumentando assim a interoperabilidade, uma vez que o recetor tem conhecimento do que vai receber [33].

O nível três vem proporcionar uma visão mais aprofundada, onde umas partes do documento destinam-se apenas a leitores humanos, enquanto outras partes são tanto para leitores humanos como para o processamento automático pois são codificadas em XML [33].

SNOMED-CT

SNOMED-CT é uma *framework* que pode ser utilizada por profissionais de saúde, administradores e investigadores na área da medicina como uma forma de melhorar a qualidade dos serviços prestados aos pacientes, através de uma representação eficiente e concisa da informação. Esta *framework* auxilia a organização dos termos médicos, e pode ser incorporada em aplicações

como **EHR**, com a intenção de permitir o armazenamento dessa informação de uma forma uniforme, auxiliando o seu processamento e análise automática. Os sistemas onde a terminologia **SNOMED-CT** se encontra implementada contêm as seguintes características [8]:

- Pesquisa eficiente nos registos clínicos do paciente;
- Extração de informação clínica relevante;
- Identificação automática dos fatores de risco associados ao paciente;
- Monitorização da resposta ao tratamento;
- Monitorização das reações adversas ao tratamento;
- Análise de doenças na comunidade e previsão da sua incidência;
- Melhora a consistência dos dados clínicos para pesquisa clínica.

DICOM

O *standard DICOM* é a representação comum de dados mais utilizada para imagens médicas, define as estruturas e serviços de dados para troca de imagens médicas e informações relacionadas. O *standard DICOM* utiliza uma codificação binária com listas hierárquicas de dados identificados por etiquetas numéricas e de um complexo [33].

2.3.2 Princípios da Interoperabilidade

As características e as metodologias intrínsecas para resolver o problema de ajuste de interoperabilidade são complexas para serem determinadas num modelo único e absoluto. Existem então vários modelos considerados válidos, esses modelos tentam clarificar abordagens de interoperabilidade com um ou mais conjuntos de atributos, definindo o nível de troca de dados e abstração, a implementação tecnológica, o ponto de vista interoperável e propósitos subjacentes [8].

A *Dublin Core Metadata Initiative* (DCMI) é uma organização que tem como objetivo fornecer *standards* simples para facilitar a descoberta, a partilha e a gestão de informação. Criou uma *framework* de interoperabilidade ao nível dos sistemas, baseia-se principalmente no nível de abstração dos registos ou dos dados permutáveis e deve ser entendido em quatro níveis, nomeadamente [8]:

- Nível 1 - Definições dos termos de partilha - definição da linguagem de partilha dos componentes de dados;
- Nível 2 - Semântica formal da interoperabilidade - dados são baseados na semântica formal;
- Nível 3 - Conjunto de descrições da interoperabilidade sintática - os dados são estruturados de acordo com o vocabulário formal;
- Nível 4 - Descrição do conjunto de perfis de interoperabilidade - o conteúdo dos dados é estruturado de acordo com o vocabulário formal e limitado por um conjunto de invariantes.

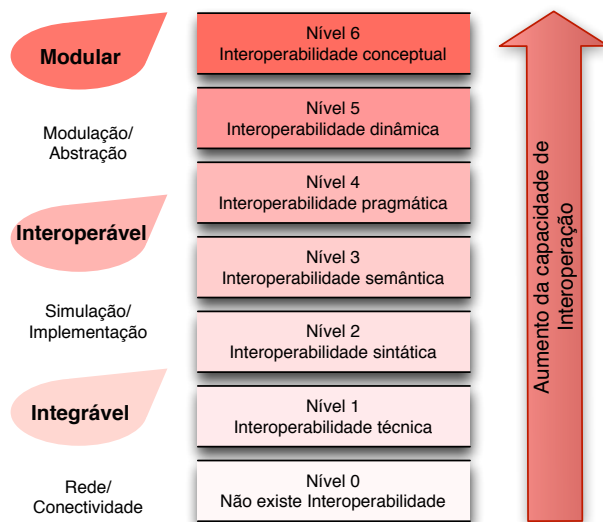


Figura 2.2: Modelo LCIM (Adaptado de [34])

Tabela 2.1: Implicações do modelo LCIM (Adaptado de [34]).

Nível	Designação	Nível de abstração da informação	Informação definida	Conteúdo claramente definido	Domínio	Foco
N6	Conceptual	Modelo comum conceptual	Suposições ou restrições	Modelos conceptuais documentados	Modelação e Abstracção	Modular
N5	Dinâmica	Modelo comum de execução	Efeito dos dados	Efeito da informação trocada		
N4	Pragmática	Modelo comum de fluxo de trabalho	Uso dos dados	Contexto da informação trocada	Simulação e Implementação	Interoperável
N3	Semântica	Modelo comum de referência	Significado dos dados	Conteúdo da informação trocada		
N2	Sintática	Estrutura comum de dados	Dados estruturados	Formato da informação trocada		
N1	Técnica	Protocolo comum de comunicação	Dados binários	Símbolo da informação trocada	Conexão à rede	Integrável
N0	X	Não há conexão	X	X		

Outro modelo proposto por *Tolk e Muqira* visa dividir os níveis conceptuais dos fatores de interoperabilidade de acordo com os níveis de abstracção dos dados partilhados, as implementações tecnológicas e o objetivo subjacente. Estes sete níveis, apresentados na Figura 2.2 e na Tabela 2.1 mais detalhadamente, mostram a distinção desde a inexistência de interoperabilidade (Nível 0) até à interoperabilidade conceptual (Nível 6). Neste modelo os conceitos abstratos e também as metodologias para resolver problemas encontra-se agrupadas de acordo com o seu impacto e o potencial do processo de interoperabilidade [34].

Estes níveis são orientados para a DCMI contudo podem ser extrapolados para outros modelos de meta-dados.

Uma descrição distinta foi adaptada por *Mykkanen e Toumainen* (Tabela 2.2) cujo intuito é avaliar e classificar os *standards* e modelos de interoperabilidade e para identificar aspetos de interoperabilidade que não são abrangidos pelos *standards* e devem ser resolvidos de outras formas específicas [30].

Tabela 2.2: Níveis de Interoperabilidade(Adaptado de [30]).

Nível	Designação	Significado
N7	Interfaces do ciclo de vida da aplicação	O ciclo de vida da aplicação, incluindo métodos de integração e de desenvolvimento
N6	Modelo de referência funcional	a informação de domínio específico ou o modelo funcional ou suposições sobre os métodos utilizados
N5	Semântica	O significado dos elementos definidos
N4	Interfaces Funcionais	A funcionalidade e a informação definida
N3	Infraestrutura da aplicação	Os pontos de integração da arquitetura distribuída das aplicações participantes
N2	Infraestrutura técnica	A infraestrutura para suportar a tecnologia da interface e da comunicação
N1	Interfaces técnicas	A tecnologia utilizada nas interfaces e implementações

Várias tendências em relação ao nível de abstração das informações trocadas e do uso que é feito de tal informação pode ser detetado com o uso destes modelos. Analisando os vários modelos apresentados pode verificar-se que a noção de nível sintático e semântico das informações é diferente. Ao definir regras sintáticas de partilha, o conteúdo de todas as informações trocadas é normalizado de forma a ser possível determinar a sua natureza. Apenas com o uso dessas regras semânticas os sistemas existentes conseguem interpretar os dados, permitindo a sua ligação com outros sistemas e ao mesmo tempo a validação dos dados de acordo com as relações semânticas definidas. A validação das informações de acordo com as restrições estabelecidas é uma

das principais características que define o nível mais elevado de interoperabilidade. Estes conceitos de interoperabilidade são fundamentais para entender e explorar as particularidades da interoperabilidade, estes requisitos são definidos tanto pelo ambiente como por ferramentas nele implementadas [8].

2.4 Agentes Inteligentes para a Interoperabilidade

Existem já diversas tecnologias através das quais é possível desenvolver sistemas cujo objetivo é garantir a interoperabilidade entre SI. Essas metodologias são geralmente baseadas em arquiteturas de comunicação comuns e *standards* como o HL7. No entanto, ainda existem algumas preocupações sobre a distribuição, tolerância a falhas e *standards* de comunicação. A tecnologia multiagente têm se destacado nesta área, inclusivamente na área da interoperabilidade nas instituições de saúde, abordando as preocupações anteriormente mencionadas [1].

Esta tecnologia está intimamente relacionada com os conceitos fundamentais que definem uma arquitetura distribuída. A computação baseada em agentes tem se destacado devido à sua capacidade de resolver problemas e/ou apresentar uma nova revolução no desenvolvimento e análise de *software*. Embora não haja uma definição aceite para agente, pode ser dito que os agentes são entendidos como artefactos computacionais que apresentam certas propriedades [1].

Os agentes devem ser definidos para serem entidades de resolução de problemas de computação autónomos capazes de operarem de uma forma eficaz em ambientes dinâmicos. Eles são muitas vezes utilizados em ambientes nos quais interagem e cooperam com outros agentes.

O *software* baseado em agentes deve ser robusto, escalável e seguro. Para conseguir isso, as arquiteturas devem permitir que os agentes sejam compatíveis, de forma a comunicarem entre si e oferecerem serviços uns aos outros [1].

Sistemas multiagente MAS oferecem uma nova forma de desenvolvimento de sistemas complexos, especialmente em ambientes abertos e dinâmicos. Al-

gumas das principais características da tecnologia baseada em agentes apoia estas capacidades. A autonomia e pró-atividade são características que lhes permitem planear e executar tarefas definidas para atingir os objetivos propostos. As habilidades sociais permitem um agente interagir no MAS e cooperar para cumprir os seus objetivos. O MAS pode ser considerado como uma tecnologia rica e altamente adaptável com um grande interesse na área da interoperabilidade [1].

Para desenvolver estes sistemas é necessário métodos de especificação *standard*, e acredita-se que uma das características para a sua elevada aceitabilidade e recomendação é a simplicidade. Na verdade, o uso de agentes inteligentes para simular a tomada de decisão humana no campo médico oferece potencial para *software* adequado para o desenvolvimento, análise prática e construção de metodologias que não distinguem os agentes dos seres humanos. Estes sistemas podem tornar-se muito mais hábeis e eficazes para monitorizar o comportamento do seus próprios agentes, com um impacto significativo no processo de aquisição e validação dos conhecimentos, ou seja o MAS vem dar a conhecer o processo de evolução dos SI, e os elementos de substituição ou delegação de tarefas geralmente executadas por seres humanos [1].

O MAS é capaz de gerir o ciclo de vida do agente, a disponibilidade dos módulos dos SIH como um todo, mantendo-se todos os agentes livremente distribuídos. Novos agentes com as mesmas características e objetivos podem ser criados através do MAS, dependendo das necessidades do sistema em que está inserido. A estrutura desses agentes e do MAS pode ser desenvolvida de acordo com os serviços que prestam e da funcionalidade lógica dos sistemas que interagem com eles [1].

Os agentes numa unidade de saúde são úteis para configurarem aplicativos ou utilitários que recolhem informações na organização. Uma vez recolhidas, as informações podem ser fornecidas diretamente para outras entidades, por exemplo, para um médico ou para um servidor, armazenadas em arquivo ou enviadas por e-mail ou outros meios de comunicação [1]

2.5 Sistemas baseados em Agentes para a Interoperabilidade

Tal como foi referido anteriormente a tecnologia multiagente tem vindo a destacar-se no campo da interoperabilidade conferindo aos SI uma maior fiabilidade, flexibilidade, robustez e facilidade de manutenção e adaptabilidade. De seguida serão então apresentados alguns exemplos de soluções para implementar a interoperabilidade com o uso de agentes.

F. Marques [35] apresentou uma arquitetura de interoperabilidade desenvolvida para ser integrada em serviços governamentais. A arquitetura é baseada em agentes e serviços *web* onde o principal objetivo é providenciar uma forma segura de distribuir os serviços integrados aos clientes (cidadãos, empresas ou administração pública).

G. Acampora usou uma abordagem [36] baseada na integração de agentes colaborativos e algoritmos miméticos, para alinhar de forma eficiente ontologias e consequentemente resolver problemas de interoperabilidade semântica. Graças ao comportamento dos agentes colaborativos, o sistema multiagente (MAS) projetado é capaz de proporcionar diferentes benefícios tais como o paralelismo, a robustez, a escalabilidade e a eficácia.

M. Contreras apresenta a CAPNET [37] uma plataforma baseada em agentes desenvolvida em .NET que suporta a sua mobilidade e visa a criação de novos agentes, em diferentes máquinas de forma a executarem tarefas e comunicarem uns com os outros, garantindo assim a interoperabilidade no ambiente em que é implementada.

É necessário ainda salientar, que para além de sistemas baseados em agentes para garantir a interoperabilidade, existem já variadíssimas soluções para a criação desses sistemas multiagentes (MAS). Diferem geralmente na linguagem de programação, influenciando assim por vezes o sistema operativo no qual podem operar, podem ou não estar de acordo com as especificações da FIPA e ainda alguns deles fornecem interfaces atrativas com ambientes em 3 dimensões ou com capacidade de integrar sistemas de informação geográficos.

Também na área da saúde a tecnologia multiagente (MAS) já foi utilizada para garantir a interoperabilidade. Alguns exemplos serão de seguida

descritos salientando que existem de certo muitas mais soluções aqui não apresentadas.

B. Orgun [27] apresentou a implementação de um sistema baseado em ontologias e MAS como *framework* para as interações num ambiente hospitalar entre sistemas distribuídos resolvendo as limitações comuns da maior parte das aproximações cliente-servidor, onde o principal objetivo é facilitar o fluxo de informações do paciente numa organização de saúde.

Vários investigadores na área da interoperabilidade hospitalar têm recorrido à tecnologia *Java Agent DEvelopment Framework* (JADE), uma biblioteca desenvolvida em JAVA que permite a implementação de MAS de acordo com as especificações FIPA de forma a garantir a interoperabilidade e escalabilidade do ambiente heterogéneo como uma instituição de saúde. Alguns dos sistemas com essas características foram apresentados em vários estudos [6, 32, 38–40].

Outros sistemas baseados em agentes, também no domínio da saúde são apresentadas [41–47] como ferramentas de gestão da dados clínicos, sistemas de suporte à decisão e ainda sistemas que permitem uma monitorização remota do paciente.

Todos estes sistemas permitem então concluir que as tecnologias baseadas em agentes oferecem valores adicionais às abordagens clássicas de software, principalmente na área da saúde. Além disso esta tecnologia suporta a integração dos sistemas antigos, combatendo as deficiências dos sistemas centralizados, como aspetos que limitam a performance, limitações de recursos e diferentes tipos de falhas. É de salientar que a maioria dos trabalhos apresentados nesta área são apenas protótipos e não são utilizados em ambientes reais.

Capítulo 3

Materiais e Métodos

Neste capítulo são apresentadas as tecnologias utilizadas para o desenvolvimento deste projeto. É apresentada em primeiro lugar a tecnologia baseada em agentes, as suas características, o seu funcionamento em sistemas multiagentes (MAS) e a sua forma de comunicação através de uma linguagem comum e *standard* (ACL). A utilização de *sockets* para a transmissão segura dessas mensagens entre os agentes, que poderão encontra-se em diferentes máquinas ligadas a uma mesma rede de Internet, também se encontra devidamente descrita e ilustrada. A *framework .NET* e as principais classes utilizadas para desenvolver a plataforma são também neste capítulo apresentadas. Por fim é apresentado o *Pentaho Community* mais precisamente o editor CDE, uma ferramenta utilizada para a construção dos gráficos introduzidos na interface desenvolvida para o controlo dos agentes que constituem a plataforma.

O projeto descrito neste documento baseia-se na metodologia de investigação *action research*, caracterizada como uma pesquisa direcionada para a resolução de problemas de forma progressiva, ou seja, ao longo do desenvolvimento do trabalho, são analisados os resultados intermédios obtidos, de forma a realizar as ações necessárias para atingir os objetivos propostos [48].

3.1 Agentes Inteligentes

Um agente inteligente é uma entidade computacional com um comportamento autónomo que lhe permite decidir e executar as suas próprias ações. São programas desenvolvidos para funcionarem de forma contínua com a capacidade de intervir num determinado ambiente que muitas vezes é habitado por outros agentes, sem a necessidade de orientação ou intervenção humana constante. Os agentes são geralmente considerados artefactos computacionais usados em dispositivos de *software* ou *hardware* de forma a automatizar tarefas repetitivas, encontrar e filtrar informações, de uma forma inteligente resumir os dados complexos e especialmente integrar informações provenientes de fontes de informação heterogêneas entre outras ações [49–51].

Para além das características anteriormente descritas, um agente exhibe determinadas propriedades tais como [1, 52, 53]:

- **Autonomia** - capacidade de atuar sem uma intervenção direta do homem. Um agente toma a iniciativa e exerce controlo sobre as suas próprias ações das seguintes formas:
 - Orientado ao objetivo : o agente aceita pedidos de alto nível, indicando o que é que um humano quer e torna-se a entidade responsável por decidir de que forma é que vai satisfazer os pedidos.
 - Colaborativo : pode fazer objeções aos pedidos efetuados, modificar solicitações, fazer perguntas de esclarecimento, ou mesmo recusar-se a satisfazer certos pedidos.
 - Flexível : as ações não seguem um conjunto de regras, o agente é capaz de escolher dinamicamente quais as ações a executar, e qual a sequência a seguir, respondendo ao estado do ambiente externo.
 - Início automático : ao contrário dos programas padrão, chamados diretamente pelo utilizador, um agente pode perceber as mudanças do seu ambiente e decidir quando agir.
- **Reatividade** - capacidade de reação e de integração num ambiente de forma a responder em tempo útil às mudanças que nele ocorrem.

- **Pró-atividade** - capacidade de resolver problemas inteligentes e planejar as suas próprias atividades, a fim de atingir os seus objetivos. Os agentes não agem simplesmente em resposta ao seu ambiente, são capazes de exibir o seu comportamento orientado ao objetivo e tomar a iniciativa.
- **Comportamento Social** - capacidade de interagir com outros agentes podendo até mudar o comportamento desses agentes como resposta à interação. Um agente pode envolver-se com outros agentes, incluindo pessoas, obter informações ou recorrer a ajuda para alcançar os seus objetivos.
- **Continuidade temporal** - capacidade de execução contínua, um agente não é um processo de execução determinada que mapeia um início e um final e depois termina.
- **Adaptabilidade** - capacidade de personalização automática consoante as preferências dos seus utilizadores e com base em experiências anteriores e com mudanças ambientais.
- **Mobilidade** - capacidade de transporte de uma máquina para outra máquina e entre diferentes arquiteturas de sistemas e plataformas.

Os agentes podem ser classificados consoante o seu nível de inteligência. Os agentes com maiores capacidades de cognição são classificados como agentes cognitivos, enquanto aqueles com menos capacidades de cognição são considerados agentes reativos [49].

Nem todos os agentes são dotados de todas as capacidades anteriormente descritas, no entanto como é possível identificar na Figura 3.1 a autonomia, a reatividade, a pró-atividade e o comportamento social são as propriedades mais comuns.

Tal como foi referido anteriormente, um agente é um componente situado num determinado ambiente (Figura 3.1) que recebe estímulos e de um modo flexível e autónomo age de forma a atingir os seus objetivos. Assim sendo, a complexidade de um agente está relacionada também com as propriedades do ambiente [15].

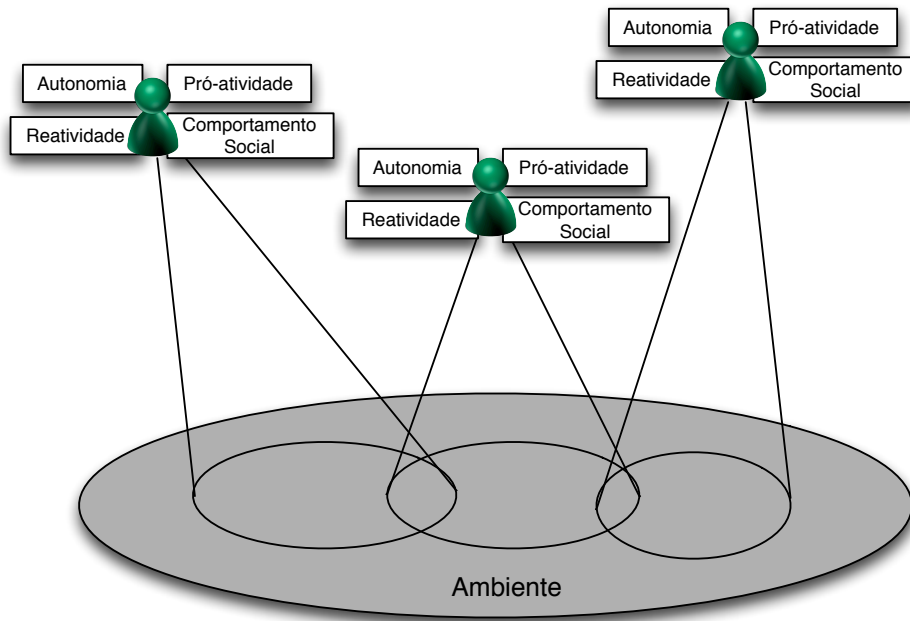


Figura 3.1: Propriedades dos Agentes e a sua interação com o ambiente.

Sempre que a resolução de um problema requer o esforço de dois ou mais agentes, é considerado um sistema multiagente (MAS). Um MAS pode ser definido como uma rede de entidades que trabalham em conjunto de forma a resolver um problema que se encontra para além das capacidades de resolução individuais e do conhecimento de um agente. Embora os agentes possam habitar num ambiente comum, cada um deles tem uma informação parcial ou uma capacidade de resolução de problemas, assim sendo é necessário garantir a autonomia de cada um deles, isto é cada um desses agentes deverá ter os seus próprios objetivos, e características específicas [15, 49].

Um MAS é simplesmente um grupo de agentes com as propriedades anteriormente mencionadas que se comunicam. O principal objetivo de um MAS pode ser alcançado através da cooperação e habilidade de coordenação dos agentes que o compõem, combinada com regras de comunicação bem definidas [15, 49].

A comunicação constitui assim a base de um MAS, contudo é necessária uma linguagem que seja compreensível para todos os agentes, a fim de

trocarem informações de uma forma eficaz [15, 49].

Tabela 3.1: Parâmetros das mensagens **FIPA-ACL** (Adaptado de [12, 15]).

Parâmetro	Categoria	Descrição
<i>Performative</i>	Tipo de ato comunicativo	Tipo de mensagem ACL
<i>Sender</i>	Participantes na comunicação	Remetente da mensagem
<i>Receiver</i>		Recetor da mensagem
<i>reply-to</i>		Recetor das mensagens subsequentes neste segmento de conversação
<i>Content</i>	Descrição do conteúdo	Conteúdo da mensagem
<i>Language encoding</i>		Linguagem em que o conteúdo é expresso
<i>ontology</i>		Método pelo qual o conteúdo é codificado
<i>protocol</i>		Ontologia(s) utilizadas para apoiar a interpretação do conteúdo
<i>protocol</i>		Protocolo de integração no qual a mensagem ACL é gerada
<i>conversation-id</i>	Controlo da conversação	Identificador utilizado para identificar a sequência contínua de atos comunicativos que em conjunto formam uma conversa
<i>reply-with</i>		Expressão utilizada para que o agente que responde a esta mensagem possa identificar a mensagem
<i>in-reply-to</i>		Uma expressão que se refere a uma ação anterior a esta mensagem
<i>reply-by</i>		Tempo de resposta ou data em que o agente emissor gostaria de receber uma resposta

Uma linguagem de comunicação entre agentes (**ACL**) foi então proposta pela **FIPA** fornecendo um conjunto de primitivas de linguagem para implementar o modelo de comunicação entre os agentes. O **FIPA-ACL** é um conjunto de especificações [12] que abrange [54]: a estrutura de uma mensagem **ACL** que pode ser utilizada pelos agentes para construir as suas mensagens; um conjunto de atos comunicativos (primitivas de linguagem) que especifica os tipos de mensagens **ACL**; um conjunto de protocolos de comunicação que suportam a interação e a troca de mensagens. Por outras palavras, é um conjunto de especificações para ajudar a garantir a interoperabilidade entre agentes e serviços heterogéneos. Assim sendo, define o envelope da mensagem e dos protocolos, sendo o conteúdo da mensagem li-

vre, ou seja pode ser representado em qualquer linguagem e interpretado consoante o seu desenvolvimento.

O envelope da mensagem **FIPA-ACL** é construído utilizando um conjunto de parâmetros especificados na Tabela 3.1, tais como o tipo de mensagem, o emissor e o recetor, o identificador da conversação, etc. O conteúdo da mensagem é expresso através de uma linguagem de conteúdo, que tal como foi referido anteriormente é de livre escolha por parte de quem desenvolve o sistema [15].

Nem todos estes parâmetros são necessários para uma comunicação eficaz entre os agentes, o único parâmetro obrigatório em todas as mensagens **ACL** é o *Performative* (tipo de ato comunicativo) contudo a maior parte das mensagens também contêm o *Sender* (Emissor), o *Receiver* (Recetor) e o *Content* (conteúdo). Como exemplo é apresentada a Figura 3.2 que se trata de uma mensagem informativa do agente_x para o agente_y.

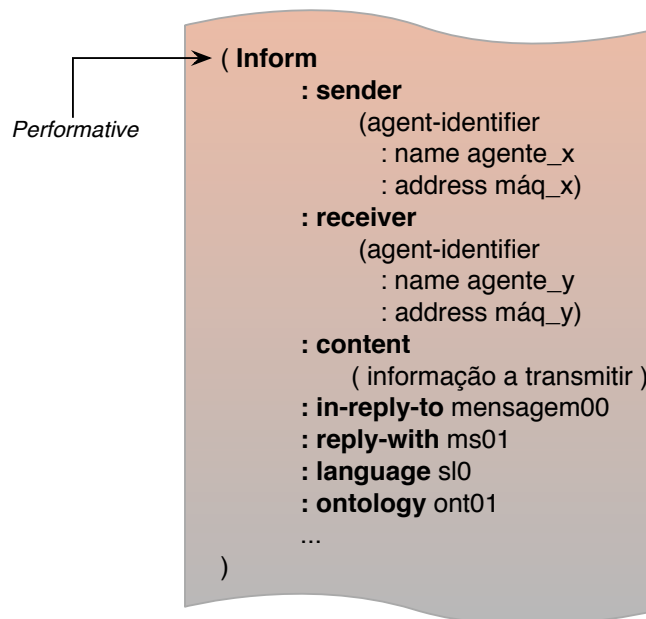


Figura 3.2: Exemplo de mensagem **ACL**.

A especificação **FIPA-ACL** define uma biblioteca de atos comunicativos (*Performatives*) composta por 22 tipos de atos comunicativos. Esta biblio-

teca encontra-se representada no Apêndice B.1 nas Tabelas B.1 e B.2.

Tal como já foi referido, um agente raramente é útil por si só. Normalmente, os agentes terão de coordenar e cooperar com outros de forma a resolver um determinado problema mais ou menos complexo. A cooperação baseia-se na capacidade de um agente comunicar com outros. Na maioria das plataformas de agentes existentes a forma mais usual de comunicação é por meio do conjunto de protocolos TCP/IP [55].

Sockets TCP/IP

Em todo mundo, diversos dispositivos contêm ligação a uma rede mundial designada de Internet. Numa visão mais técnica pode referir-se que a Internet permite que diversos programas se comuniquem e troquem informações, mesmo que se encontrem a milhares de quilómetros de distância e mesmo que não exista uma ligação física entre eles, sendo somente necessário que ambos os dispositivos se encontrem conectados a uma rede de Internet. A maioria desses programas utiliza a interface de programação *socket* para obter acesso aos serviços de comunicação na rede [56].

A interface *Sockets Berkeley*, conhecida universalmente como *socket* genericamente é uma interface para programação de aplicações *standard* para a rede, que emerge como uma tecnologia que abrange uma ampla gama de sistemas operativos. Foi projetada para proporcionar o acesso a serviços genéricos de comunicação entre processos que podem ser aplicados por qualquer protocolo [56].

É necessário então perceber que um canal de comunicação é um meio de transmissão de sequências de *bytes* de uma máquina para outra, podendo utilizar diversas tecnologias de transmissão (ex: *Ethernet*). Essa sequência de *bytes* é geralmente designada de pacote. Um pacote contém informações de controlo que a rede usa para a comunicação e por vezes também contém informação do utilizador. Um protocolo é então um acordo sobre os pacotes trocados na comunicação entre os programas. Especifica como é que os pacotes são estruturados, onde é que se encontram as informações do destinatário, o tamanho do pacote e também a forma como a informação deve ser

interpretada [56].

O **TCP/IP** constitui um conjunto de soluções, muitas vezes designado conjunto de protocolos usado na rede de Internet constituído essencialmente pelo Protocolo de Internet (**IP**), pelo Protocolo de Controlo de Transmissão (**TCP**) e pelo Protocolo de Datagramas de Utilizador (*User Datagram Protocol* (**UDP**)). Estes protocolos têm a função de passar dados de uma aplicação/máquina para outra de forma a manter intacta a integridade dos dados transferidos [56].

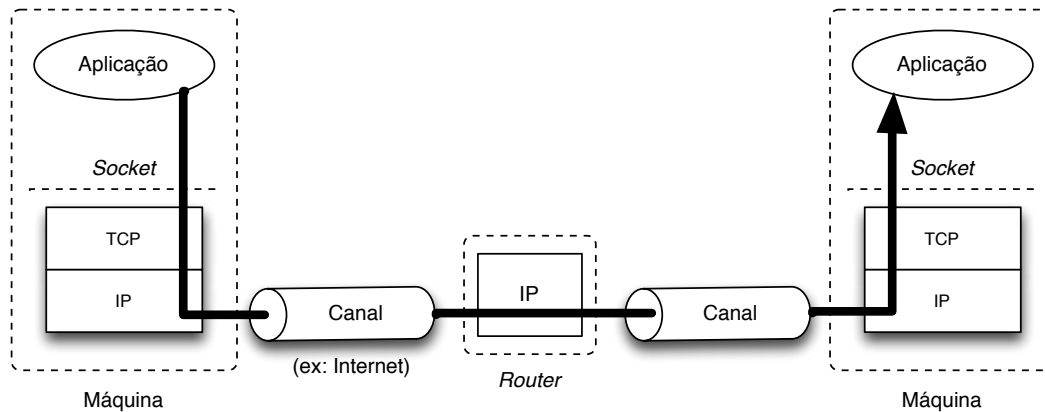


Figura 3.3: Exemplo de transferência de dados utilizado *sockets* TCP/IP (Adaptado de [56]).

A figura 3.3 mostra a relação entre os protocolos, as aplicações/máquinas e os *sockets*, bem como o fluxo de dados de uma aplicação para outra. Assim sendo, um *socket* corresponde a uma interface entre a camada de aplicação e a camada de transporte. Dessa forma um processo recorre ao *socket* para enviar mensagens e ao enviar a mensagem, o processo recorre à infraestrutura de transporte **TCP** controlada pelo sistema operativo que vai transferir a mensagem de uma forma fiável recorrendo à Internet. A máquina de destino também necessita de um *socket* utilizado para o processo de receção de mensagens de forma análoga à anteriormente descrita.

Neste tipo de comunicações são muito utilizados os termos de cliente e de servidor. O programa cliente inicia a comunicação, enquanto o programa ser-

vidor aguarda passivamente e, em seguida responde a clientes que entraram em contacto com ele. Para uma melhor compreensão é apresentada a Figura 3.4 que descreve detalhadamente a aplicação desta arquitetura. Seguindo a Figura é possível verificar que em primeiro lugar o servidor cria um *socket* com um determinado endereço de IP conhecido pelo cliente numa determinada porta e aguarda por conexões por parte de um cliente. Por sua vez, o cliente também cria um socket com um determinado endereço de IP que não necessita de ser conhecido por parte do servidor e solicita a conexão do seu *socket* ao *socket* do servidor e aguarda que a conexão seja estabelecida. O servidor aceita a conexão, nesta fase o cliente envia uma mensagem que o servidor recebe, e envia a resposta ao cliente. Por fim ambos fecham a conexão.

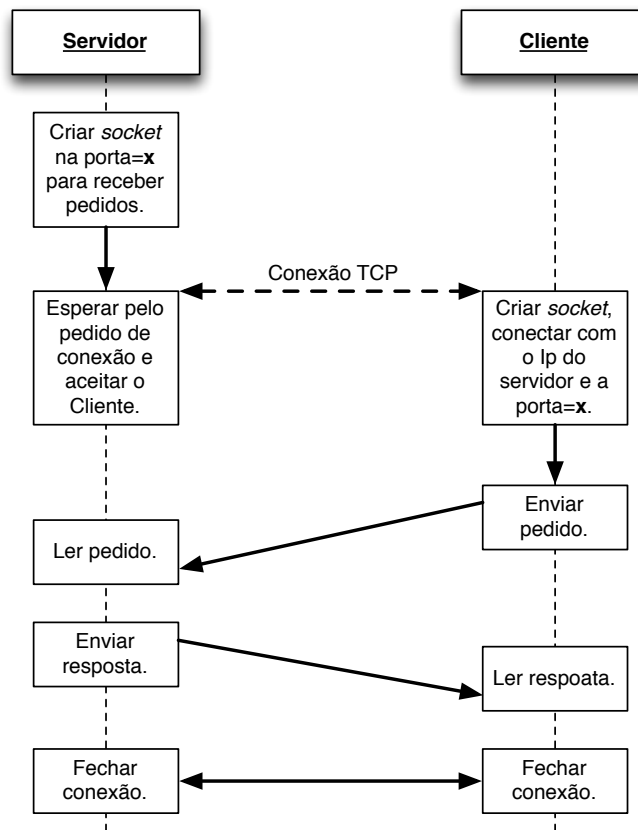


Figura 3.4: Arquitetura cliente servidor recorrendo a *sockets* (Adaptado de [57]).

3.2 .NET

A *framework* .NET é um modelo de programação abrangente e consistente da *Microsoft* para criar aplicações que têm interfaces de utilizador visualmente atrativas, comunicações totalmente integradas e seguras e a capacidade de modelar um intervalo de processos de negócio. É uma tecnologia que suporta a construção e execução de aplicações e serviços *web*. Esta plataforma oferece vantagens como a capacidade de desenvolver aplicações multiplataforma, gestão de recursos automática e a simplificação da implementação da aplicação desenvolvida. Como a segurança é uma parte essencial, a .NET fornece suporte de segurança, formas de verificar a autenticidade do código,

recursos para autorização de acessos, segurança declarativa e imperativa e métodos de segurança criptográfica para introduzir na aplicação. Fornece um modelo orientado a objetos simples para aceder à maioria das interfaces de programação de aplicações *Windows* (APIs), mecanismos através dos quais é possível utilizar código nativo existente. Para além disso, estende a plataforma de desenvolvimento de forma a fornecer ferramentas e tecnologias para desenvolver aplicações distribuídas baseadas em Internet [58].

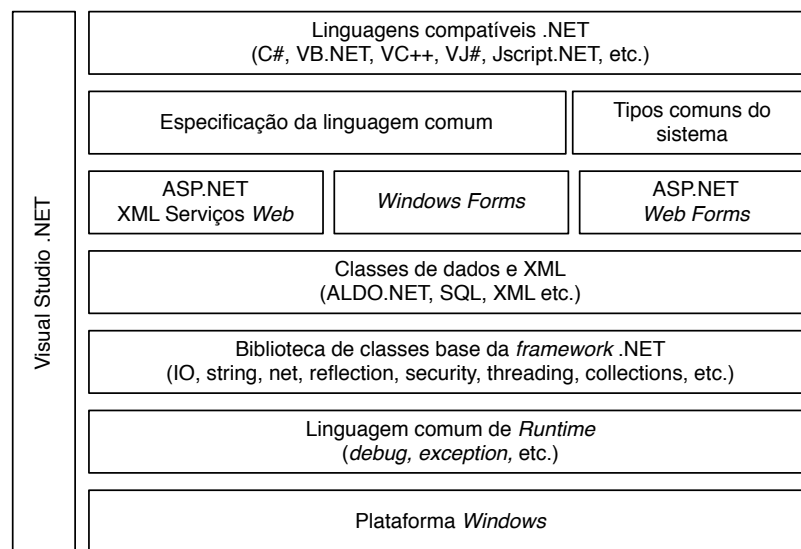


Figura 3.5: Camadas da Plataforma .NET (Adaptado de [55, 58]).

A plataforma .NET é um ambiente de execução de aplicações em várias linguagens de programação que de uma forma transparente gere serviços com infraestruturas centrais. Constitui um conjunto de múltiplas linguagens/tecnologias utilizadas para o desenvolvimento e criação de *web Forms*, Serviços *web* e aplicações *Windows*. O ambiente é composto por várias camadas, apresentadas esquematicamente na Figura 3.5 [58].

No topo da estrutura surgem as linguagens compatíveis com esta plataforma bem como a especificação da linguagem comum e um *standard* para definir os tipos de dados que são usados e declarados. Na terceira camada encontram-se os componentes de nível superior, os Serviços *web*, as *Windows Forms* e as *web Forms*. Em seguida surge uma camada dedicada à utiliza-

ção de uma série de classes para a manipulação de dados, usando bases de dados ou XML seguindo-se da camada que contém as classes base que oferecem funcionalidades *standard* necessárias para a maioria dos programas. Por fim encontra-se provavelmente a camada mais importante que assenta diretamente na plataforma *Windows* e é denominada de linguagem comum de *Runtime*, do inglês *Common Language Runtime (CLR)*. O CLR proporciona o ambiente no qual os programas são carregados e executados, gere a segurança das aplicações, a memória, os processos e os *threads* [55, 58].

A plataforma .NET providencia uma biblioteca de classes que fornece ao utilizadores acesso a determinadas funcionalidades do sistema. O *Namespace System* é o mais utilizado, contudo só serão descritas as classes utilizadas neste projeto.

3.2.1 *System.Data.OracleClient*

O *Namespace System.Data.OracleClient* contém classes para o acesso e gestão de dados provenientes de uma base de dados do tipo *Oracle*. Permite ao utilizador conectar-se à base de dados através da introdução das suas credenciais de acesso, aceder à informação pretendida, e até fazer uma gestão, alteração ou introdução de informação [59].

3.2.2 *System.Net*

Os *Namespace*s pertencentes ao *System.Net* contém classes que fornecem uma interface de programação simples para vários protocolos da internet. A classe *Dns* fornece funcionalidades simples de resolução de nomes de domínios, permite ao utilizador solicitar informações sobre o endereço de IP da máquina onde se encontra bem como o nome do *host* da mesma [60].

O *Namespace System.Net.Sockets* fornece ao utilizador uma implementação da interface do *Windows Sockets* para programadores que necessitam de controlar o acesso à rede. As classes *TcpListener* e *TcpClient* possibilitam a criação de conexões TCP com a Internet, permitindo dessa forma que haja uma troca de mensagens através dessas conexões [61].

3.2.3 *System.Threading*

Os *Namespaces* pertencentes ao *System.Threading* contêm classes que permitem a programação *multithread*. Nomeadamente a classe *thread* permite que o utilizador crie e controlo um *thread*, defina a sua prioridade e obtenha o seu estado [62].

3.2.4 *System.Xml*

O *System.Xml* é um *Namespace* responsável pelo suporte e processamento de XML. Permite ao utilizador criar e editar documentos XML, ler e interpretar a sua estrutura de dados. Assim, de uma forma simples, o utilizador poderá manipular informação e guarda-la de uma forma estruturada num documento XML [63].

3.2.5 *System.Diagnostics*

Os *Namespaces* pertencentes ao *System.Diagnostics* contêm classes que permitem interagir com processos do sistema, *logs* de eventos e medidores de desempenho. A classe *Process* fornece funcionalidade para monitorizar processos do sistema através da rede, bem como iniciar e interromper processos do sistema local. Além de fornecer listas de processos em execução (especificando o computador, o nome do processo ou o seu identificador) ou exibir informações sobre o processo é possível obter um conhecimento detalhado dos módulos do processo [64].

3.2.6 *System.Timers*

O *Namespace System.Timers* fornece o componente *Timer*, que permite acionar um evento num determinado intervalo de tempo especificado. Isto é o utilizador, especifica um intervalo de tempo no qual quer que uma determinada atividade seja executada periodicamente [65].

3.3 *Pentaho Community*

O *Pentaho Community* foi criado em 2004 e constitui uma ferramenta de *Business Intelligence* desenhada para proporcionar a integração de dados, criação de relatórios, gráficos e *dashboards* baseados em tecnologias de processamento de dados. *Pentaho Community* contém diversos produtos e extensões, assim sendo vai ser apresentado somente o editor utilizado neste trabalho, o CDE.

O CDE foi criado para simplificar os processos de criação, edição e interpretação de *dashboards*. É uma ferramenta muito poderosa e completa, combinando a interface visual com fontes de dados e componentes personalizados de forma a criar um *layout* personalizado e adicionar e configurar diferentes componentes como caixas de texto, tabelas e gráficos. O *layout* de uma *dashboard* pode ser criado por uma simples combinação de linhas e colunas, juntamente com alguns blocos HTML e possivelmente algumas imagens. A origem dos dados que constituem o *dashboard* também pode ser definida pelo utilizador desta ferramenta, basicamente é necessário especificar de onde vêm os dados e a forma como o utilizador pretende que eles sejam apresentados. Desta forma o utilizador pode criar as suas *dashboards* dinâmicas que poderão combinar informações provenientes de várias fontes de dados de forma a auxiliar e facilitar a análise dos mesmos [66].

Capítulo 4

Plataforma baseada em Agentes para a Interoperabilidade

Neste capítulo, em primeiro lugar irá ser apresentada a plataforma **AIDA** implementada no **CHP**. Todas as suas características são descritas, bem como a sua arquitetura e os seus componentes. É feito o levantamento das principais vantagens e desvantagens do sistema **AIDA** de forma a responder à **Questão 1**. Posteriormente é apresentada a **BMaPI** desenvolvida ao longo deste projeto, como um sistema que vem melhorar a performance global da plataforma **AIDA** e de certa forma eliminar ou enfatizar uma limitação deste sistema. A sua arquitetura, funcionalidades e a descrição detalhada dos seus componentes são apresentados neste capítulo, fazendo-se acompanhar de esquemas e imagens intuitivas para facilitar a compreensão dos mesmos. Finalmente é descrita a implementação da **BMaPI** na **AIDA** do **CHP**.

4.1 Plataforma AIDA

Técnicas baseadas em Inteligência Artificial têm demonstrado um elevado potencial quando introduzidas em soluções aplicadas em ambientes hospitalares. Na verdade, muitas dessas soluções encontram-se já em produção em várias instituições de saúde portuguesas centradas nas áreas dos sistemas de integração e nos sistemas de suporte à decisão. De uma forma mais específica

uma solução desenvolvida por um grupo de inteligência artificial da Universidade do Minho denominada de **AIDA** [11] corresponde já à unidade central de interoperabilidade em várias organizações de grande dimensão como o **CHP**, o Centro Hospitalar do Tâmega e Sousa, Centro Hospitalar do Alto Ave e Unidade local de Saúde do Norte Alentejano. Esta plataforma baseada no paradigma orientado a agentes tem demonstrado uma grande adaptabilidade, modularidade e efetividade através da utilização de um **MAS** básico que tem crescido de acordo com as necessidades particulares de cada instituição.

A **AIDA** foi desenhada para auxiliar as aplicações médicas sob a forma de uma rede de sistemas inteligentes de processamento de informação, os seus principais subsistemas, o seu papel funcional, e o controlo do fluxo de informação com um nível de autonomia ajustável. Foi desenvolvida para permitir a difusão e integração de informações geradas em ambiente hospitalar. É um sistema complexo, composto por subsistemas especializados entendidos como entidades inteligentes e flexíveis, neste trabalho designados como agentes inteligentes, encarregados de tarefas como a comunicação entre sistemas heterogêneos, envio e receção de informações (ex: relatórios clínicos, imagens, conjuntos de dados, prescrições, etc.), gestão e armazenamento da informação e resposta a pedidos de uma forma correta e atempada [1, 11, 20, 24].

Esta plataforma contém muitos recursos de integração diferentes, nomeadamente tecnologias como **SOA** e **MAS** para a implementação da interoperabilidade num ambiente distribuído e específico. O principal objetivo deste sistema é, tal como o nome indica, integrar, divulgar e arquivar grandes conjuntos de dados provenientes de diversas fontes (ex: serviços, departamentos, unidades, computadores e equipamentos médicos, etc.). Fornece também ferramentas para implementar e facilitar a comunicação com os seres humanos através de serviços baseados na *web*.

Para além das características anteriormente mencionadas, em ambiente hospitalar a plataforma **AIDA** tem-se destacado como o elemento central, como é possível verificar na Figura 4.1, que garante essencialmente a interoperabilidade e a comunicação entre os seguintes sistemas [1, 3, 7, 14]:

- O **Processo Clínico Eletrónico (PCE)**, constitui um repositório de informação sobre a história clínica de um paciente que foi sujeito a cuidados

de saúde, num formato que pode ser processado pelo computador, armazenado e transmitido, a partir de um sistema seguro e acessível para vários utilizadores autorizados.

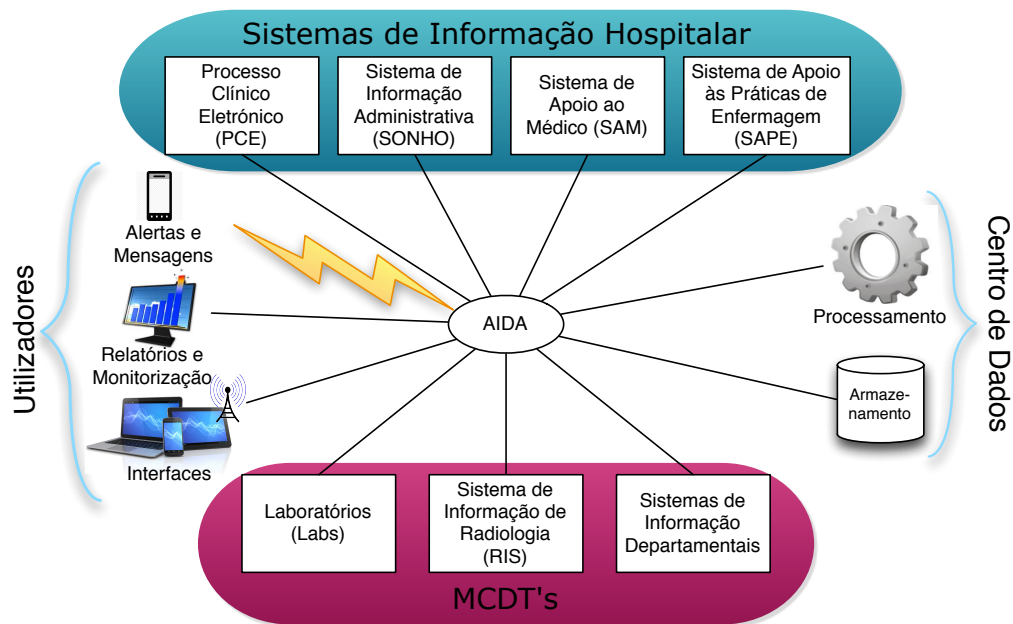


Figura 4.1: Arquitetura da AIDA (Adaptado de [3, 7, 14]).

- O Sistema de Informação Administrativa, em Portugal designado SONHO visa representar, gerir e arquivar as informações administrativas durante um episódio (recolha de todas as operações atribuídas a um determinado paciente desde o início até ao final do tratamento).
- O SAM que pretende representar, gerir e arquivar informação clínica durante o episódio.
- O SAPE que pretende representar, gerir e arquivar informação sobre as práticas de enfermagem durante o episódio.
- Os sistemas de Informação de todos os serviços e sistemas do hospital, em particular os Laboratórios e os serviços de imagiologia e radiologia

que constituem os Meios Complementares de Diagnóstico e Terapêutica (MCDT).

Esta arquitetura permite que toda a informação seja reunida, armazenada e distribuída automaticamente de forma a poder estar disponível para as autoridades autorizadas, quando e onde elas necessitarem da informação. Tal como foi referido anteriormente, toda esta arquitetura é assegurada pela ação e cooperação de vários subsistemas designados de agentes inteligentes. AIDA constitui portanto um MAS puramente comunicativo, ou seja o ambiente não tem qualquer influência na ação de cada agente, os agentes apenas comunicam entre si através de mensagens. Para além disso é importante referir que a plataforma AIDA tem diferentes tipos de agentes representados na Figura 4.2 e de seguida, brevemente definidos [1, 11, 14]:

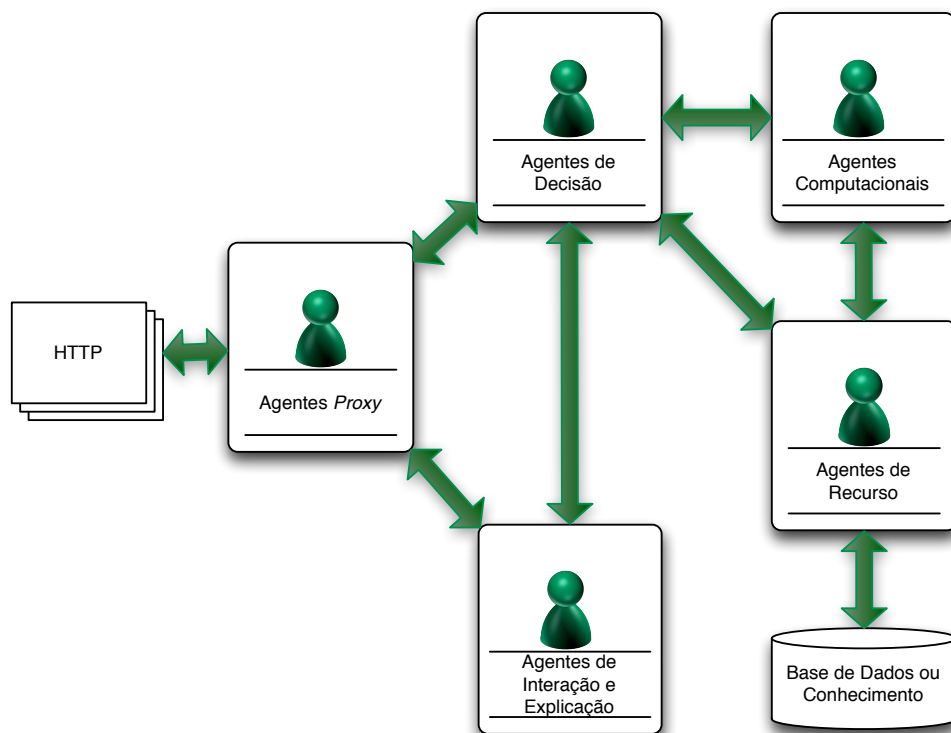


Figura 4.2: Tipos de agentes que constituem a Plataforma AIDA (Adaptado de [11, 14]).

- **Agentes *Proxy*:** Fornecem a ponte entre os utilizadores e o sistema através de questões que o utilizador possa fazer ao sistema, explicações que o sistema possa fornecer, ajudas na tomada de decisão ou simplesmente visualização de dados e resultados de pesquisas realizadas.
- **Agentes de Decisão:** Fornecem a capacidade de intermediário, aceitam ou não tarefas requeridas pelos Agentes *Proxy*, e requerem muitas vezes o auxílio dos Agentes Computacionais para processar certas tarefas. Posteriormente integram os resultados recebidos pelos Agentes Computacionais.
- **Agentes Computacionais:** Aceitam pedidos de tarefas específicas requeridas por parte dos Agentes de Decisão, retornando o respetivo resultado.
- **Agentes de Recurso:** Fornecem todo o conhecimento necessário para aceder a um recurso de dados específico.
- **Agentes de Interação e Explicação:** Funcionam com base em processos argumentativos que são alimentados com dados e/ou conhecimento, tanto dos Agentes *Proxy* como dos Agentes de Decisão. Note-se que os planos de execução recebidos pelos Agentes de Decisão pode ser parcial, de modo a que só após a conclusão de uma tarefa seja possível criar uma explicação e entregar aos Agentes *Proxy* ou aos Agentes de Decisão.

Visto que a plataforma [AIDA](#) tem vindo a tomar um papel vital no funcionamento normal dos [SIH](#) nas instituições onde se encontra implementada e nomeadamente no [CHP](#), é muito importante garantir que ela oferece as melhores funcionalidades e que os utilizadores diretamente relacionados com o seu funcionamento estão satisfeitos com a sua performance. Para isso, foram levantadas as principais vantagens e desvantagens da plataforma [AIDA](#) como forma de resposta à [Questão 1](#), apresentadas na tabela [4.1](#) e posteriormente discutidas.

É então possível constatar que para além de todas as características anteriormente apresentadas, a plataforma [AIDA](#) é dotada de muitos pontos

Tabela 4.1: Vantagens e Desvantagens da Plataforma [AIDA](#) implementada no [CHP](#).

Vantagens	Desvantagens
Elevado poder de gestão de mudanças no sistema	Falta de documentação sobre o sistema
Capacidade de personalização de objetos	Necessidade de uma elevada Educação e formação dos profissionais
Alta disponibilidade, acessibilidade e suporte em tempo útil	Recursos físicos antigos, lentos e limitados
Elevada segurança da informação e do sistema	Inexistência de um sistema de recuperação em caso de catástrofe
Tecnologicamente moderno	Dificuldade no controlo dos agentes e respetivas atividades
Facilidade de manutenção e de uso	
Simplicidade do sistema	

positivos, e apesar de também conter alguns pontos negativos, estes têm uma influência muito reduzida no bom funcionamento do sistema global. Como principal vantagem destaca-se a elevada disponibilidade, acessibilidade e suporte em tempo útil, uma plataforma de interoperabilidade entre [SIH](#) necessita mesmo destas propriedades, tem de estar sempre disponível, o acesso aos seus serviços tem de ser assegurado e o suporte por parte de especialistas na área tem de ser garantido. A segurança é também uma temática importante nesta área e neste ambiente particular, a informação sobre cada indivíduo tem de estar segura e respeitar certas normas éticas e legais. O sistema garante a integridade, confidencialidade e disponibilidade da informação. A facilidade de manutenção de uso e de gestão é também um ponto importante pois a maioria dos profissionais de saúde tem pouca formação na área das ciências da computação e é necessário garantir que os profissionais de saúde não perdem muito tempo a aceder ou registar informação, porque esse tempo pode ser essencial no tratamento/diagnóstico do paciente, daí surge a ideologia da [AIDA](#) baseada na simplicidade.

Relativamente às falhas é possível referir que apesar de não influenciar diretamente as funcionalidades do sistema, uma completa documentação do sistema seria muito útil para os seus utilizadores, gestores e mesmo para os membros que integram a equipa de suporte ao sistema. Essa documentação poderia também reduzir a necessidade de uma educação e formação dos profissionais, ou simplesmente basear essa educação na documentação sobre o sistema. Com o crescente aumento tecnológico, e com as dificuldades financeiras do País e do elevado custo de aquisição de novos equipamentos é difícil a substituição dos recursos físicos, nos quais a plataforma desenvolve as suas atividades, limitando assim também o sistema. Seria também importante a existência de um sistema de recuperação em caso de catástrofe contudo também seria necessário um elevado investimento. Por fim, surge como desvantagem da plataforma *AIDA* a inexistência de um sistema de controlo dos agentes presentes neste *MAS*. Deste aspeto deriva a necessidade de desenvolver um sistema capaz de controlar esses agentes. Assim sendo, com este trabalho pretende-se contornar e eliminar esta desvantagem desenvolvendo e implementando uma plataforma baseada em agentes de interoperabilidade que tem como principal objetivo resolver esta necessidade de haver um maior controlo sobre os agentes e a monitorização da sua pró-atividade.

4.2 Arquitetura e Implementação

De forma a contornar esta desvantagem da *AIDA* e ao mesmo tempo dar resposta à *Questão 2*, projetou-se a *BMaPI*, um sistema que permite controlar e administrar uma comunidade de agentes, permitindo a sua sobrevivência num ambiente heterogéneo. Nomeadamente uma plataforma que integre todos os agentes desse ambiente, independentemente da máquina onde estes executam a sua atividade, garantindo que para além de todos se comunicarem é criado um ambiente interoperável. Para facilitar ainda mais esse controlo sobre os agentes, foi desenvolvida uma interface gráfica, para que o utilizador possa gerir, de uma forma simples e atrativa, os agentes que constituem o sistema e ainda as respetivas atividades.

Numa primeira fase foi necessário fazer um levantamento das principais

funcionalidades da BMaPI, quais tecnologias que melhor se adaptam, e qual a melhor arquitetura para a desenvolver.

Tendo em conta as necessidades anteriormente descritas e aproveitando para responder à **Questão 3** pretende-se que a BMaPI, plataforma desenvolvida para integrar a AIDA implementada no CHP permita:

- Garantir um maior controlo sobre os agentes que constituem a AIDA.
- Facilitar ao utilizador a criação e registo de novos agentes localmente ou remotamente.
- Permitir que o utilizador ative e desative serviços, através do lançamento ou paragem de um determinado agente.
- Facilitar a programação e reprogramação da atividade dos agentes.
- Monitorizar a atividade dos agentes.

Para a programação de todo o sistema utilizou-se a *framework* .NET, a linguagem C# e diversas classes disponíveis para esta *framework*, de forma a manter a coerência do ambiente do CHP, que utiliza maioritariamente tecnologias baseadas no sistema operativo *Windows*.

Várias Plataformas de agentes têm sido desenvolvidas com especial atenção às temáticas da interoperabilidade e da compatibilidade. Nesse sentido, o modelo de referência FIPA tem se destacado como um *standard* para a programação orientada a agentes, que entre as demais especificações criou uma para padronizar a comunicação entre os agentes de forma a garantir as suas capacidades sociais. Essa especificação, denominada FIPA-ACL foi utilizada no desenvolvimento desta plataforma para garantir a uniformização das mensagens trocadas entre os agentes.

Desta forma é possível então responder à **Questão 4** referindo que os agentes vão comunicar-se através de mensagens ACL, contudo é necessário saber ainda como é que essas mensagens vão ser transportadas entre os agentes que poderão encontrar-se em diferentes máquinas.

No CHP todas as máquinas que suportam agentes encontram-se conectadas a uma mesma rede de Internet, assim sendo é possível recorrer aos

protocolos **TCP/IP**, à utilização de *sockets* e aos termos cliente e servidor, para garantir a transmissão correta das mensagens entre os agentes.

Por fim foi também utilizada uma ferramenta designada *Pentaho Community* para possibilitar a construção de alguns gráficos de auxílio para facilitar aos utilizadores do sistema o controlo da pró-atividade dos agentes que constituem a plataforma.

Todos estes protocolos, tecnologias e ferramentas já foram devidamente explicadas e ilustradas no Capítulo 3, ainda assim todo o sistema da **BMaPI** será descrito e acompanhado de esquemas para tornar mais simples a compreensão do mesmo.

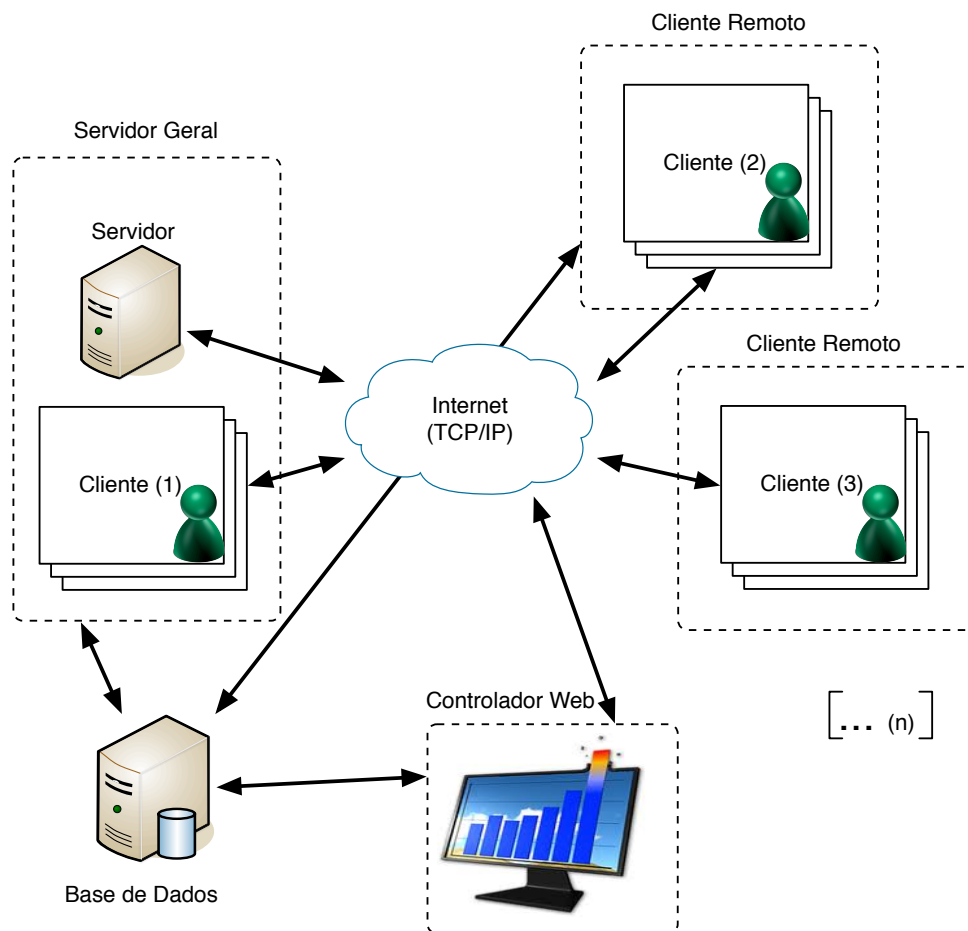


Figura 4.3: Arquitetura da BMaPI.

Começando pela arquitetura geral da BMaPI apresentada na Figura 4.3, é possível verificar que existem três componentes principais. O programa **Servidor Geral** instalado numa máquina que servirá como o próprio nome indica de servidor, essa mesma máquina também terá pelo menos um cliente, o Agente Monitor daqui em diante representado por *Agent Management System* (AMS) responsável por supervisionar o acesso e o registo dos agentes desta máquina na plataforma. O programa **Cliente Remoto** instalado nas restantes máquinas, que terá também pelo menos um cliente, o AMS responsável pelos agentes de cada uma dessas máquinas. É de salientar que cada agente corresponde a um cliente e que os agentes instalados na máquina principal são controlados pelo programa Servidor Geral e aqueles que se encontrem em outras máquinas serão controlados pelo programa Cliente Remoto que nelas se encontra.

Para além destes dois programas relativamente semelhantes existe ainda o **Controlador web** que permite que o utilizador, controle quais os agentes que se encontram em cada máquina, aceda a todas as informações de cada agente de uma forma atrativa, agende as atividades dos agentes e ainda que consulte e monitorize a pró-atividade dos mesmos.

Todos estes programas fazem uso de uma base de dados para manter a informação dos agentes segura, disponível e atualizada.

4.2.1 Descrição dos componentes

De seguida são descritos de forma mais detalhada e técnica os diversos componentes que compõem a BMaPI. Tal como foi definido, anteriormente existem três componentes principais, contudo o Servidor Geral e o Cliente Remoto são dois componentes bastante semelhantes e portanto irão ser apresentados em conjunto sendo as suas diferenças destacadas ao longo dessa apresentação.

Servidor Geral e Cliente Remoto

Estes dois programas desenvolvidos na totalidade recorrendo à *framework* .NET, baseiam-se em duas *Windows Forms* distintas que funcionam como

interface Sistema-Utilizador, contudo o seu aspeto visual é muito semelhante de forma a uniformizar a BMaPI. A interface relativa ao programa Servidor Geral encontra-se apresentada na Figura 4.4, e é através dela que toda a plataforma é iniciada de uma forma automática, oferecendo também ao utilizador uma secção para a criação de novos agentes apresentada na Figura 4.5.

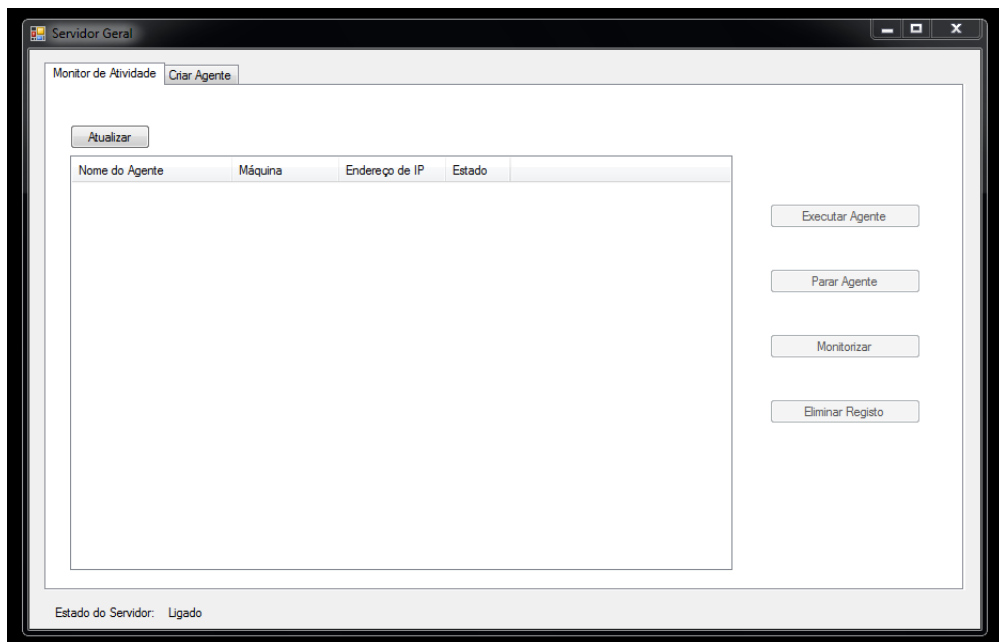


Figura 4.4: Interface do programa Servidor Geral.

Para além disso, permite que utilizador solicite ao agente a sua execução, a interrupção da sua atividade e também eliminação o seu registo. E apesar de não ter funções de gestão, nem de monitorização, esta interface permite ainda ao utilizador consultar algumas informações relativas à BMaPI, como por exemplo os agentes que se encontram registados e a lista de atividades executadas por cada um deles.

Toda a informação que diz respeito aos agentes, a sua identidade e informação relativa à atividade que realizam encontra-se armazenada numa base de dados *Oracle* que confere à BMaPI uma maior robustez, segurança dos dados, confiança, escalabilidade e organização. Permite ainda que os dados sejam editados, atualizados e consultados por diversos utilizadores ao mesmo

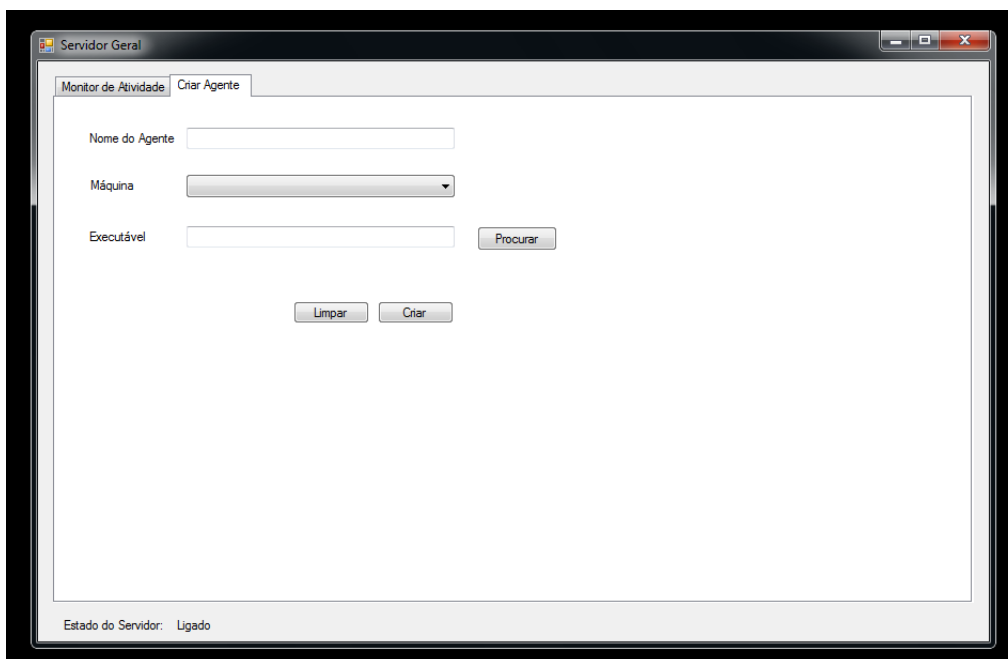


Figura 4.5: Interface do programa Servidor Geral para a criação de novos agentes.

tempo garantindo dessa forma a disponibilidade de toda a informação armazenada nesta estrutura. Ainda assim, só os utilizadores autorizados poderão manipular a informação garantindo também a integridade da mesma. Para além dessas características, a *framework .NET* oferece um conjunto de classes pertencentes ao *Namespace System.Data.Oracleclient* que facilitam a comunicação entre a aplicação desenvolvida e a base de dados, elas permitem que a aplicação funcione como utilizador da base de dados, aceda aos dados e faça a gestão dos mesmos. Este conjunto de classes foi usado durante o desenvolvimento da BMaPI, portanto, sempre que for mencionado, por exemplo, que se fez um registo de informação, acesso a informação ou atualização de informação na base de dados, subentende-se que essa ação aconteceu através dos recursos disponibilizados por este conjunto de classes.

Relativamente aos agentes, são registados dados importantes no momento da sua criação: o seu código de agente, o seu nome, a máquina na qual se encontra alojado, o endereço de IP dessa máquina, a sua localização completa no interior dessa máquina mais especificamente a sua *path* e por fim

o seu estado. Todas estas informações são essenciais para garantir uma boa comunicação com o agente e conseqüentemente um bom funcionamento do mesmo. De forma a facilitar a atividade do utilizador no momento da criação de um novo agente, este só terá de dar um nome ao agente, escolher qual o nome da máquina onde quer criar o agente e referir a sua localização, todos os outros dados serão automaticamente atribuídos. Esta ação só é possível devido à classe *Dns* fornecida pelo *Namespace System.Net*, esta possibilita obter o endereço de IP de uma determinada máquina colocando unicamente o seu *Hostname*, garantindo assim que não ocorrem erros de digitação do endereço, evitando também falhas na comunicação.

É de salientar ainda relativamente à criação de novos agentes que o programa Servidor Geral difere nesta fase do programa Cliente Remoto na medida em que o primeiro permite a criação de agentes em quaisquer das máquinas ligadas à plataforma enquanto que o Cliente Remoto permite apenas a criação de novos agentes na própria máquina.

Transitando agora de uma explicação a nível do utilizador para uma explicação mais técnica relacionada com a programação do sistema, serão então apresentados e devidamente descritos os principais processos desenvolvidos, cada um deles auxiliado por um diagrama para que a explicação se torne mais simples.

Em termos gerais a BMaPI baseia-se numa aplicação *multithread* Cliente/Servidor utilizando *sockets TCP/IP* para a troca de mensagens entre os seus intervenientes. Com o recurso a esta tecnologia *multithread* o servidor tem a capacidade de comunicar com mais do que um cliente ao mesmo tempo através da criação de um *thread* dedicado a cada cliente.

O uso deste conjunto de tecnologias vem conferir à BMaPI características importantes como a fiabilidade, escalabilidade, adaptabilidade e portabilidade. A sua arquitetura distribuída vem também beneficiar a BMaPI diminuindo o seu tempo de processamento e conseqüentemente o seu tempo de resposta. Além disso, estas tecnologias vêm conferir à BMaPI um elevado nível de interoperabilidade.

A inclusão da BMaPI na AIDA, vem então com as características anteriormente mencionadas, melhorar a performance global do sistema. Ainda

assim pretende-se que a **BMaPI** seja o mais autónoma possível, ou seja que não necessite que um utilizador a controle. Seguindo esse pressuposto as únicas atividades que necessitam expressamente do auxílio do utilizador correspondem à criação de novos agentes e à programação da sua atividade.

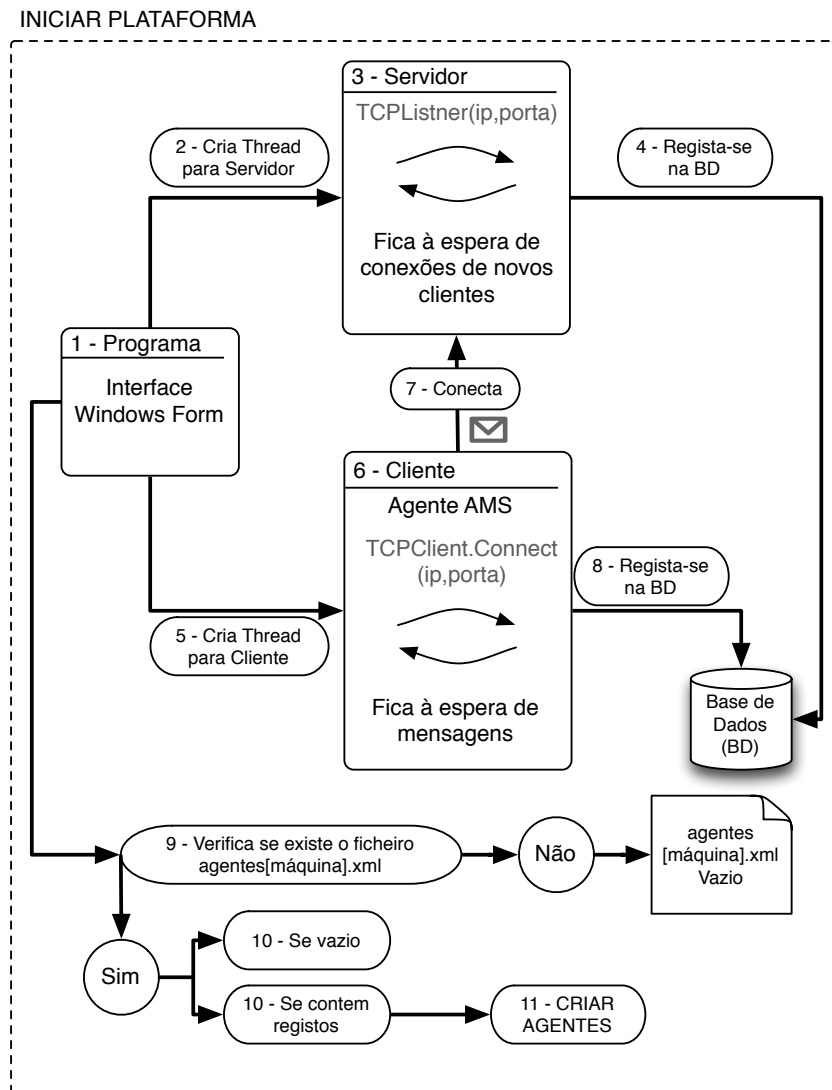


Figura 4.6: Processo automático de Inicialização da **BMaPI**.

O primeiro processo descrito corresponde à inicialização da **BMaPI**, representado pela Figura 4.6 e executado automaticamente quando o programa

Servidor Geral é executado.

Seguindo a numeração da Figura 4.6 é então possível constatar que em primeiro lugar o processo inicial, correspondente ao programa (1) cria um novo *thread* para poder, nesse subprocesso criar o servidor do sistema. A criação de novos *threads* (2 e 5) é possível recorrendo à classe *Thread* pertencente ao *Namespace System.Threading* que fornece um conjunto de métodos úteis para a sua implementação. O servidor é o primeiro componente a ser construído (3) a sua construção baseia-se na classe *TcpListener* pertencente ao *Namespace System.Net.Sockets*, esta classe fornece métodos simples de deteção e aceitação de solicitações de conexão por parte de um determinado cliente. Para criar o servidor basta então definir em que endereço de IP e em que porta é que se pretende que o servidor espere por novos clientes. Após a sua inicialização, regista na base de dados (4) a máquina na qual foi inicializado e o respetivo endereço de IP para que sempre que outra máquina queira servir de cliente para este servidor, saiba em que endereço de IP o servidor se encontra, sendo que a porta utilizada foi sempre a mesma. Desta forma o programa Servidor Geral fica apto para ser executado em qualquer máquina ou pode simplesmente mudar de máquina sem que seja necessária qualquer modificação no código da aplicação.

Tal como já foi referido anteriormente cada uma das máquinas que aloja agentes necessita de executar um dos programas (Servidor Geral ou Cliente Remoto) contudo cada uma delas irá conter um agente comum, o AMS. Este agente é responsável pelo controlo dos agentes que se encontram nessa máquina. A sua principal função é o registo de novos agentes na BMaPI. Após a construção do servidor, será criado um novo *thread* (5) para o agente AMS e em seguida recorrendo à classe *TcpClient* e ao IP do servidor, o cliente conecta-se ao servidor enviando-lhe uma primeira mensagem de criação. O servidor aceita o cliente e cria automaticamente um novo *thread* para manter a conexão com este, podendo assim ficar à espera de novos clientes, enquanto que o cliente fica à espera de mensagens. Este agente tal como o servidor regista-se na base de dados, permitindo dessa forma que a informação sobre a sua existência fique disponível para todo o sistema. É de salientar que no programa Cliente Remoto não existem as ações de criação do servidor (2-4),

tendo o processo início na criação do agente AMS (5).

De forma a automatizar ainda mais a BMaPI, cada vez que um novo agente é criado, a sua informação, para além de ser registada na base de dados ela é também registada na própria máquina recorrendo a um ficheiro XML. Este registo permite que sempre que o programa inicia, após a criação do AMS haja uma verificação (9) da existência do ficheiro apelidado como “agentes[máquina].xml”. Se o ficheiro não existir vai ser criado para posteriormente receber os dados dos agentes. Caso o ficheiro exista, se não for vazio (10), cada um dos agentes lá registados serão criados através do processo descrito na Figura 4.7. Este processo também é executado sempre que o utilizador cria um novo agente.

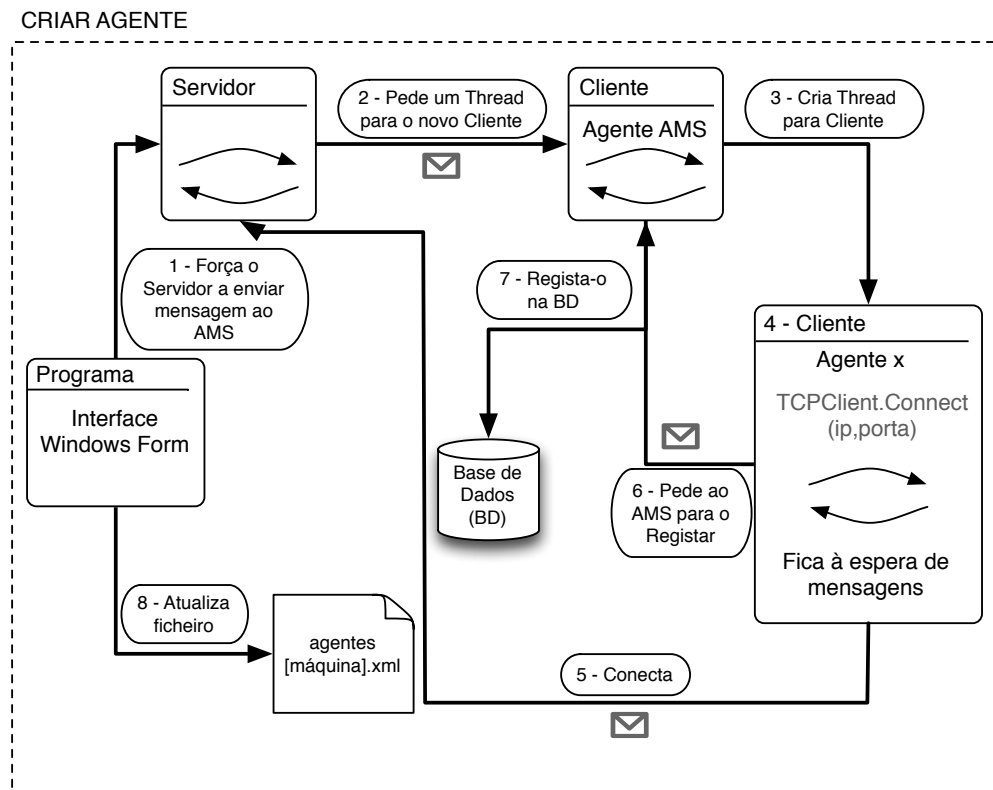


Figura 4.7: Processo de criação de um agente.

Analogamente à criação dos outros dois componentes, um novo agente,

ou seja um novo cliente necessita de um *thread* para ser criado. Dessa forma a própria interface após a receção dos dados do agente a criar, força (1) o servidor a enviar uma mensagem ao AMS da máquina onde o utilizador pretende criar o agente, requerendo um *thread* (2) para o respetivo agente. Após a receção da mensagem cabe ao agente AMS criar um *thread* (3) para o novo cliente fazendo nele correr o novo agente. O processo de criação do novo agente (4) é igual ao descrito anteriormente para o caso do AMS, contudo esse agente não tem a capacidade de se registar. Portanto após a aceitação por parte do servidor (5) o novo agente pede ao AMS para que este o registre na base de dados (7). Concluída a criação e o registo, a própria interface atualiza o ficheiro XML (8) recorrendo ao conjunto de classes oferecido pelo *Namespace System.Xml*. É de salientar ainda que os agentes que são criados automaticamente no final da inicialização da BMaPI, caso tenham uma forma de execução (ex: de 1 em 1 minuto) abordada posteriormente, automaticamente começarão a sua ação de acordo com este parâmetro.

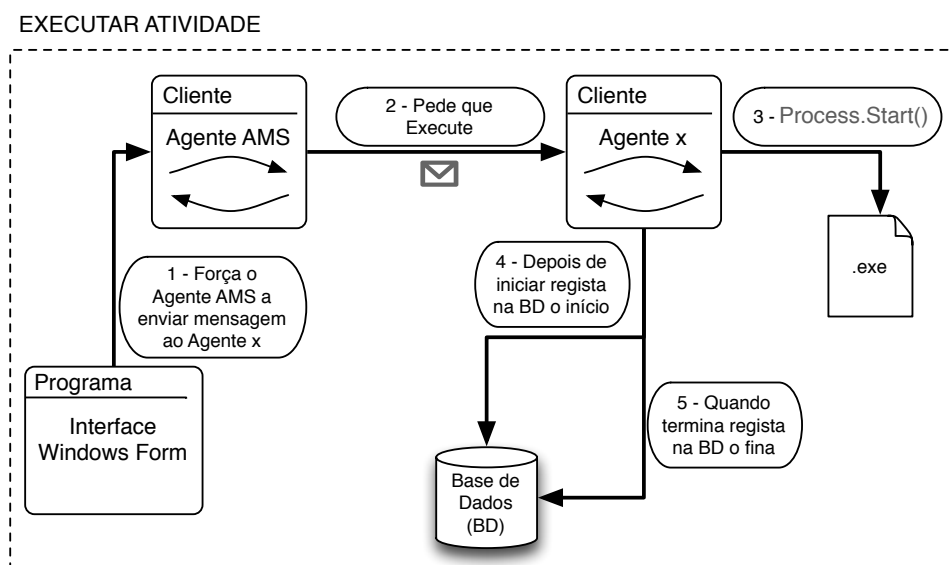


Figura 4.8: Processo de execução da atividade de um agente.

Um outro processo importante e indispensável no sistema é o processo de execução de atividade. Tal como já foi referido o utilizador pode pedir ao agente que execute a sua atividade. Esse processo é representado pela Figura

4.8.

De um modo semelhante aos restantes processos já descritos, neste processo específico, quando o utilizador pretende que o agente x execute a sua atividade a própria interface força o AMS a enviar uma mensagem ao agente x (1) requerendo que este execute a sua atividade (2). O agente x recorrendo à classe *Process* pertencente ao *Namespace System.Diagnostics* executa a sua atividade (3). No início dessa atividade o agente regista na base de dados a data e a hora de início da atividade (4) e de um modo semelhante quando a atividade é terminada também executa um registo na base de dados (5). Estes registos são importantes para o controlo da atividade de cada agente, contudo será aprofundado posteriormente o modo de utilização dos mesmos.

Para além dos processos apresentados, existem ainda outros processos importantes para o bom funcionamento da BMaPI, contudo devido à sua semelhança com os primeiros não irão ser descritos. É o caso do processo de eliminação do registo de um determinado agente, atividade atribuída ao agente AMS, análoga ao registo, cujo objetivo é banir o agente da plataforma, sendo que, caso o mesmo agente volte a ser criado pelo utilizador, o seu histórico de atividade permanece inalterado. A paragem da atividade de um agente é mais uma funcionalidade destes programas já anteriormente referida e muito semelhante ao processo de execução, fazendo com que a atividade seja interrompida imediatamente. Por fim existe um processo invisível para o utilizador que acontece sempre que um dos programas Cliente Remoto é terminado. Este processo informa ao servidor que o conjunto clientes daquela máquina foram destruídos, ou seja que não pode enviar mensagens para aqueles agentes.

Controlador *web*

A principal funcionalidade deste componente, desenvolvido recorrendo à plataforma ASP.NET e a uma extensão da ferramenta *Pentaho Community* designada CDE, é fornecer ao utilizador um ambiente atrativo de controlo e monitorização dos agentes registados na plataforma. Tem a estrutura de uma aplicação *web* de forma a permitir que o utilizador do sistema possa

controlar a **BMaPI** em qualquer lugar e em qualquer equipamento desde que o mesmo esteja conectado à rede, garantindo assim a elevada disponibilidade da mesma. Especificamente, este componente permite que o utilizador para além de consultar o número de agentes que estão registados na **BMaPI**, consulte as propriedades de cada um deles, atribua uma regra para a sua execução, agende as execuções de cada agente e ainda analise/monitorize a atividade de cada um deles.

O Controlador *web* disponibiliza uma página inicial com o panorama geral da **BMaPI**, isto é as máquinas que a constituem e os respetivos agentes nelas alojados. Essa página, apresentada na Figura 4.9 permite que o utilizador escolha qual o agente a analisar, sendo cada uma das lupas um botão que impulsiona a abertura de um novo separador relativo ao agente escolhido, podendo o utilizador ter aberto um separador para cada agente simultaneamente.

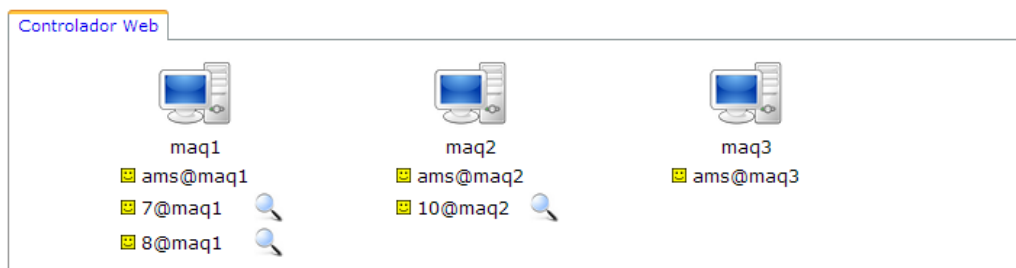


Figura 4.9: Página inicial do Controlador *web*.

O separador relativo a um determinado agente encontra-se subdividido em três páginas, a primeira, aberta por *default* corresponde às propriedades do agente, seguindo-se da que se encarrega do agendamento das atividades do agente e por fim, a página de monitorização.

A Figura 4.10 apresenta a página das propriedades organizada sob a forma de uma tabela composta por quatro grupos. O primeiro encarrega-se das propriedades do agente propriamente dito, o seu código, o seu nome, o seu estado e ainda a localização do seu executável. Permitindo assim que o utilizador conheça a localização física do agente e possa identificar o agente

Controlador Web 7@maq1

Fechar Separador

Propriedades Agendamento Monitorização

Detalhes do Agente

Propriedades do Agente

Código	7
Nome	7@maq1
Estado	Ativo
Executável	C:\agentes\ag7.exe

Propriedades do Sistema

Máquina	maq1
Endereço	192.168.1.7

Propriedades da Atividade

Agendamento	--
Data de Agendamento	--

Análise de Atividades

Última execução	--
Duração da última atividade	--
Número de execuções	0
Duração média das atividades executadas	--
Número de Erros ocorridos	0
Erro mais comum	--

Figura 4.10: Separador das Propriedades do agente selecionado.

através do seu nome ou do seu código.

As propriedades do sistema mostram a máquina onde o agente se encontra e o respetivo endereço de IP. Informação importante para a comunicação dos agentes.

As propriedades da atividade explicitam a forma de execução de um agente, ou seja de quanto em quanto tempo é que o agente vai executar a sua atividade, este parâmetro é definido na página de agendamento e a data e hora da sua definição corresponde à “Data de Agendamento”. Dando assim, a possibilidade ao utilizador de saber detalhes sobre a atividade do agente e caso a forma de execução seja alterada refere desde quando é que a

atividade se encontra com a forma atual.

O último grupo contém uma análise das atividades já executadas pelo agente. Apresenta a data e hora da última execução do agente, a duração dessa atividade, o número total de execuções, o percentil 50 da duração de todas as atividades e por fim o número de erros ocorridos e qual o mais comum. Todos estes dados apresentados baseiam-se na informação existente na base de dados da BMaPI.

Controlador Web 7@maq1

Fechar Separador

Propriedades Agendamento Monitorização

Atividades Agendadas

Agendar Atividade

Executar Agora

Executar às: 01 hr 00 min do(s) Dia(s):

Outubro de 2013						
seg	ter	qua	qui	sex	sáb	dom
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Limpar Seleção

Diariamente às: 01 hr 00 min

Semanalmente às: 01 hr 00 min às(aos): segundas-feiras

Mensalmente às: 01 hr 00 min no dia: 01

a cada: 1 segundos

Agendar

Não existem Atividades Agendadas

00:00

Figura 4.11: Separador para Agendamento de atividades.

A página de Agendamento é a página mais importante, sendo a única parte deste componente que comunica com os agentes. Ela permite que o utilizador programe e re programe a atividade do agente, isto é permite impor ao agente a forma como ele tem de executar a sua atividade sem que seja

necessário alterar o seu executável.

A Figura 4.11 mostra, do lado esquerdo a forma de execução do agente selecionado, caso este a tenha. Do lado direito permite que o utilizador solicite ao agente a sua execução, ou agende a atividade do agente. O utilizador tem opção de programar para vários dias, do calendário apresentado, a uma determinada hora, diariamente, semanalmente ou mensalmente também a uma determinada hora e num dia específico da semana ou do mês respetivamente. Pode ainda optar por programar o agente para executar de x em x tempo, segundos, minutos, horas ou dias. É de salientar que cada agente pode ter uma única forma de execução mas que esta pode ser alterada através do cancelamento da primeira e reprogramação da nova forma de execução.

Estas duas atividades, programação e cancelamento da forma de execução, são feitas através do envio de uma mensagem ao agente, a mensagem é diferente consoante a forma de programação e o processo será descrito na Figura 4.12.

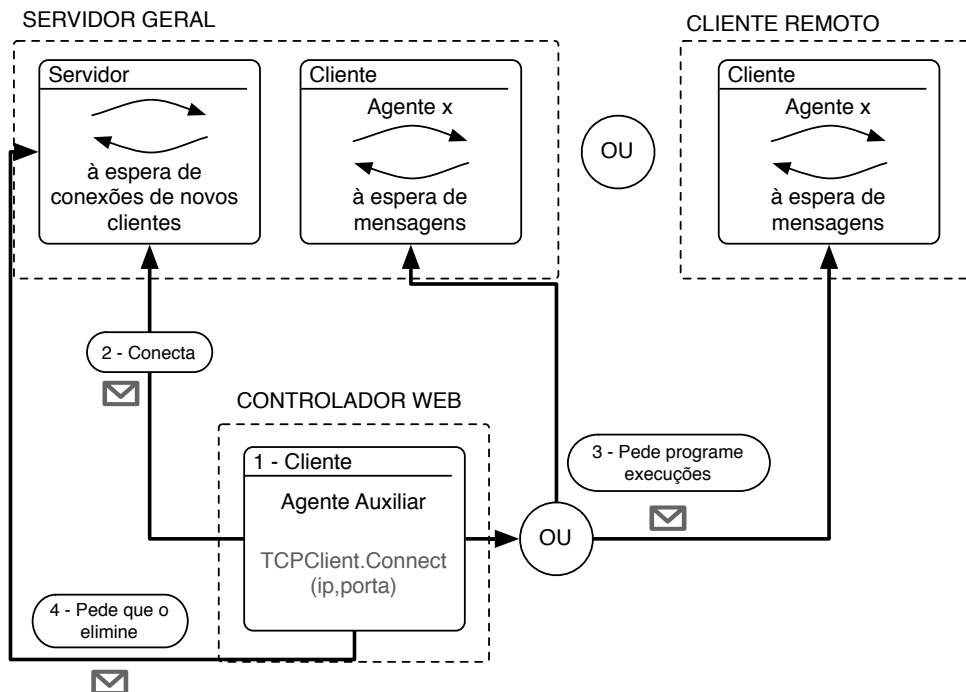


Figura 4.12: Separador para a Monitorização das atividades.

Contrariamente aos processos anteriormente descritos, o agente auxiliar criado no controlador *web* não é criado num novo *thread*, nem fica sempre à espera de mensagens. Este conecta-se ao servidor (2) e após a aceitação por parte do servidor, envia uma mensagem para o cliente agente *x* (3) que pode estar na mesma máquina que está o servidor ou noutra máquina. Por fim pede ao servidor que o elimine (4) para que sempre que seja necessário fazer uma programação, este volte a ser criado. O processo de cancelamento de uma programação é bastante semelhante a este e portanto não será explorado. O agente que recebe a mensagem, ou seja o agente que o utilizador pretende programar faz uso da classe *Timer* pertencente ao *Namespace System.Timers* para poder programar a sua atividade.

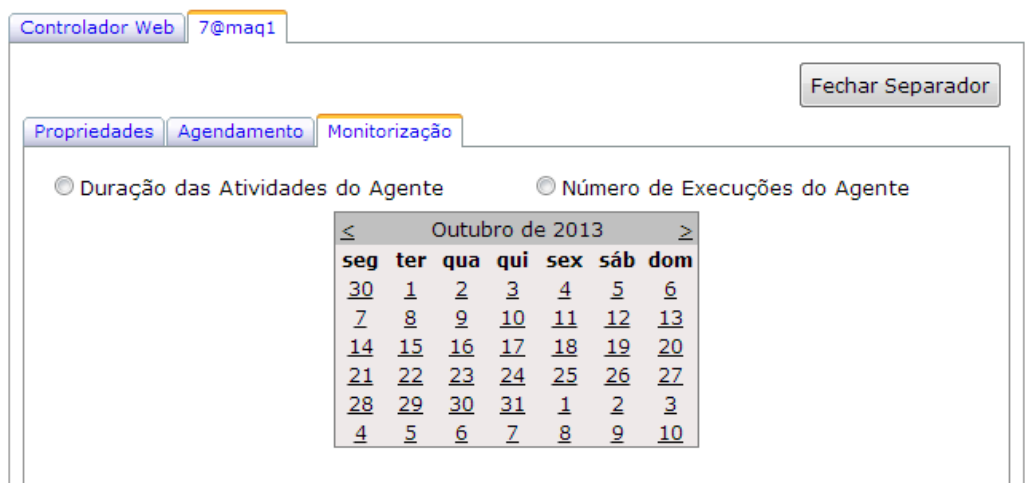


Figura 4.13: Separador para a Monitorização das atividades.

A página de monitorização apresentada em parte na Figura 4.13 permite ao utilizador visualizar dois tipos de gráficos, gráficos relativos à duração das atividades do agente e gráficos relativos ao número de execuções do agente. Sempre que o utilizador escolhe um tipo de gráfico e um determinado dia do calendário, surgem três gráficos na parte inferior da página. O primeiro gráfico corresponde à média, de duração das atividades ou ao número de atividades, por hora no dia escolhido. Seguindo-se de um outro gráfico com os mesmos parâmetros mas por dia, da semana correspondente ao dia sele-

cionado. Por fim o último gráfico corresponde também à média de duração e ao número de execuções por dia para todos os dias do mesmo mês do dia selecionado pelo utilizador.

Estes gráficos são gerados recorrendo à ferramenta *Pentaho Community* e integrados na aplicação *web* recorrendo a uma *iframe*. O modo de passagem de dados entre elas é através da *url* uma vez que a extensão *CDE* gera *dashboards* que podem ser visualizadas utilizando um qualquer *browser*. Os gráficos são dinâmicos, a sua construção recorre à informação que consta na base de dados através de uma *query* que neste caso recolhe da *url* o nome do agente pretendido e o dia selecionado no calendário, de forma a poder construir os gráficos pretendidos.

4.2.2 Implementação do Protótipo em Ambiente Real

A *BMaPI* foi implementada num ambiente laboratorial com recurso a várias máquinas e agentes básicos criados unicamente para o teste da plataforma. Todas as funcionalidades foram devidamente testadas e validadas. Após esse teste a *BMaPI* foi implementada numa das máquinas que constitui a *AIDA* implementada no *CHP*.

É importante referir que os recursos da máquina utilizada foram monitorizados antes e após a implementação da *BMaPI*, através de um sistema de monitorização e prevenção de falhas de máquinas, permitindo dessa forma comparar a carga que a *BMaPI* exerce sob a máquina, posteriormente apresentado e discutido.

A máquina em questão aloja três importantes agentes designados de 609, 35 e 36. O agente 609, no sistema proposto com nome *609@hsa-aida04* devido ao seu alojamento na máquina *hsa-aida04* é responsável por garantir a interoperabilidade com o *SAPE*, sistema de informação no qual são registados os dados resultantes de práticas de enfermagem. Os agentes 35 e 36 também nomeados da mesma forma que o 609, são agentes de validação de dados de exames. O 35 encarrega-se da validação de dados laboratoriais, por exemplo análises clínicas. Já o 36 encarrega-se de validação de dados pertencentes a outros *MCDT's*, podendo ter origem em diversos departamentos/serviços da

instituição.

Os resultados obtidos com esta implementação da [BMaPI](#) serão apresentados e discutidos no próximo Capítulo.

Capítulo 5

Apresentação e Discussão dos Resultados

Neste capítulo são apresentados e discutidos os resultados obtidos ao longo deste projeto que advêm do desenvolvimento da **BMaPI** e da sua implementação num ambiente real, o **CHP**. Em primeiro lugar será analisada a implementação da **BMaPI** na máquina *hsa-aida04* e os resultados gerados após a criação e programação dos agentes 609, 35 e 36, agentes que constituem a plataforma **AIDA**. O capítulo é finalizado com uma discussão sobre a performance global da **BMaPI** e o seu contributo para o sucesso da plataforma **AIDA**, respondendo dessa forma à **Questão 5**.

5.1 Monitorização da Atividade dos Agentes

Tal como já foi referido anteriormente, no Capítulo 4 o sistema desenvolvido foi implementado em ambiente real através da introdução do componente Servidor Geral numa das máquinas (*hsa-aida04*) que aloja agentes pertencentes à plataforma **AIDA**. Também o componente Controlador *web* foi publicado numa máquina (*hsa-portint-tes*) pertencente à mesma rede permitindo assim aceder à aplicação *web* em qualquer computador dessa rede. A base de dados e o servidor do *Pentaho Community* utilizados neste estudo, encontravam-se numa outra máquina (*hsa-aida19*) também pertencente à

mesma rede que as anteriores.

A BMaPI foi instalada no dia 10 de setembro de 2013 por volta das 15 horas e 30 minutos sendo os dados apresentados neste Capítulo referentes ao período de 10 de setembro a 16 de setembro de 2013. Os resultados apresentados são referentes aos três agentes criados pelo utilizador, recorrendo ao componente Servidor Geral, contudo são dados que o utilizador pode consultar no Controlador *web* permitindo uma deliberação, aqui discutida, sobre os agentes e a respetiva atividade.

A página inicial do Controlador *web* apresentada na Figura 5.1 permite ao utilizador visualizar a constituição da BMaPI, quais as máquinas envolvidas e quais os agentes que estas alojam. Para a análise dos resultados utilizou-se o exemplo do agente 609 podendo consultar no Apêndice C todos os resultados obtidos para os restantes agentes.

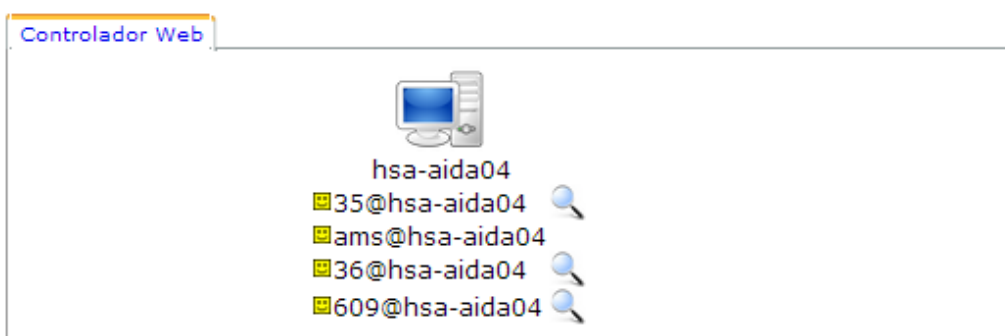


Figura 5.1: Página inicial do Controlador *web* no período analisado.

Contextualizando o Agente 609, agente de elevada importância no CHP, na medida em que garante a interoperabilidade com o SAPE e assim mantém toda a informação relativa a práticas de enfermagem disponível e acessível. É caracterizado como um agente de processamento de longa duração, tendo isso em consideração, a sua atividade foi programada para ser executada a cada 10 minutos, através da página de agendamento da Figura 5.2, salientado que o agente não executa a sua atividade caso ela já esteja a ser executada, isto é o agente é “chamado” de 10 em 10 minutos, supondo que o agente demora

mais de 10 minutos, quando é novamente “chamado” ainda se encontra a executar a atividade anterior, nesse caso não executa a atividade atual sendo “chamado” novamente 10 minutos depois.

The screenshot shows a web interface for scheduling an agent's activity. At the top, there are browser tabs for 'Controlador Web' and three agent identifiers: '609@hsa-aida04', '36@hsa-aida04', and '35@hsa-aida04'. A 'Fechar Separador' button is in the top right. Below the tabs are three menu items: 'Propriedades', 'Agendamento', and 'Monitorização'. The main content area is titled 'Atividades Agendadas' and 'Agendar Atividade'. On the left, under 'Atividades Agendadas', there is a refresh icon and a box showing 'Atividade - a cada: 10 minutos' and 'Atividade Agendada em: 10-09-2013 15:37:07'. A 'Cancelar Atividade' button is below this. On the right, under 'Agendar Atividade', there is an 'Executar Agora' button and a radio button for 'Executar às: 01 hr 00 min do(s) Dia(s):'. Below this is a calendar for 'Outubro de 2013' with days of the week and dates. A 'Limpar Seleção' button is below the calendar. There are three radio buttons for scheduling frequency: 'Diariamente às: 01 hr 00 min', 'Semanalmente às: 01 hr 00 min às(aos): segundas-feiras', and 'Mensalmente às: 01 hr 00 min no dia: 01'. There is also a radio button for 'a cada: 1 segundos'. An 'Agendar' button is at the bottom.

Figura 5.2: Página de programação da atividade do agente 609.

Após algum tempo de execução, neste caso aproximadamente 6 dias de execução, o utilizador da BMaPI já consegue tirar algumas conclusões sobre o agente através da página de propriedades apresentada na Figura 5.3.

Para além das suas propriedades individuais e da forma como executa a sua atividade, esta página apresenta uma análise geral sobre a atividade do agente selecionado.

Refere a data e a hora da última execução do agente, e a duração desta atividade em segundos, podendo o utilizador deliberar que a atividade co-

The screenshot shows a web interface for monitoring agents. At the top, there are tabs for 'Propriedades', 'Agendamento', and 'Monitorização'. The 'Propriedades' tab is active, displaying the following information:

Detalhes do Agente

Propriedades do Agente

Código	609
Nome	609@hsa-aida04
Estado	Ativo
Executável	C:\hgasa\vb\SILAg609.exe

Propriedades do Sistema

Máquina	hsa-aida04
Endereço	172.21.201.22

Propriedades da Atividade

Agendamento	a cada 10 minutos
Data de Agendamento	Atividade Agendada em: 10-09-2013 15:37:07

Análise de Atividades

Última execução	16-09-2013 11:36:46
Duração da última atividade	604
Número de execuções	827
Duração média das atividades executadas	329s
Número de Erros ocorridos	0
Erro mais comum	--

Figura 5.3: Página de propriedades do agente 609.

meçada às 11 horas e 36 minutos do dia 16 de setembro teve uma duração de 604 segundos, ou seja 10 minutos e 066 segundos, o que significa que a atividade agendada para as 11 horas e 46 minutos não foi executada sendo a próxima às 11 horas e 56 minutos.

A Figura apresenta também o número de execuções que o agente teve desde a sua criação até ao momento da análise e a duração média dessas atividades, essa duração média foi calculada recorrendo ao percentil 50 de forma a apresentar a medida de tendência central, eliminado dessa forma alguns *outliners*. Com estes dados, para além de o utilizador poder verificar o valores que fornecem pode ainda calcular de quanto em quanto tempo é que o agente executou a sua atividade na realidade, ou seja se o agente executou a sua atividade 604 vezes durante aproximadamente 6 dias, por dia executou cerca de 100 vezes o que significa que executou a atividade mais ou menos

de 25 em 25 minutos. É de salientar que a duração da atividade do agente não pode ser relacionada apenas com os recursos que este dispõe para a sua execução, uma vez que o agente é dinâmico, tanto pode ter uma quantidade elevada de dados para processar como pode nem ter nada para processar.

Para finalizar a análise desta página é possível ainda verificar o número de erros ocorridos ao longo das 827 atividades executadas pelo agente 609 e também desses, qual o erro que mais se verificou.

O componente Controlador *web* permite também ao utilizador o controlo da atividade do agente em tempo real, isto é uma monitorização das atividades de uma forma dinâmica. A página de monitorização dispõe então de dois tipos de gráfico um gráfico para a duração das atividades e outro para o número de atividades executadas.

Selecionando o tipo de gráfico pretendido e o dia para o qual pretende verificar a atividade do agente, surge um conjunto de 3 gráficos, dos quais só dois serão apresentados neste documento. O primeiro gráfico diz respeito a uma análise diária, seguindo-se de uma análise semanal e por fim uma análise mensal. Como o período apresentado é de curta duração não foi possível apresentar o terceiro gráfico.

5.1.1 Duração média das execuções do agente

Analisando em primeiro lugar a duração das atividades do agente 609 no dia 11, a análise diária apresentada na Figura 5.4, permite ao utilizador verificar que a atividade do agente teve durações menores no período das 07 horas às 08 horas na ordem dos 281,83 segundos e alcançou um pico das 11 horas às 12 horas com uma média de duração de 457,62. Este gráfico permite ao utilizador verificar quais os picos de duração da ação do agente, podendo neste caso relacionar com o horário de maior afluência aos SAPE, que se verifica durante a noite das 03 horas às 04 horas e durante o dia tendo um comportamento crescente abrupto deste as 08 horas até as 12 horas e depois um decréscimo gradual até às 23 horas, podendo ainda verificar-se uma pequena quebra na hora do almoço.

Contíguo a este gráfico é gerado um outro gráfico apresentado na Figura

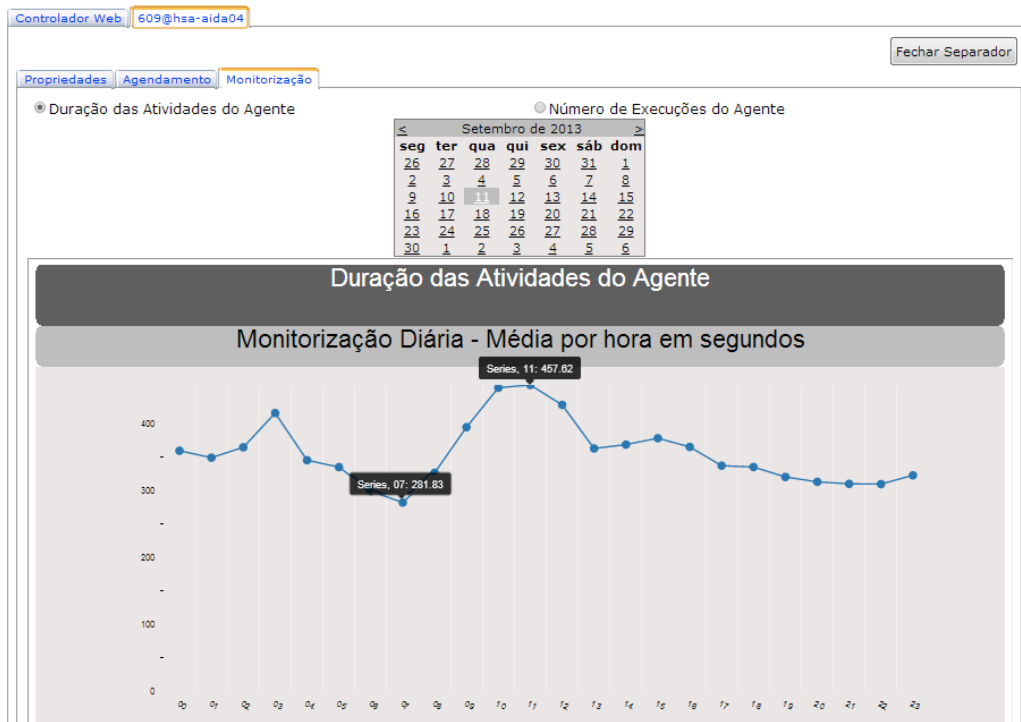


Figura 5.4: Análise diária do agente 609 para o dia 11 de setembro de 2013.

5.5 com a análise da semana no qual o dia selecionado está inserido. Este gráfico permite comparar a duração das atividades do agente executadas na semana em questão.

O utilizador pode então verificar que a atividade do agente 609 teve maior duração temporal no dia 11 com uma média de 390,55 segundos e uma menor duração nos dias 14 e 15 com uma média de 322,04 segundos. É possível então verificar que durante o fim de semana o agente executa a sua atividade com cerca de 50 segundos de diferença em relação aos outros dias da semana, sendo o meio da semana, a quarta-feira o dia que o agente mais tempo demora.

Com estes dois gráficos relativos à duração da atividade do agente, o utilizador poderá fazer uma melhor gestão de toda a plataforma sabendo em que horário e em que dias é que os agentes têm mais funções, ou seja que a sua atividade é mais demorada, dessa forma pode até deliberar qual será o melhor horário para fazer alterações de rotina ao sistema, atualizações, entre

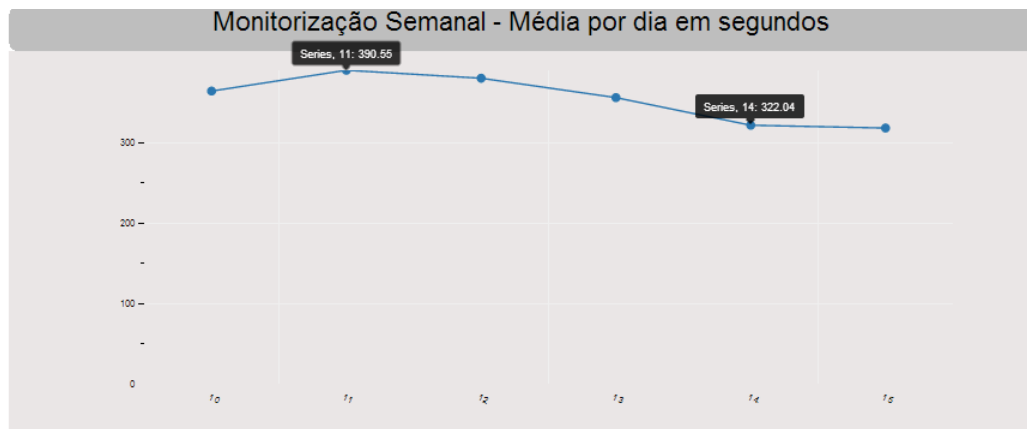


Figura 5.5: Duração média das atividades executadas por dia da semana de 10 a 15 de setembro.

outras operações para melhorar a performance global do sistema.

5.1.2 Número de execuções do agente

O outro tipo de gráfico gerado é o correspondente ao número de vezes que o agente executou a sua atividade. Analogamente aos primeiros gráficos mostrados é possível uma análise diária, semanal e mensal.

A análise diária é apresentada na Figura 5.6 e mostra o número de vezes que o agente foi executado em cada hora do dia 11 de setembro. É possível constatar que das 10 horas às 16 horas não foram realizadas todas as ações, ou seja ocorreu pelo menos uma ação por hora que demorou mais do que 10 minutos. É de salientar que apesar possa parecer incoerente, que das 15 horas às 16 horas o número de ações tenha diminuído, sendo que não se trata do pico de duração das atividades visto no gráfico da Figura 5.4, o que o gráfico permite referir é que nesse período de tempo apesar da média de duração ser de sensivelmente 355 segundos houve 2 ações cuja duração foi de mais de 10 minutos.

Por fim e de uma forma semelhante, é apresentado o resultado da análise semanal do número de execuções que contabiliza o número de vezes que o agente executou a sua atividade em cada dia da semana selecionada pelo

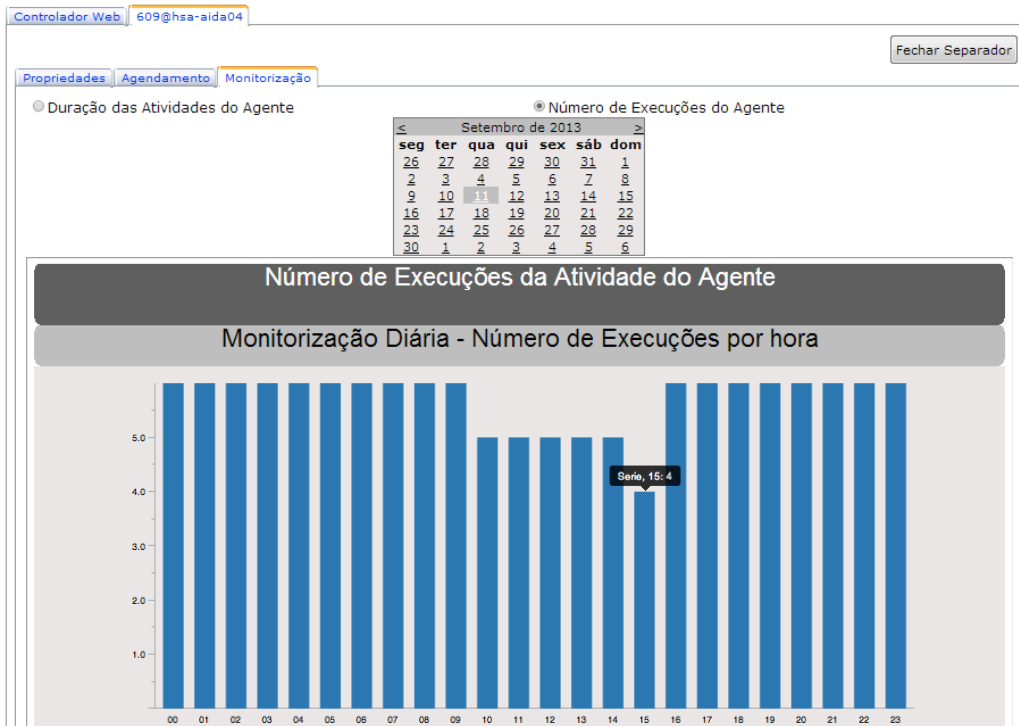


Figura 5.6: Número de vezes que o agente executou a sua atividade em cada hora do dia 11 de setembro.

utilizador. Tal como seria de prever após a discussão dos gráficos anteriores, os dias 11 e 12 apresentam um menor número de execuções do que os dias 13, 14 e 15. O dia 10 corresponde ao dia em que a plataforma foi implementada e portanto só começou a atividade dos agentes a partir das 15 horas e 30 minutos. Tal como já foi referido é natural que os dias 14 e 15 tenham mais execuções pois a duração das execuções do agente nestes dias também é inferior, podendo o utilizador constatar que nenhuma das execuções ultrapassou os 10 minutos de duração.

Esta análise do número de execuções de cada agente permite ao utilizador saber exatamente quando é que o agente foi executado e a que horas é que se verificou uma alteração do número de execuções, de forma a identificar o possível problema, podendo até o motivo ser a duração prolongada do agente identificada pelos gráficos de duração.



Figura 5.7: Número de vezes que o agente executou a sua atividade em cada dia da semana de 10 a 15 de setembro.

5.2 Performance Geral

Antes e durante a implementação da [BMAPI](#) todas as máquinas que constituem a [AIDA](#) foram controladas através de um sistema de monitorização e prevenção de falhas de máquinas. Através desse sistema e dos gráficos que ele gera é possível então comparar a percentagem de utilização de vários recursos antes e durante a implementação da [BMAPI](#).

Os recursos comparados foram a percentagem de *Central Processing Unit* (CPU) livre e a percentagem de memória *Random Access Memory* (RAM) livre na máquina *hsa-aida04* durante os períodos de 21 a 25 de agosto e 11 a 15 de setembro de forma a verificar o impacto que o novo sistema implementado causou nos recursos da máquina onde foi implementado.

A Figura 5.8 apresenta a comparação da percentagem de CPU livre nessa máquina nos períodos anteriormente referidos. É de salientar que os períodos selecionados correspondem aos mesmos dias semanais para que a comparação seja rigorosa uma vez que se verificou nos resultados anteriormente apresentados que o dia semanal influenciava a atividade dos agentes.

É possível então verificar que no geral houve um ligeiro aumento da percentagem de CPU livre. No primeiro período a percentagem média ronda os

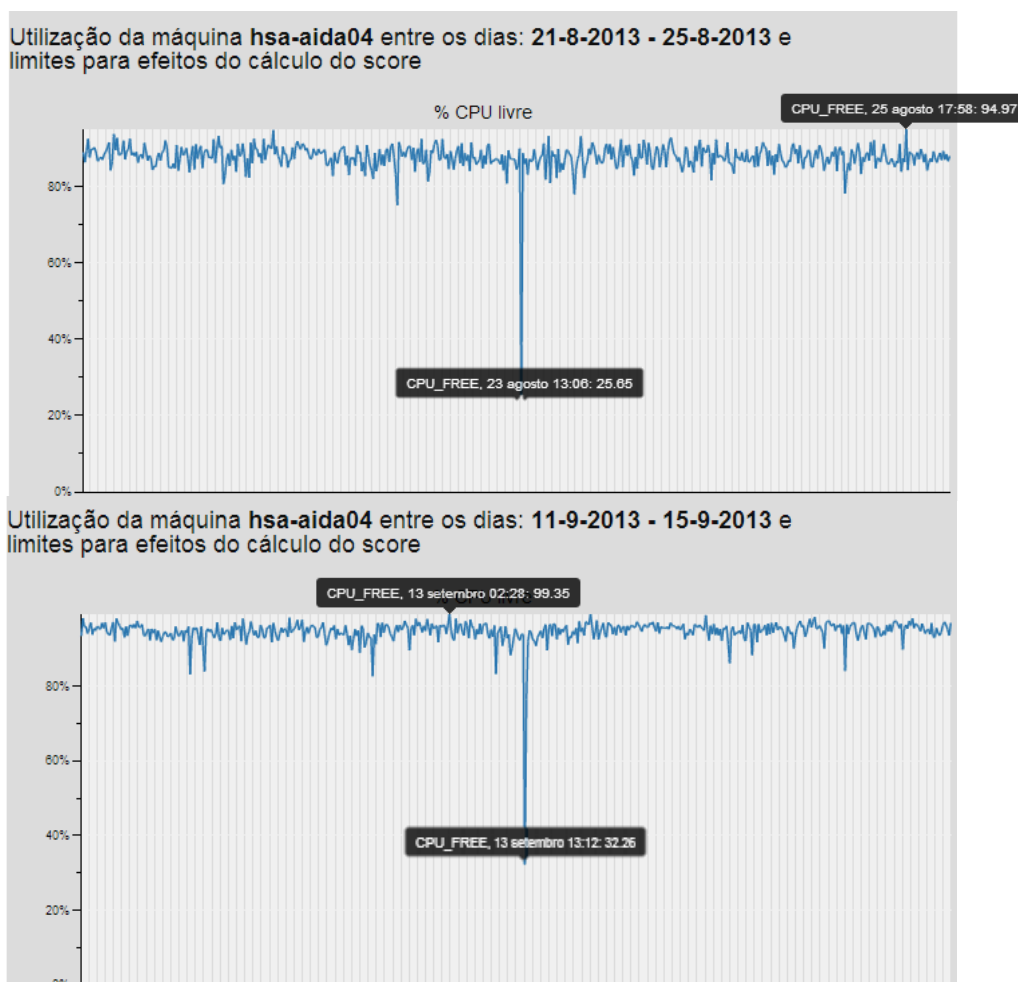


Figura 5.8: Comparação da percentagem de CPU livre.

83% enquanto que no segundo ronda os 87% , o que significa que a BMaPI veio de certa forma reduzir o uso dos recursos da máquina. No período de 21 a 25 de agosto o maior pico registado foi de 94,97% que é necessário ter em conta que ocorreu num domingo e este encontra-se praticamente isolado, enquanto que o pico no período de 11 a 15 de setembro alcançou os 99,35% sendo até confundido com mais 2 ou 3 ocorrências de picos na ordem dos 99%. Analisando a depressão abrupta, é possível concluir que ocorreu no mesmo dia da semana, à mesma hora contudo analogamente aos picos, no período de implementação da BMaPI verifica-se uma percentagem maior do

que no período de comparação.

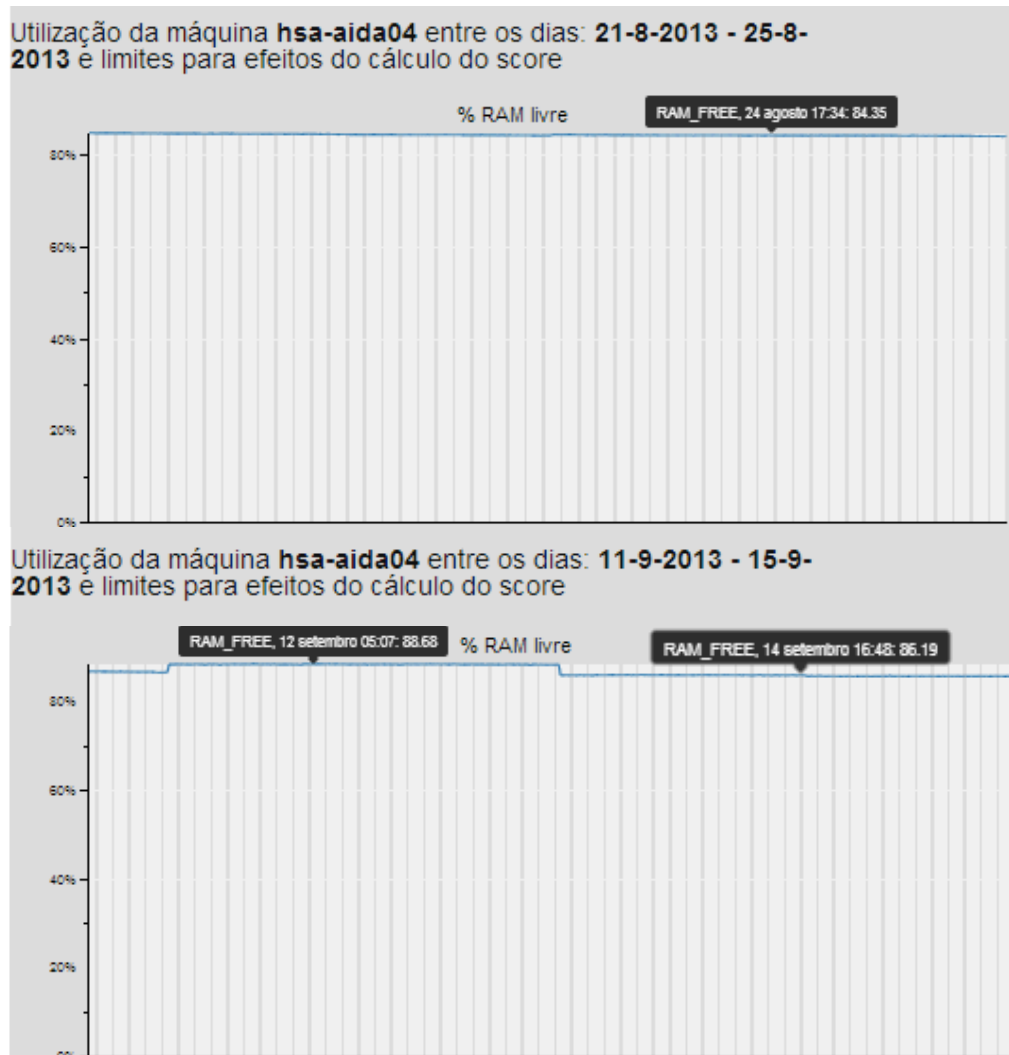


Figura 5.9: Comparação da percentagem de memória RAM livre.

A percentagem de memória RAM livre também se comporta de uma forma semelhante à de CPU (Figura 5.9), que apesar de ser constante, no período de 21 a 25 de agosto é na ordem dos 84,35% enquanto que no período no qual a BMaPI foi implementada é de 88,68% e de 86,19%.

É assim possível provar que a implementação da BMaPI não compromete os recursos da máquina na qual foi implementada, pelo contrário aumenta as

respetivas percentagens livres.

Desta forma e respondendo desde já à **Questão 5**, a plataforma **BMaPI** desenvolvida e apresentada neste documento, surge como um sistema que vem fornecer à plataforma **AIDA** uma maior funcionalidade, na medida em que promove funcionalidades que satisfazem os utilizadores e as necessidades declaradas pelos mesmos. Apresentam também uma maior usabilidade, pois constitui um sistema de fácil compreensão e funcionamento com uma interface atraente.

A sua arquitetura distribuída e as tecnologias utilizadas conferem à **BMaPI** uma elevada eficiência, na medida em que os recursos envolvidos são compatíveis com o nível de desempenho do sistema. Nomeadamente o recurso à tecnologia *multithread* que garante menor sobrecarga das máquinas, pois está provado que a gestão de *threads* gasta menos recursos do que a gestão de processos isolados e ainda a comunicação entre *threads* é uma ação muito mais eficiente do que a comunicação entre processos.

A arquitetura Cliente/Servidor confere à **BMaPI** para além das suas propriedades de interoperabilidade uma maior fiabilidade na medida que, a não ser que o servidor da plataforma falhe, o sistema mantém-se sempre ativo, grande escalabilidade e capacidade de crescimento e ainda uma elevada portabilidade e adaptabilidade, pois é capaz de ser transferido para um outro ambiente e adaptar-se a novas tecnologias respetivamente.

Por fim, o recurso a uma base de dados externa à aplicação confere à plataforma uma maior robustez, segurança dos dados (confidencialidade, integridade e disponibilidade), confiança, escalabilidade e organização. O uso da base de dados externa mantém a plataforma e a sua informação sempre acessível e disponível para que o utilizador possa usufruir das suas funcionalidades.

A construção e implementação da **BMaPI** descrita ao longo deste documento, vem então melhorar a performance global da plataforma **AIDA**, resolvendo de uma forma simples e eficaz uma das limitações da mesma, a dificuldade no controlo dos agentes e respetivas atividades. Esta plataforma vem assim, de um modo geral, contribuir para uma maior interoperabilidade entre os **SIH** e também entre as máquinas que constituem a **AIDA**, favo-

recendo assim um melhor fluxo de trabalho e de informação e uma melhor qualidade na prestação de cuidados de saúde, nas unidades hospitalares que usam a plataforma [AIDA](#), nomeadamente o [CHP](#).

Capítulo 6

Conclusões

Este projeto enquadra-se numa importante temática da atualidade, a interoperabilidade, que ao longo dos anos se tem destacado como um requisito na maior parte das entidades prestadoras de serviços. Na saúde, e com a elevada importância da informação clínica para as práticas médicas, existe uma enorme necessidade de distribuição da informação, de partilha entre serviços e especialidades. Numa primeira fase do trabalho foram estudados os [SIH](#) no geral e principalmente a necessidade de comunicação entre eles, bem como a tecnologia orientada a agentes como um novo conceito para a implementação da interoperabilidade.

A plataforma [AIDA](#) implementada em diversos hospitais portugueses há já alguns anos, tem revelado um enorme sucesso no que diz respeito à interoperabilidade entre os [SIH](#). É sua função o arquivo, a integração e a disseminação de informação médica em unidades de saúde e constitui basicamente um sistema multiagente ([MAS](#)), onde diversos agentes comunicam entre si de forma a atingirem um objetivo global, a interoperabilidade.

6.1 Contributos

A plataforma [AIDA](#) tem vindo a destacar-se devido ao seu papel vital na interoperabilidade entre os [SIH](#), é então importante garantir a satisfação dos utilizadores relativamente à performance da plataforma. Para isso, foi feito

um estudo de forma a apurar as suas principais vantagens e desvantagens. Em termos de vantagens destaca-se a elevada disponibilidade, acessibilidade e suporte em tempo útil da plataforma e também a elevada segurança da informação. Relativamente às limitações, é possível referir que as limitações que foram verificadas, não comprometem diretamente o bom funcionamento da plataforma, no entanto, a necessidade de uma maior controlo sobre os agentes surge como uma limitação da plataforma que se pretende contornar com o desenvolvimento deste projeto (Questão 1).

Desse modo, foi então proposto o desenvolvimento e implementação de uma plataforma baseada em agentes para a interoperabilidade (BMaPI), integrada na plataforma AIDA, que permita controlar e administrar uma comunidade de agentes, garantindo a sua sobrevivência num ambiente heterogéneo (Questão 2).

Tendo em conta a necessidades propostas, a plataforma BMaPI foi desenvolvida de forma a facilitar ao utilizador a criação e registo de novos agentes localmente e remotamente, permitir que o utilizador ative e desative serviços através do lançamento ou paragem de agentes, facilitar a programação e reprogramação da atividade dos agentes e monitorizar a atividade dos agentes (Questão 3).

Recorrendo à programação orientada a agentes, à sua comunicação através de mensagens ACL e a uma arquitetura cliente/servidor *multithread* com o uso de *sockets* TCP/IP foi então possível o desenvolvimento de dois dos três componentes da BMaPI. O terceiro componente, para além das tecnologias anteriormente referidas faz uso de uma ferramenta para facilitar a construção de gráficos dinâmicos das atividades executadas pelos agentes de forma a tornar mais simples a monitorização das suas atividades por parte do utilizador (Questão 4).

Após uma fase de testes e ajustes em ambiente laboratorial, uma parte da BMaPI foi implementada num ambiente real. Um dos componentes da BMaPI foi introduzido numa das máquinas que constituem a plataforma AIDA presente no CHP, na qual três agentes se encontram alojados. Os agentes foram criados e registados na BMaPI e a sua atividade foi monitorizada, apresentada e discutida durante o período de 10 a 16 de setembro de

2013. Os resultados apresentados permitem ao utilizador fazer uma análise detalhada sobre a atividade de cada agente, uma vez que são construídos gráficos dinâmicos nos quais o utilizador pode selecionar o dia que pretende explorar e dessa forma consultar a média de duração das atividades ou o número de atividades realizadas pelo agente numa determinada hora do dia ou num determinado dia da semana.

Relativamente ao agente 609, o utilizador pode constatar através do controlador *web*, que desde a sua criação até ao momento da recolha dos resultados o agente executou a sua atividade 867 vezes tendo essas atividades uma duração média de 329 segundos. Pode também verificar que o agente atinge um pico de duração às 11 horas da manhã e que durante os fins de semana demora menos tempo a realizar a sua atividade.

No que diz respeito à performance geral da *BMaPI* foi possível provar, recorrendo a um sistema de monitorização e prevenção de falhas de máquinas, que a *BMaPI* é uma plataforma eficiente, na medida em que não compromete os recursos da máquina no qual foi implementada, fazendo ainda com que as percentagens de *CPU* e *RAM* livres aumentem. Pode ainda referir-se que a sua implementação na plataforma *AIDA* vem de certo modo elevar a sua funcionalidade e usabilidade garantindo a interoperabilidade de uma forma fiável com grande escalabilidade e capacidade de crescimento (*Questão 5*).

É importante referir ainda que o trabalho desenvolvido deu origem a duas publicações, um capítulo de livro (Apêndice *D.3*) que foca a importância da interoperabilidade para melhorar a qualidade de prestação de serviços de saúde, e ainda um artigo de conferência submetido e que se encontra em fase de revisão pelos pares (Apêndice *D.4*), que foca a necessidade deste sistema e de um outro sistema de monitorização na plataforma *AIDA* de forma a melhorar a sua performance global. Foram ainda feitas mais duas publicações (Apêndices *D.1* e *D.2*) relacionadas com outros *SIH* implementados na plataforma *AIDA*.

6.2 Trabalho Futuro

Apesar da **BMaPI** implementada ser capaz de atingir todas as funcionalidades para ela idealizadas, existem sempre aspetos a melhorar, aproveitando para responder à **Questão 6** serão apresentadas algumas medidas a tomar.

Numa primeira etapa seria necessário implementar a **BMaPI** nas restantes máquinas que contém agentes e fazer pequenos ajustes de forma a adaptar o mesmo, na totalidade ao ambiente real.

Sugere-se também a criação de um agente responsável pelo tratamento dos dados dos outros agentes, isto é cada vez que um agente executa a sua atividade faz um registo na base de dados, se for um agente de curta duração e com atividade a decorrer num pequeno intervalo de tempo, este irá gerar centenas de registos diariamente. A função desse agente seria substituir todos os registos de uma determinada hora por um registo médio, ou seja um registo com a informação que é apresentada no gráfico, sendo dessa forma também necessária uma reestruturação da base de dados utilizada.

Novos gráficos deveriam ser construídos de forma a dar ao utilizador mais informações sobre os agentes e as respetivas atividades. Uma das sugestões seria um gráfico com a percentagem de **CPU** e memória **RAM** que o agente utiliza em cada uma das suas atividades.

Uma medida mais trabalhosa mas que seria interessante, é fazer com que esta plataforma comunicasse com o sistema de monitorização e prevenção de falhas de máquinas, utilizado para verificar a eficiência deste sistema, de forma a fornecer à **BMaPI** informações sobre os recursos de cada máquina. Com o recurso a essas informações respeitantes às máquinas e com a implementação de agentes móveis, seria possível que sempre que a máquina atingisse uma determinada percentagem de **CPU** ou memória **RAM** ocupada, os agentes presentes nessa máquina migrariam para outra que oferecesse mais recursos, fazendo dessa forma uma gestão automática dos recursos das máquinas que constituem a **AIDA**.

Por fim, seria uma mais valia a avaliação da funcionalidade e da usabilidade da plataforma de forma a encontrar mais aspetos que os utilizadores considerem importantes para integrar na **BMaPI** e aspetos que necessitam

de ser alterados na mesma de forma a melhorar a sua performance global.

Sugere-se ainda a utilização dos resultados apresentados neste documento para a elaboração de uma ou mais publicações científicas de forma a apresentar a [BMaPI](#) e as suas funcionalidades à comunidade científica.

Bibliografia

- [1] L. Cardoso, F. Marins, C. Quintas, F. Portela, M. Santos, A. Abelha, and J. Machado, “Interoperability in healthcare,” in *Cloud Computing Application for Quality Health Care Delivery*, A. Mourtzoglou, Ed. IGI Global Book (accepted for publication).
- [2] D. Isern, D. Sánchez, and A. Moreno, “Agents applied in health care: A review,” *International Journal of Medical Informatics*, vol. 79, no. 3, pp. 145–166, 2010.
- [3] J. Duarte, M. Salazar, C. Quintas, M. Santos, J. Neves, A. Abelha, and J. Machado, “Data quality evaluation of electronic health records in the hospital admission process,” in *9th IEEE/ACIS International Conference on Computer and Information Science (ICIS)*, Japão, 2010, pp. 201–206.
- [4] J. Weber-Jahnke, L. Peyton, and T. Topaloglou, “ehealth system interoperability,” *Information Systems Frontiers*, vol. 14, no. 1, pp. 1–3, 2012.
- [5] L. Palazzo, P. Sernani, A. Claudi, G. Dolcini, G. Biancucci, and A. F. Dragoni, “A Multi-Agent Architecture for Health Information Systems,” 2013. [Online]. Available: <http://netmed2013.dii.univpm.it/sites/netmed2013.dii.univpm.it/files/papers/paper5.pdf>
- [6] M. Miranda, M. Salazar, F. Portela, M. Santos, A. Abelha, J. Neves, and J. Machado, “Multi-agent systems for hl7 interoperability services,” *Procedia Technology*, vol. 5, pp. 725–733, 2012.
- [7] H. Peixoto, M. Santos, A. Abelha, and J. Machado, “Intelligence in interoperability with aida,” *Lecture Notes in Computer Science*, 2012.
- [8] M. Miranda, J. Duarte, A. Abelha, J. Machado, J. Neves, and J. Neves, “Interoperability in healthcare,” *EUROPEAN SIMULATION AND MODELLING CONFERENCE (ESM 2010), Hasselt, Belgium, 2010 – “Proceedings of the European Simulation and Modelling Conference (ESM 2010) ”*. [S.l.], 2010.
- [9] R. Rogers, Y. Peres, and W. Müller, “Living longer independently—a healthcare interoperability perspective,” *e & i Elektrotechnik und Informationstechnik*, vol. 127, no. 7, pp. 206–211, 2010.

- [10] H. L. S. International. About HL7. [Online]. Available: <http://www.hl7.org/about/index.cfm?ref=nav>
- [11] A. Abelha, J. Machado, M. Santos, S. Allegro, F. Rua, M. Paiva, and J. Neves, "Agency for integration, diffusion and archive of medical information," in *Artificial Intelligence and Applications*. Second IASTED International Conference, 2002.
- [12] F. for Intelligent Physical Agents, *FIPA ACL Message Structure Specification*, g ed., Foundation for Intelligent Physical Agents (FIPA), 12 2002.
- [13] F. Marins, L. Cardoso, F. Portela, M. Santos, A. Abelha, and J. Machado, "Intelligent information system to tracking patients in intensive care units," in *Lecture Notes in Computer Science*. Springer, 2013.
- [14] J. Machado, A. Abelha, P. Novais, J. Neves, and J. Neves, "Quality of service in healthcare units," *International Journal of Computer Aided Engineering and Technology*, vol. 2, no. 4, pp. 436–449, 2010.
- [15] M. Štula, D. Stipaničev, and J. Maras, "Distributed computation multi-agent system," *New Generation Computing*, vol. 31, no. 3, pp. 187–209, 2013.
- [16] T. Lippeveld, R. Sauerborn, C. Bodart *et al.*, *Design and implementation of health information systems*. World Health Organization Geneva, 2000.
- [17] J. Zaleski, *Integrating device data into the electronic medical record*. Wiley. com, 2008.
- [18] R. Haux, "Individualization, globalization and health—about sustainable information technologies and the aim of medical informatics," *International journal of medical informatics*, vol. 75, no. 12, pp. 795–808, 2006.
- [19] W. Kirch, *Encyclopedia of public health*. Springer, 2008, vol. 1154.
- [20] J. Duarte, C. F. Portela, A. Abelha, J. Machado, and M. F. Santos, "Electronic health record in dermatology service," in *ENTERprise Information Systems*. Springer, 2011, pp. 156–164.
- [21] A. Hoerbst and E. Ammenwerth, "Electronic health records," *Methods Inf Med*, vol. 49, no. 4, pp. 320–336, 2010.
- [22] D. Kalra and D. Ingram, "Electronic health records," in *Information technology solutions for healthcare*. Springer, 2006, pp. 135–181.
- [23] J. Neves, M. Santos, J. Machado, A. Abelha, S. Allegro, and M. Salazar, "Electronic health records and decision support local and global perspectives," *WSEAS transactions on biology and biomedicine*, vol. 5, no. 8, pp. 189–198, 2008.

- [24] J. Machado, A. Abelha, J. Neves, and M. Santos, "Ambient intelligence in medicine," in *Biomedical Circuits and Systems Conference, 2006. BioCAS 2006. IEEE*. IEEE, 2006, pp. 94–97.
- [25] J. Duarte, M. Miranda, A. Abelha, J. Machado, J. Neves, M. Santos, C. Alberto, M. Salazar, C. Quintas, A. Cabral *et al.*, "Agent-based group decision support in medicine," 2009.
- [26] F. Barbarito, F. Pinciroli, J. Mason, S. Marceglia, L. Mazzola, and S. Bonacina, "Implementing standards for the interoperability among healthcare providers in the public regionalized healthcare information system of the lombardy region," *Journal of Biomedical Informatics*, 2012.
- [27] B. Orgun and J. Vu, "H17 ontology and mobile agents for interoperability in heterogeneous medical information systems," *Computers in Biology and Medicine*, vol. 36, no. 7, pp. 817–836, 2006.
- [28] C. N. Mead *et al.*, "Data interchange standards in healthcare it-computable semantic interoperability: Now possible but still difficult. do we really need a better mouse-trap?" *Journal of Healthcare Information Management*, vol. 20, no. 1, p. 71, 2006.
- [29] J. Park and S. Ram, "Information systems interoperability: What lies beneath?" *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 4, pp. 595–632, 2004.
- [30] J. A. Mykkänen and M. P. Tuomainen, "An evaluation and selection framework for interoperability standards," *Information and Software Technology*, vol. 50, no. 3, pp. 176–197, 2008.
- [31] W. A. Khan, M. Hussain, A. M. Khattak, M. B. Amin, and S. Lee, "Saas based interoperability service for semantic mappings among health-care standards," in *8th International Conference on Innovations in Information Technology*, 2012.
- [32] M. Miranda, G. Pontes, A. Abelha, J. Neves, and J. Machado, "Agent based interoperability in hospital information systems," in *Biomedical Engineering and Informatics (BMEI), 2012 5th International Conference on*. IEEE, 2012, pp. 949–953.
- [33] M. Eichelberg, T. Aden, J. Riesmeier, A. Dogac, and G. B. Laleci, "A survey and analysis of electronic healthcare record standards," *ACM Computing Surveys (CSUR)*, vol. 37, no. 4, pp. 277–315, 2005.
- [34] W. Wang, A. Tolk, and W. Wang, "The levels of conceptual interoperability model: applying systems engineering principles to m&s," in *Proceedings of the 2009 Spring Simulation Multiconference on ZZZ*. Society for Computer Simulation International, 2009, p. 168.

- [35] F. Marques, G. P. Dias, and A. Zúquete, “A general interoperability architecture for e-government based on agents and web services,” in *Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on*. IEEE, 2011, pp. 1–6.
- [36] G. Acampora and A. Vitiello, “Collaborative memetic agents for enabling semantic interoperability,” in *Intelligent Agent (IA), 2013 IEEE Symposium on*. IEEE, 2013, pp. 21–27.
- [37] M. Contreras, E. Germán, M. Chi, L. Sheremetov *et al.*, “Design and implementation of a fipa compliant agent platform in .net,” *Journal of Object Technology*, vol. 3, no. 9, pp. 5–28, 2004.
- [38] C.-J. Su and C.-Y. Wu, “Jade implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring,” *Applied Soft Computing*, vol. 11, no. 1, pp. 315–325, 2011.
- [39] M. Miranda, J. Machado, A. Abelha, and J. Neves, “Healthcare interoperability through a jade based multi-agent platform,” in *Intelligent Distributed Computing VI*, ser. Studies in Computational Intelligence, G. Fortino, C. Badica, M. Malgeri, and R. Unland, Eds. Springer, 2013, vol. 446, pp. 83–88.
- [40] S. Bhat, N. S. Sidnal, and S. S. Manvi, “Agent based approach in accessing distributed health care services,” in *Security-Enriched Urban Computing and Smart Grid*. Springer, 2011, pp. 212–221.
- [41] R. Cruz-Correia, P. Vieira-Marques, P. Costa, A. Ferreira, E. Oliveira-Palhares, F. Araújo, and A. Costa-Pereira, “Integration of hospital data using agent technologies-a case study,” *Ai Communications*, vol. 18, no. 3, pp. 191–200, 2005.
- [42] P. M. Vieira-Marques, S. Robles, J. Cucurull, R. J. Cruz-Correia, G. Navarro, and R. Marti, “Secure integration of distributed medical data using mobile agents,” *Intelligent Systems, IEEE*, vol. 21, no. 6, pp. 47–54, 2006.
- [43] M. D. Rodríguez, J. Favela, A. Preciado, and A. Vizcaíno, “Agent-based ambient intelligence for healthcare,” *Ai Communications*, vol. 18, no. 3, pp. 201–216, 2005.
- [44] M. Tentori, J. Favela, and M. D. Rodriguez, “Privacy-aware autonomous agents for pervasive healthcare,” *Intelligent Systems, IEEE*, vol. 21, no. 6, pp. 55–62, 2006.
- [45] S. Kirn, O. Herzog, P. Lockemann, and O. Spaniol, *Multiagent Engineering: Theory and Applications in Enterprises (International Handbooks on Information Systems)*. Springer-Verlag New York, Inc., 2006.
- [46] L. Braun, F. Wiesman, v. d. J. Herik, and A. Hasman, “Agent support in medical information retrieval,” in *Working notes of the IJCAI-05. Workshop on agents applied in health care*, 2005, pp. 16–25.

- [47] P. Kostkova, J. Mani-Saada, G. Madle, and J. Weinberg, "Agent-based up-to-date data management in national electronic library for communicable disease," in *Applications of Software Agent Technology in the Health Care Domain*. Springer, 2003, pp. 105–124.
- [48] J. McNiff, *Action research: Principles and practice*. Routledge, 2013.
- [49] J. Moraes, W. Souza, L. Ferreira Pires, L. T. Cavalini, and A. Prado, "A novel architecture for message exchange in pervasive healthcare based on the use of intelligent agents," in *Proceedings of the 10th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, 2013.
- [50] M. Warkentin, V. Sugumaran, and R. Sainsbury, "The role of intelligent agents and data mining in electronic partnership management," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13 277–13 288, 2012.
- [51] R. Rodrigues, P. Gonçalves, M. Miranda, F. Portela, M. Santos, J. Neves, A. Abelha, and J. Machado, "An Intelligent Patient Monitoring System," in *20th International Symposium on Methodologies for Intelligent Systems*, vol. 7661, World Intelligence Congress, Macau. Lecture Notes in Computer Science - Springer, 2012.
- [52] O. Etzioni and D. S. Weld, "Intelligent agents on the internet: Fact, fiction, and forecast," *IEEE expert*, vol. 10, no. 4, pp. 44–49, 1995.
- [53] M. Wooldridge, N. R. Jennings *et al.*, "Intelligent agents: Theory and practice," *Knowledge engineering review*, vol. 10, no. 2, pp. 115–152, 1995.
- [54] S. Li and M. M. Kokar, "Agent communication language," in *Flexible Adaptation in Cognitive Radios*. Springer, 2013, pp. 37–44.
- [55] M. Fasli and M. Michalakopoulos, "Developing Software Agents using .NET."
- [56] M. J. Donahoo and K. L. Calvert, *TCP/IP sockets in C: practical guide for programmers*. Morgan Kaufmann, 2009.
- [57] P. Lakshmi Anand K, Director of MSPVL Polytechnic College, "Client /server communication using tcp /ip." [Online]. Available: <http://www.ustudy.in/node/7215>
- [58] Microsoft, "Chapter 1: Introduction to .net." [Online]. Available: <http://technet.microsoft.com/en-us/library/bb496996.aspx>
- [59] Microsoft, "Namespace system.data.oracleclient," 2013. [Online]. Available: [http://msdn.microsoft.com/pt-br/library/system.data.oracleclient\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.data.oracleclient(v=vs.110).aspx)
- [60] Microsoft, "Classe dns," 2013. [Online]. Available: [http://msdn.microsoft.com/pt-br/library/system.net.dns\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.net.dns(v=vs.110).aspx)
- [61] Microsoft, "Namespace system.net.sockets," 2013. [Online]. Available: [http://msdn.microsoft.com/pt-br/library/system.net.sockets\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.net.sockets(v=vs.110).aspx)

- [62] Microsoft, “Namespace system.threading,” 2013. [Online]. Available: [http://msdn.microsoft.com/pt-br/library/system.threading\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.threading(v=vs.110).aspx)
- [63] Microsoft, “Namespace system.xml,” 2013. [Online]. Available: [http://msdn.microsoft.com/pt-br/library/system.xml\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.xml(v=vs.110).aspx)
- [64] Microsoft, “Namespace system.diagnostics,” 2013. [Online]. Available: [http://msdn.microsoft.com/pt-br/library/system.diagnostics\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.diagnostics(v=vs.110).aspx)
- [65] Microsoft, “Namespace system.timers,” 2013. [Online]. Available: [http://msdn.microsoft.com/pt-br/library/system.timers\(v=vs.110\).aspx](http://msdn.microsoft.com/pt-br/library/system.timers(v=vs.110).aspx)
- [66] Webdetails, “Cde documentation.” [Online]. Available: <http://www.webdetails.pt/ctools/cde.html>

Apêndice A

Glossário

.NET

Iniciativa da *Microsoft*, que visa uma plataforma única para o desenvolvimento e execução de sistemas e aplicações. Todo o código gerado pela .NET pode ser executado em qualquer dispositivo que possua a *framework* da plataforma. [xiii](#), [7](#), [31](#), [33](#), [42–44](#), [54](#), [56](#), [58](#), [100](#)

Namespace

Forma de organização de classes de um determinado programa. Pode ser definido pelo programador ou como parte de uma biblioteca de classes existente. [7](#), [44](#), [45](#), [58](#), [59](#), [61](#), [63](#), [64](#), [69](#)

framework

Estrutura de suporte definida para auxiliar a organização e o desenvolvimento de um projeto de *software*. Pode incluir programas de suporte, bibliotecas de código, *scripts* ou outros *software* para auxiliar o desenvolvimento de um projeto. [7](#), [24](#), [26](#), [32](#), [33](#), [42](#), [54](#), [56](#), [58](#), [99](#), [100](#)

iframe

Código [HTML](#) que faz com que uma determinada página seja aberta dentro de outra. [70](#)

multithread

Capacidade de execução de várias tarefas de um mesmo programa, aparentemente em simultâneo. [45](#), [59](#), [84](#), [88](#)

socket

Interface de comunicação bidirecional controlada pelo sistema operativo, criada para que os processos possam enviar/receber mensagens de/para outros processos, através de uma rede. [xiii](#), [7](#), [21](#), [33](#), [39–42](#), [55](#), [59](#), [88](#)

thread

Forma de um processo se dividir a si mesmo em duas ou mais tarefas que podem ser executadas simultaneamente. Fluxo único de controlo sequencial dentro de um determinado programa. Permitem que um programa simples possa executar várias tarefas diferentes ao mesmo tempo, independentemente umas das outras. [45](#), [59](#), [61](#), [63](#), [69](#), [84](#)

trigger

Recurso de programação executado sempre que um evento associado ocorrer. [22](#), [23](#)

web

Sistema de documentos ligados através de hipermédia (hiperligações em forma de texto, vídeo, som e outras animações digitais) interligados e executados através da Internet. [xiii](#), [xiv](#), [4](#), [19](#), [31](#), [42](#), [43](#), [48](#), [56](#), [64](#), [65](#), [69](#), [70](#), [73](#), [74](#), [77](#), [89](#), [100](#)

ASP.NET

Plataforma desenvolvida pela *Microsoft* para o desenvolvimento de aplicações *web*. É um componente do IIS que permite através de uma linguagem de programação integrada na *framework* .NET criar páginas *web* dinâmicas. [64](#)

Apêndice B

Mensagens ACL

B.1 *Performatives* e o seu significado

Tabela B.1: Tipos de atos comunicativos **FIPA** - Parte 1 (Adaptado de [15]).

<i>Performative</i>	Utilização
<i>Accept Proposal</i>	Agente <i>i</i> aceita uma proposta para executar uma ação, anteriormente submetida pelo Agente <i>j</i> .
<i>Agree</i>	Agente <i>i</i> concorda executar algumas ações.
<i>Cancel</i>	Agente <i>i</i> informa o Agente <i>j</i> que o Agente <i>i</i> não tenciona que o Agente <i>j</i> realize determinadas ações.
<i>Call for Proposal</i>	Agente <i>i</i> pede a apresentação de propostas para executar uma determinada ação, é geralmente usado para iniciar um processo de negociação.
<i>Confirm</i>	Agente <i>i</i> informa o Agente <i>j</i> que uma dada proposição é verdadeira quando o Agente <i>j</i> desconfia determinada proposição.
<i>Disconfirm</i>	Agente <i>i</i> informa o Agente <i>j</i> que uma dada proposição é falsa quando o Agente <i>j</i> acredita que determinada proposição é verdadeira.
<i>Failure</i>	Agente <i>i</i> informa o Agente <i>j</i> que tentou executar uma ação, mas esta falhou.

Tabela B.2: Tipos de atos comunicativos FIPA - Parte 2 (Adaptado de [15]).

<i>Performative</i>	<i>Utilização</i>
<i>Inform</i>	Agente <i>i</i> informa o Agente <i>j</i> que uma dada proposição é verdadeira.
<i>Inform If</i>	Agente <i>i</i> informa o Agente <i>j</i> quando é que uma proposição é verdadeira e quando não é. É geralmente utilizado como par de <i>Query If</i> .
<i>Inform Ref</i>	Agente <i>i</i> informa o Agente <i>j</i> algum objeto que corresponde a um descritor (ex: nome). É geralmente utilizado como par de <i>Query Ref</i>
<i>Not Understood</i>	Agente <i>i</i> informa o Agente <i>j</i> que não percebeu a ação anterior do Agente <i>j</i> (ex: uma mensagem).
<i>Propagate</i>	Agente <i>i</i> envia uma mensagem para o Agente <i>j</i> e quer que o Agente <i>j</i> identifique os agentes indicados pelo respectivo descritor e que propague a mensagem para esses agentes.
<i>Propose</i>	Agente <i>i</i> apresenta uma proposta de execução ao Agente <i>j</i> .
<i>Proxy</i>	Agente <i>i</i> solicita ao Agente <i>j</i> para identificar agentes que satisfazem um dados descritor e enviar a mensagem a esses agentes.
<i>Query If</i>	Agente <i>i</i> pergunta ao Agente <i>j</i> se uma determinada proposição é ou não verdadeira.
<i>Query Ref</i>	Agente <i>i</i> pergunta ao Agente <i>j</i> qual o objeto identificado por um determinado descritor
<i>Refuse</i>	Agente <i>i</i> recusa uma determinada ação ao Agente <i>j</i> e explica qual a razão para essa recusa.
<i>Reject Proposal</i>	Agente <i>i</i> rejeita uma proposta de execução de uma ação durante uma negociação.
<i>Request</i>	Agente <i>i</i> solicita ao Agente <i>j</i> para que este execute uma ação.
<i>Request When</i>	Agente <i>i</i> pede ao Agente <i>j</i> que execute uma ação quando uma dada proposição se torna verdade.
<i>Request Whenever</i>	Agente <i>i</i> pede ao Agente <i>j</i> que execute uma ação quando uma dada proposição se torna verdade e, posteriormente sempre que a proposição se tornar verdade.
<i>Subscribe</i>	Agente <i>i</i> solicita a notificação persistente sobre um determinado valor de referência. Este ato só termina com um ato de <i>Cancel</i>

Apêndice C

Monitorização da atividade dos Agentes

C.1 Agente 35

Controlador Web | 609@hsa-aida04 | 36@hsa-aida04 | 35@hsa-aida04

Fechar Separador

Propriedades | **Agendamento** | Monitorização

Atividades Agendadas

↻

Atividade - a cada:
1 minutos
Atividade Agendada em: 10-09-2013 15:36:42

Cancelar Atividade

Agendar Atividade

Executar Agora

Executar às: 01 hr 00 min do(s) Dia(s):

≤ Outubro de 2013 ≥

seg	ter	qua	qui	sex	sáb	dom
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Limpar Seleção

Diariamente às: 01 hr 00 min

Semanalmente às: 01 hr 00 min às(aos): segundas-feiras

Mensalmente às: 01 hr 00 min no dia: 01

a cada: 1 segundos

Agendar

Figura C.1: Página de programação da atividade do agente 35.

104 APÊNDICE C. MONITORIZAÇÃO DA ATIVIDADE DOS AGENTES

Controlador Web 609@hsa-aida04 36@hsa-aida04 35@hsa-aida04

Fechar Separador

Propriedades Agendamento Monitorização

Detalhes do Agente

Propriedades do Agente

Código	35
Nome	35@hsa-aida04
Estado	Ativo
Executável	C:\hgsa\vb\SILAg35.exe

Propriedades do Sistema

Máquina	hsa-aida04
Endereço	172.21.201.22

Propriedades da Atividade

Agendamento	a cada 1 minutos
Data de Agendamento	Atividade Agendada em: 10-09-2013 15:36:42

Análise de Atividades

Última execução	16-09-2013 11:51:21
Duração da última atividade	10
Número de execuções	8415
Duração média das atividades executadas	6s
Número de Erros ocorridos	0
Erro mais comum	--

Figura C.2: Página de propriedades do agente 35.

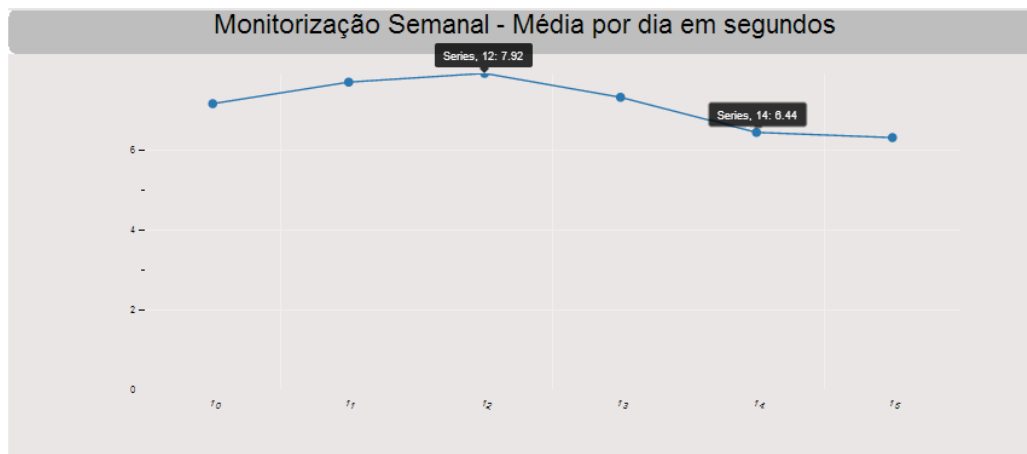


Figura C.3: Duração média das atividades executadas por dia da semana de 10 a 15 de setembro.

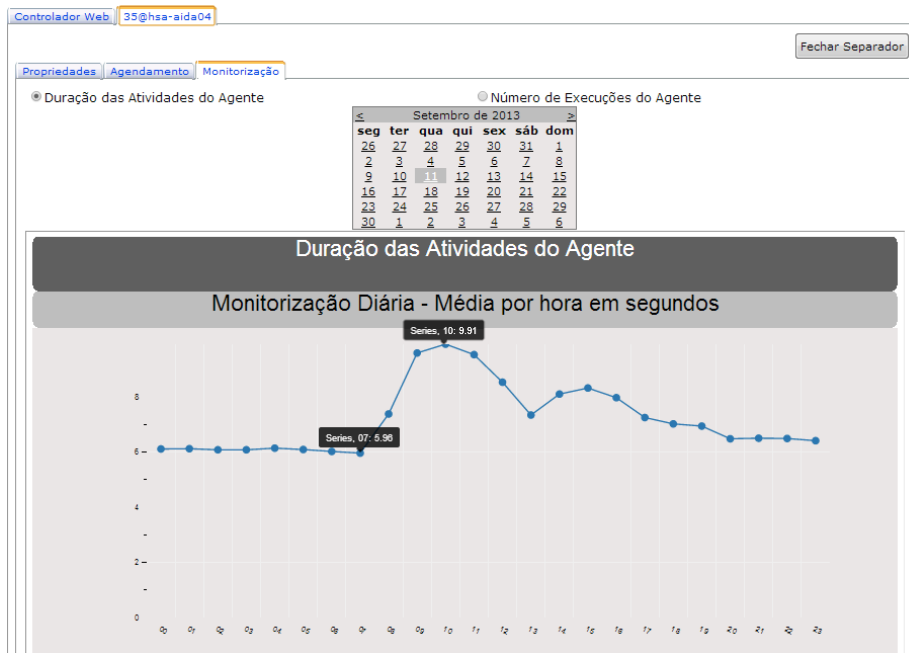


Figura C.4: Duração média das atividades executadas por hora para o dia 11 de setembro de 2013.

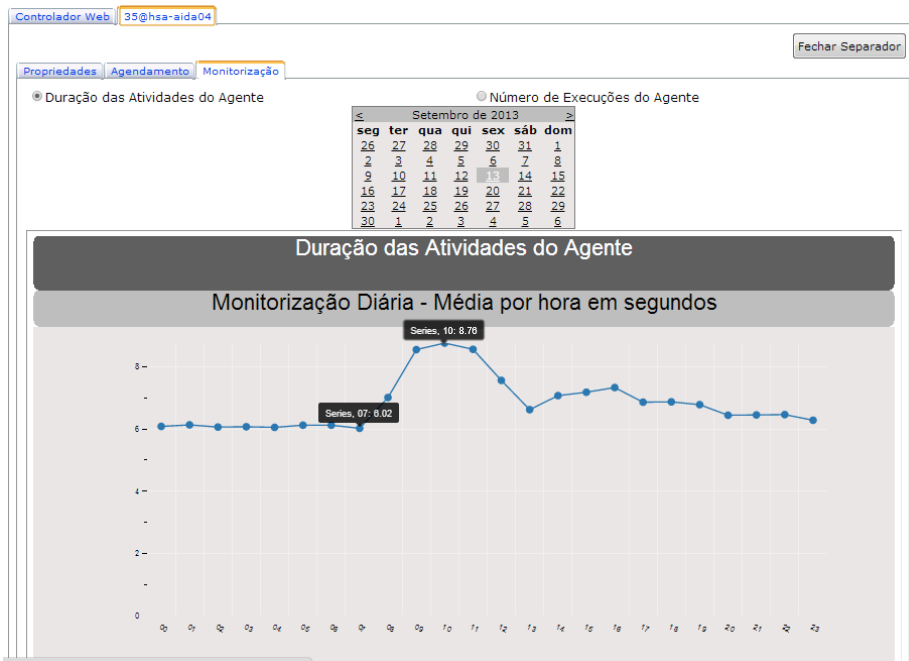


Figura C.5: Duração média das atividades executadas por hora para o dia 13 de setembro de 2013.

106 APÊNDICE C. MONITORIZAÇÃO DA ATIVIDADE DOS AGENTES

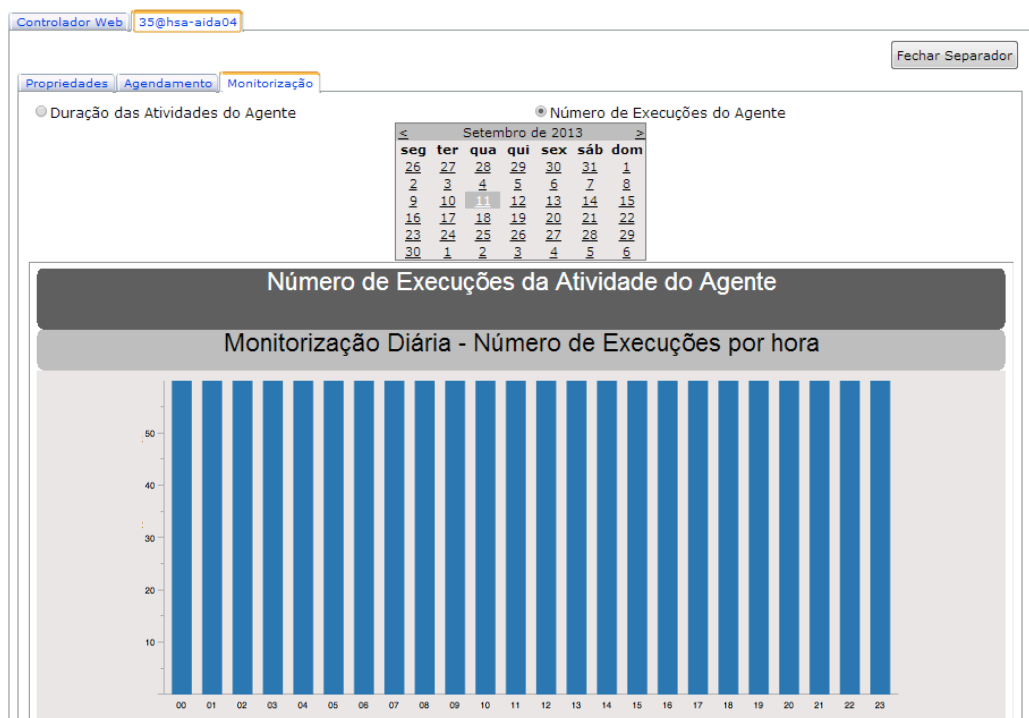


Figura C.6: Número de vezes que o agente executou a sua atividade por hora para o dia 11 de setembro de 2013.

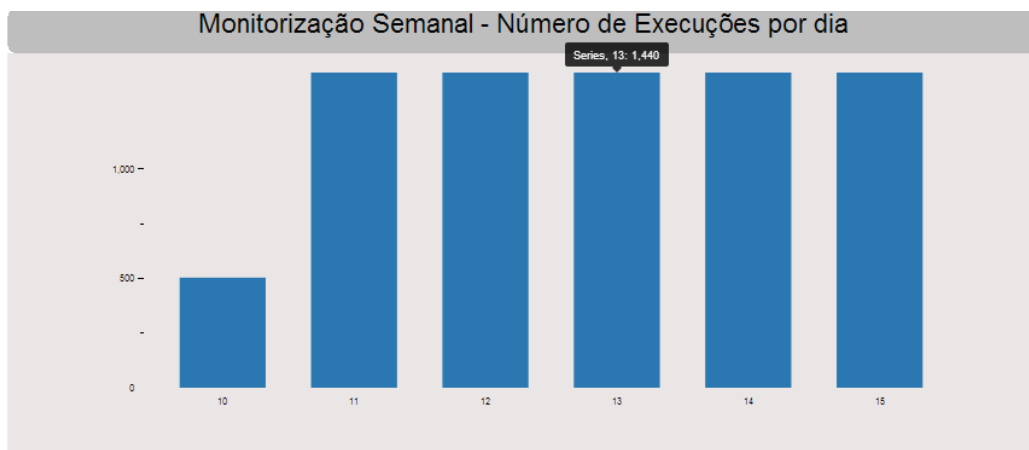


Figura C.7: Número de vezes que o agente executou a sua atividade em cada dia da semana de 10 a 15 de setembro.


C.2 Agente 36

Controlador Web 609@hsa-aida04 36@hsa-aida04 35@hsa-aida04

Propriedades **Agendamento** Monitorização

Fechar Separador

Atividades Agendadas



Atividade - a cada:

1 minutos

Atividade Agendada em:
10-09-2013 15:36:54

Cancelar Atividade

Agendar Atividade

Executar Agora

Executar às: 01 hr 00 min do(s) Dia(s):

≤ Outubro de 2013 ≥						
seg	ter	qua	qui	sex	sáb	dom
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Limpar Seleção

Diariamente às: 01 hr 00 min

Semanalmente às: 01 hr 00 min às(aos):
segundas-feiras

Mensalmente às: 01 hr 00 min no dia:
01

a cada: 1 segundos

Agendar

Figura C.8: Página de programação da atividade do agente 36.

108 APÊNDICE C. MONITORIZAÇÃO DA ATIVIDADE DOS AGENTES

Controlador Web | 609@hsa-aida04 | 36@hsa-aida04 | 35@hsa-aida04

Fechar Separador

Propriedades | Agendamento | Monitorização

Detalhes do Agente

Propriedades do Agente

Código	36
Nome	36@hsa-aida04
Estado	Ativo
Executável	C:\hgsa\vb\SILAg36.exe

Propriedades do Sistema

Máquina	hsa-aida04
Endereço	172.21.201.22

Propriedades da Atividade

Agendamento	a cada 1 minutos
Data de Agendamento	Atividade Agendada em: 10-09-2013 15:36:54

Análise de Atividades

Última execução	16-09-2013 11:50:38
Duração da última atividade	12
Número de execuções	8414
Duração média das atividades executadas	9s
Número de Erros ocorridos	0
Erro mais comum	--

Figura C.9: Página de propriedades do agente 36.

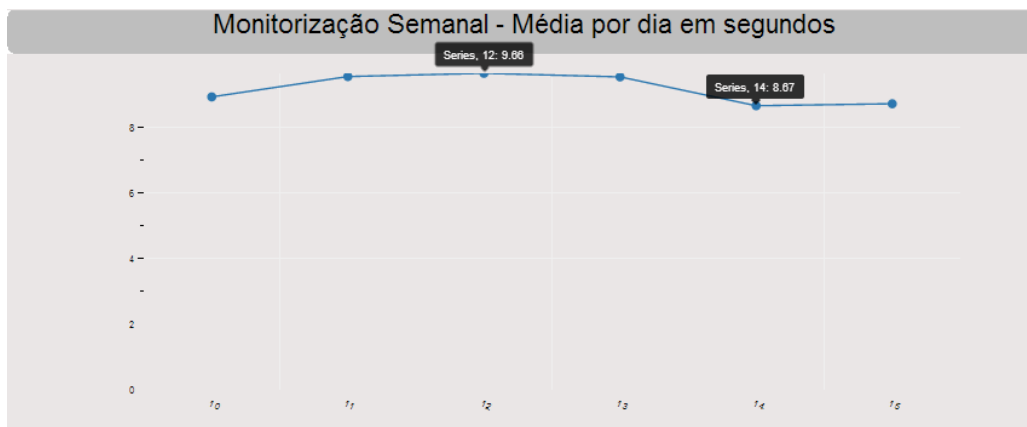


Figura C.10: Duração média das atividades executadas por dia da semana de 10 a 15 de setembro.

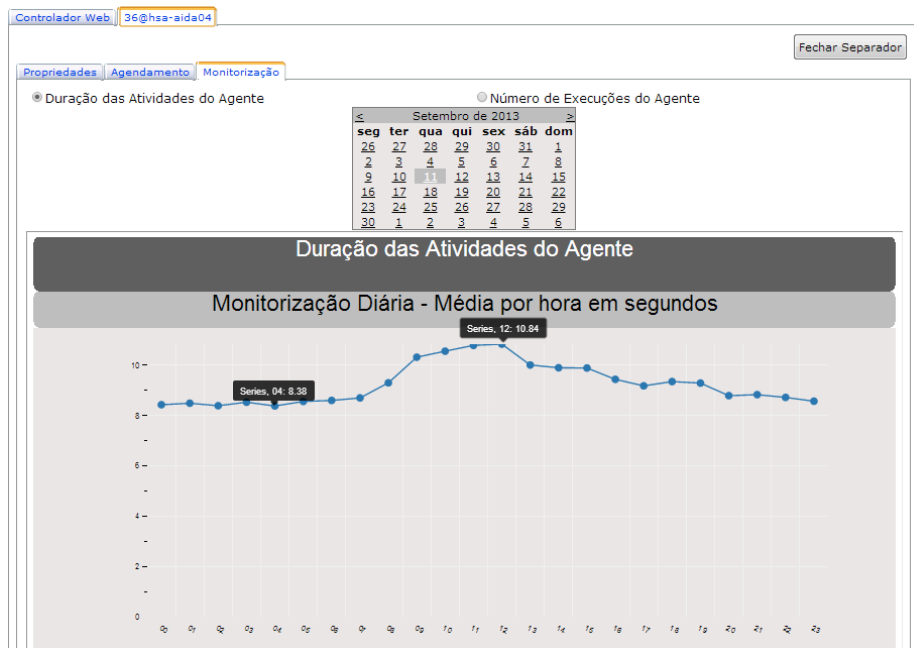


Figura C.11: Duração média das atividades executadas por hora para o dia 11 de setembro de 2013.

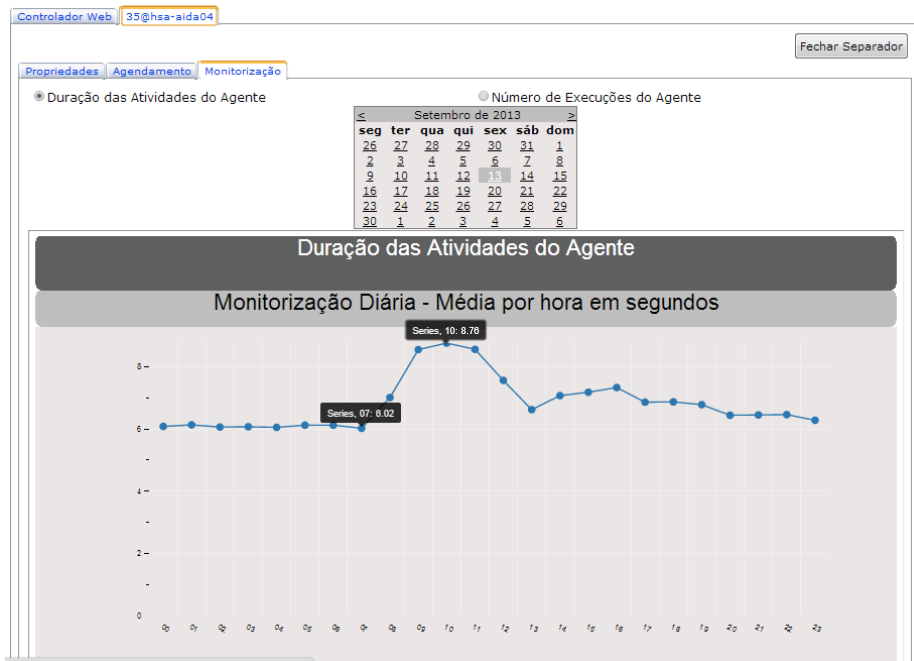


Figura C.12: Duração média das atividades executadas por hora para o dia 13 de setembro de 2013.

110 APÊNDICE C. MONITORIZAÇÃO DA ATIVIDADE DOS AGENTES

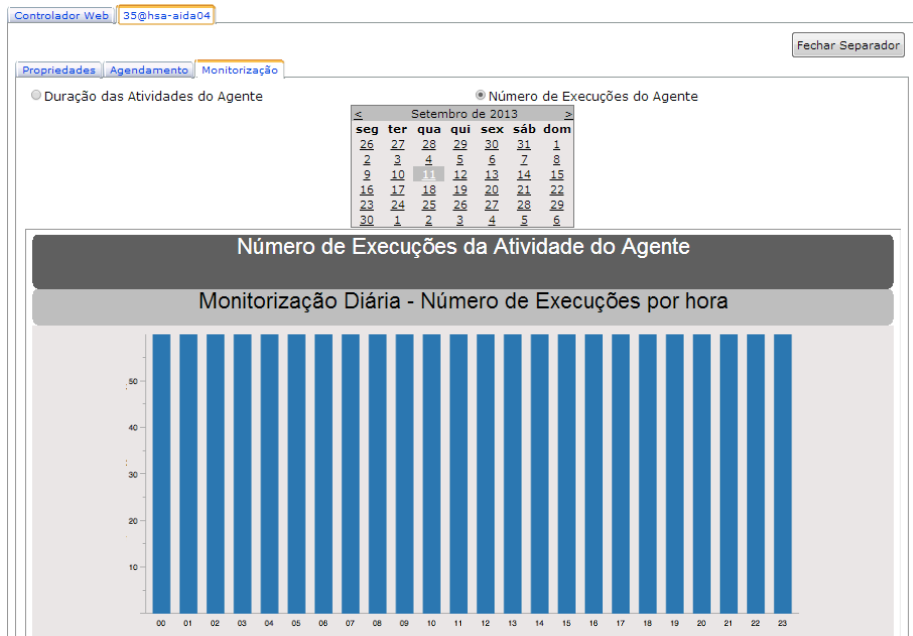


Figura C.13: Número de vezes que o agente executou a sua atividade por hora para o dia 11 de setembro de 2013.

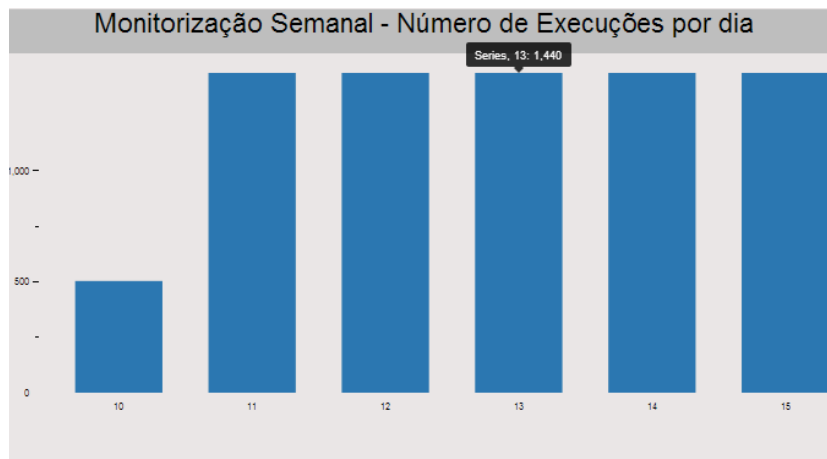


Figura C.14: Número de vezes que o agente executou a sua atividade em cada dia da semana de 10 a 15 de setembro.

Apêndice D

Publicações

D.1 Tracking People and Equipment Simulation inside Healthcare Units

Autores: Cátia Salgado, Luciana Cardoso, Pedro Gonçalves, António Abelha, and José Machado

Conferência: 4th International Symposium on Ambient Intelligence (ISAmI 2013)

Livro: Ambient Intelligence – Software and Applications, Advances in Intelligent Systems and Computing Volume 219, pp. 9-16

Editora: Springer International Publishing Switzerland

Ano: 2013

Estado: Publicado

Abstract: *“Simulating the trajectory of a patient, health professional or medical equipment can have diverse advantages in a healthcare environment. Many hospitals choose and to rely on RFID tracking systems to avoid the theft or loss of equipment, reduce the time spent looking for equipment, finding missing patients or staff, and issuing warnings about personnel access to unauthorized areas. The ability to successfully simulate the trajectory of*

an entity is very important to replicate what happens in RFID embedded systems. Testing and optimizing in a simulated environment, which replicates actual conditions, prevent accidents that may occur in a real environment. Trajectory prediction is a software approach which provides, in real time, the set of sensors that can be deactivated to reduce power consumption and thereby increase the system's lifetime. Hence, the system proposed here aims to integrate the aforementioned strategies - simulation and prediction. It constitutes an intelligent tracking simulation system able to simulate and predict an entity's trajectory in an area fitted with RFID sensors. The system uses a Data Mining algorithm, designated SK-Means, to discover object movement patterns through historical trajectory data."

Keywords: *RFID object tracking, Trajectory prediction, Simulation, Healthcare, Data Mining, SK-Means.*

D.2 Intelligent Information System to Tracking Patients in Intensive Care Units

Autores: Fernando Marins, Luciana Cardoso, Filipe Portela, Manuel Santos, António Abelha, and José Machado

Conferência: 7th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2013)

Livro: Lecture Notes in Computer Science, Volume 8276, pp. 54–61

Editora: Springer International Publishing Switzerland

Ano: 2013

Estado: Publicado

Abstract: *“With the increasing expansion of health information systems, there is a need to create an interface: human, machine and the surrounding environment. This interface is called Ambient Intelligence and it has been increasing in the healthcare area. In this paper it is presented the Ambient Intelligence system implemented in the Intensive Care Unit of Centro Hospitalar do Porto, a hospital in the north of Portugal. This Ambient Intelligence is consisted by INTCare system, which the main goal is monitoring the patients’ vital signs, PaLMS system, responsible for the patient’s localisation and identification and AIDA, the platform that guarantees the interoperability from all information systems in the hospital. Furthermore, an usability evaluation was performed, described in this article, to find out what can be improved.”*

Keywords: *Ambient Intelligence, Monitoring System, Interoperability.*

D.3 Interoperability in Healthcare

Autores: Luciana Cardoso, Fernando Marins, Cesar Quintas, Filipe Portela, Manuel Santos, António Abelha and José Machado

Livro: Cloud Computing Applications for Quality Health Care Delivery, Anastasius Moutzoglou (ed)

Editora: IGI Global Book

Ano: 2014

Estado: Aceite para Publicação

Abstract: *“With the advancement of technology, all patient information has been being computerized in order to facilitating the work of healthcare professionals and to improve de quality of healthcare delivery. However, there are many heterogeneous information systems that need to communicate so as to share information and to make it available when and where it is needed. To respond to this requirement it was created the Agency for Integration, Dissemination and Archiving of medical information (AIDA), a multi-agent and service based platform that ensures interoperability among healthcare information systems.*

In order to improve the performance of the platform, beyond the SWOT analysis performed, it was created a system to prevent failures that may occur in the platform database and also in machines where the agents are executed. It was possible conclude that in the Centro Hospitalar do Porto, the critical workload of AIDA is the period between 10:00 and 12:00.”

Keywords: *Interoperability, AIDA, Intelligent Agent, HL7, Multi Agent System, Fault forecasting, Database workload, SWOT analysis.*

D.4 Intelligent Systems for Monitoring and Preventing Interoperability Procedures

Autores: Luciana Cardoso, Fernando Marins, Filipe Portela, Manuel Santos, António Abelha, and José Machado

Conferência: 29th Symposium On Applied Computing (SAC 2014)

Editora: ACM/IEEE digital library

Ano: 2014

Estado: Submetido

Abstract: *“Interoperability is no longer a technological option, it is a fundamental requirement, especially when the environment consists of different information sources. It contributes to delivering effective care and ensuring the health and well-being of million of patients world-wide.*

The Agency for Integration, Diffusion and Archive of medical information (AIDA) is a system that ensures the inter-operability of Healthcare Information Systems. This paper presents two kinds of systems that are being tested in Centro Hospitalar do Porto (CHP). These systems were developed not only to improved system performance as well as the information security, specifically to improve the confidentiality, integrity and availability of the system.

In general, the prototypes presented represent two fault forecasting systems, which monitor and prevent faults in AIDA’s machines and agents and also a system for the management and control of agents.

During a tests phase, the fault forecasting systems demonstrated that are capable to prevent faults in AIDA’s agents and machines. The monitoring process, which is the basis of these systems, demonstrated being a fundamental tool for the resources management of AIDA.

These systems developed and presented in this paper demonstrated that contribute to the improvement of integrity and availability of AIDA.”

Keywords: *Interoperability in healthcare, AIDA, fault forecasting, multi-agent system.*