Universidade do Minho
Escola de Engenharia

Fernando de Abreu Marins

Monitorização e Prevenção em Plataformas de Interoperabilidade Hospitalar

outubro de 2013

Monitorização e Prevenção em Plataformas de Interoperabilidade Hospitalar
Fernando de Abreu Marins
UMinho | 2013

Universidade do Minho
Escola de Engenharia

Fernando de Abreu Marins

Monitorização e Prevenção em Plataformas de Interoperabilidade Hospitalar

Dissertação de Mestrado em Informática Médica
Mestrado Integrado em Engenharia Biomédica

Trabalho efetuado sob orientação do
Professor Doutor José Manuel Ferreira Machado
e coorientação do
Mestre Carlos Filipe da Silva Portela

outubro de 2013

Declaração

Nome: Fernando de Abreu Marins

Endereço eletrónico: f.abreu.marins@gmail.com

Cartão de Cidadão: 13453100

Título da Dissertação: Monitorização e Prevenção em Plataformas de Interoperabilidade Hospitalar

Orientador: Professor Doutor José Manuel Ferreira Machado
Co-orientador: Mestre Carlos Filipe Portela

Ano de conclusão: 2013

Designação do Mestrado: Mestrado Integrado em Engenharia Biomédica
Área de Especialização: Ramo de Informática Médica

Universidade do Minho, __ / __ / __

Assinatura: _____

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao meu orientador, Professor Doutor José Machado, pela oportunidade, por toda a atenção concebida na realização deste projeto e sobretudo pelos assuntos que aprendi com ele, não só acerca de Informática Médica, mas também sobre outros assuntos banais da vida quotidiana. Isto tudo sempre com uma boa pitada de humor, que tornou todo este processo mais estimulante.

É com grande apreço que agradeço todo o apoio do Mestre Carlos Filipe Portela, co-orientador desta dissertação, que se mostrou sempre disponível para auxiliar no que fosse preciso.

Não posso deixar de referir aqui uma palavra de gratidão ao Centro Hospitalar do Porto, mais concretamente ao Departamento de Sistemas de Informação do Hospital Santo António, pela oportunidade de desenvolver o meu trabalho e usufruir de um dos sistemas informáticos mais avançados dos hospitais portugueses.

Gostaria também de agradecer aos meus amigos de curso, com quem fiz verdadeiros laços de amizade que levo comigo para a vida e que me acompanharam nos grandiosos, nos espetaculares, e vá, nos menos bons momentos. Agradeço todas essas brutais aventuras pioneiras e às horas felizes que aqui passamos juntos.

Um grande obrigado aos meus amigos da velha guarda da Tertúlia Valasarense pelos fantásticos momentos passados que serviram para descontrair das frequências e dos trabalhos da universidade.

Por último, resta-me agradecer aos amigos e familiares mais próximos que sempre me acompanharam e acreditaram em mim. Ao meu irmão Guilherme, meu amigo incansável. Ao meu pai, certamente a pessoa que mais contribui

para o meu conhecimento quer com factos completamente aleatórios que ele lê nos livros mais ocultos, quer conhecimentos realmente importantes. À minha mãe, lutadora dos meus sonhos juntamente com o meu pai e que proporcionou-me a "comidinha da mamãe" nos momentos mais difíceis da elaboração desta dissertação.

Finalmente, o agradecimento mais especial vai para a Luciana, pela paciência que teve comigo ao longo deste trabalho e principalmente a que tive com ela, por estar sempre presente no meu percurso académico e nos bons e maus momentos passados.

Resumo

A implementação da interoperabilidade nos Sistemas de Informação Hospitalar (SIH) é cada vez mais um requisito e não uma opção. A Agência para a Integração, Difusão e Arquivo de Informação Médica e Clínica (AIDA) consiste numa plataforma de interoperabilidade hospitalar desenvolvida por investigadores da Universidade do Minho e que se encontra instalada no Centro Hospitalar do Porto (CHP). A AIDA assegura a interoperabilidade entre os SIH e para além disto, assegura também a confidencialidade, integridade e disponibilidade dos dados. A AIDA deve possuir um elevado nível de disponibilidade e um funcionamento eficiente 24 horas por dia. Um pequeno período de paragem poderá trazer graves consequências para a qualidade dos serviços prestados. Esta plataforma possui mecanismos de recuperação e tolerância de falhas, contudo devido à sua elevada importância, é preciso agir antes da ocorrência das falhas, evitando sérios danos. Os processos de monitorização e prevenção de falhas devem ser implementados nos "órgãos vitais" da AIDA, que são as base de dados, máquinas e agentes inteligentes.

Uma vez que a prevenção de falhas em base de dados da AIDA já ter sido alvo de estudo, esta dissertação aborda a monitorização e prevenção de falhas nas máquinas e agentes. Para prever as falhas, foram criados modelos baseados no *Modified Early Warning Score* (MEWS). Este modelo através da recolha frequente dos valores dos sinais vitais, calcula um conjunto de *scores* para determinar o nível de risco a que o paciente está submetido.

Foram desenvolvidos sistemas de monitorização de prevenção para as máquinas e agentes que permitem não só prevenir falhas, mas também observar e avaliar o comportamento destes componentes através de *dashboards* de monitorização. A prevenção de falhas nos agentes foi baseada na frequência com que estes registam as suas atividades nos seus ficheiros *log*, enquanto que para as máquinas a prevenção foi baseada em indicadores de desempenho como a memória e o CPU. Apurou-se que os componentes, em geral, encontram-se com os seus principais recursos bem balanceados e que os sistemas de prevenção desenvolvidos detetaram situações críticas com sucesso, contribuindo para um aumento da integridade e disponibilidade da AIDA do CHP.

Abstract

Implementing interoperability in Health Information Systems (HIS) it is increasingly a requirement rather than an option. The Agency for Integration, Diffusion and Archive of medical information (AIDA) is an interoperability healthcare platform developed by researchers at the Minho University and it is installed in the Centro Hospitalar do Porto (CHP). AIDA ensures the interoperability among HIS and besides this, it also ensures the confidentiality, integrity and availability of data. AIDA should have a high level of availability and an efficient operation 24 hours a day. A small stop period will be able to bring serious consequences for the quality of services provided. This platform has mechanisms for fault tolerance and recovery, however due to its high importance, it is needed to act before the occurrence of faults, avoiding serious damage. The processes of monitoring and failure prevention should be implemented in the AIDA's "vital organs", which are the databases, machines and intelligent agents.

Once the prevention of failures in the AIDA's database has already been studied, this dissertation addresses monitoring and preventing failures in machines and agents. To forecast faults, models were created based on Modified Early Warning Score (MEWS). This model through the frequent collection of values of vital signs, calculates a set of scores to determine the level of risk to which the patient is subjected.

Monitoring and prevention systems were developed for machines and agents, which allow not only prevent faults, but also watch and evaluate the behaviour of these components through monitoring dashboards. The prevention of failures in agents was based on the frequency that they record their activities in their log files, while for the machines prevention was based on performance indicators such as memory and CPU. It was found that components, in general, are with their main resources well balanced and the prevention systems developed detected critical situations successfully, contributing to an increase in the integrity and availability of AIDA in CHP.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

AIDA Agência para a Integração, Difusão e Arquivo de Informação Médica e Clínica. v, xiii, xiv, xvii, 2, 3, 5, 35–41, 46, 55, 56, 58–60, 65, 66, 68, 69, 71–73, 75–80, 82–84, 89, 90, 96, 97, 99–104, 119, 120, 122, 124, 126

BI *Business Intelligence*. 30, 31, 33, 56, 101

CDE *Community Dashboard Editor*. 32, 33, 56, 101
CHP Centro Hospitalar do Porto. v, vii, xiii, xiv, xvii, 2, 3, 5, 19, 23, 29, 35, 39, 40, 65, 66, 68, 71, 73, 75–80, 82, 84, 99, 101, 102
CIM *Common Information Model*. 26
CIPE Classificação Internacional para a Prática de Enfermagem. 8
CLR *Common Language Runtime*. 22
CPU *Central Processing Unit* (processador). v, vii, xiv, 23–26, 28–30, 47, 48, 58, 59, 61, 63, 65–70, 73, 74, 79–81, 83, 90, 91, 100, 128

DICOM *Digital Imaging and Communication in Medicine*. 12
DIS *Department Information System*. 36

GUI *Graphical User Interface*. 22

HL7 *Health Level 7*. 2, 12, 120, 124
HTML *HyperText Markup Language*. 32, 56
HTTP *HyperText Transfer Protocol*. 37

I/O *Input/Output*. 24, 25, 48, 58, 59, 73, 74, 79–81, 100
IEEE *Institute of Electrical and Electronics Engineers*. 9
ISO *Internacional Organization for Standardization*. 9
ISP *Internet Service Provider*. Glossário: ISP. 28

LCIM *Levels of Conceptual Interoperability Model*. xiii, 10, 11
LIS *Laboratory Information System*. 8, 36

MCDT Meios Complementares de Diagnóstico e Terapêutica. 36
MEWS *Modified Early Warning Score*. v, vii, xvii, 4, 16–18, 27–30, 47, 101, 104, 122, 126

PACS *Picture Archive and Communication System*. 8, 12
PCE Processo Clínico Eletrónico. 8, 36
PID Identificador do Processo. Glossário: PID. 51–53

RIS *Radiology Information System*. 8, 36

SAM Sistema de Apoio ao Médico. 8, 36
SAPE Sistema de Apoio à Prática de Enfermagem. 8, 36
SIH Sistemas de Informação Hospitalar. v, 1, 2, 4, 7–9, 13, 14, 21
SIS *Sirurgical Information System*. 9
SMA Sistema Multi-Agente. xiii, 2, 4, 13, 14, 21, 36, 37, 103, 104, 120, 124
SMS *Short Message Service*. 103
SOA *Service-Oriented Architectures*. 22, 36
SONHO Sistema de Gestão de Doentes Hospitalares. 7, 8, 36, 39
SPMS Serviços Partilhados do Ministério da Saúde. 7, 8
SQL *Structured Query Language*. 26, 32

WBEM *Web-Based Enterprise Management*. 26
WMI *Windows Management Instrumentation*. 25–27, 49, 100
WQL *Windows Query Language*. 26, 47, 49–51, 53, 100

XML *eXtensible Markup Language*. 32

Capítulo 1

Introdução

Desde os finais do século passado até aos dias que correm, os sistemas computacionais têm vindo a revolucionar as organizações na forma que prestam os seus serviços e nos meios que utilizam para produzir o que pretendem. Desse modo, o processo de informatização destas organizações têm vindo a prevalecer, tornando estes sistemas uma peça imprescindível para o bom funcionamento e eficácia do trabalho desenvolvido nas mais diversas organizações como empresas, serviços públicos, universidades, hospitais, entre muitas outras.

No campo da saúde, devido ao facto do volume e complexidade de informação ter vindo a aumentar e também ao facto destas organizações armazenarem informações de extrema importância e alta confidencialidade, o processo de gestão da informação tem sido dificultado. Para combater este problema, novas metodologias foram introduzidas nos serviços das unidades de saúde [1, 2]. Surgiram os Sistemas de Informação Hospitalar (SIH), sistemas responsáveis pela aquisição, processamento e apresentação de toda a informação acerca de todos os intervenientes (pacientes, médicos, enfermeiros, entre outros) e todos os serviços (administrativos, clínicos, entre outros). Os principais objectivos dos SIH são garantir uma eficiente prestação dos cuidados de saúde e melhorar a qualidade dos serviços [3, 4].

No entanto, existem vários SIH provenientes de todos os departamentos e serviços do hospital. Estes SIH, para além de serem complexos, são distribuí-

dos e heterogéneos. Por exemplo, equipamentos médicos são produzidos por entidades diferentes e por vezes com objetivos diferentes. Inúmeras aplicações têm sido desenvolvidas para auxiliar os profissionais de saúde, contudo estas aplicações têm um baixo impacto no ambiente hospitalar por manipularem a informação de forma individualizada, não interagindo com outros sistemas. Assim, surge a necessidade de estabelecer um vínculo comunicativo entre todos estes sistemas. Através da interoperabilidade é possível compartilhar todas as informações entre os serviços, evitando informação omnipresente, heterogénea e distribuída. É necessário o desenvolvimento de um processo contínuo e eficiente de integração e interação que deve ter em consideração a escalabilidade, flexibilidade, portabilidade e segurança [4–7].

Várias metodologias têm emergido com o objetivo de implementar sistemas de informações interoperáveis nas unidades de saúde. Entre estas encontra-se a norma *Health Level 7* (HL7) [8], que tal como os Sistemas Multi-Agente (SMAs) apresentam fortes potencialidades de resolver problemas de distribuição e comunicação em prol da interoperabilidade entre os SIH [5, 9].

Dentro deste contexto, surge uma plataforma inteligente e dinâmica com o objetivo de tornar os SIH interoperáveis - Agência para a Integração, Difusão e Arquivo de Informação Médica e Clínica (AIDA). Esta agência fornece agentes inteligentes eletrónicos que são responsáveis por diversas tarefas como a comunicação entre sistemas heterogéneos, arquivo e gestão da informação, bem como a partilha de informação entre todos os sistemas de informação [7, 10–13].

Atualmente a AIDA encontra-se instalada em várias unidades hospitalares de Portugal, entre elas no Centro Hospitalar do Porto (CHP). Devido ao seu papel fundamental nestas unidades de saúde, a AIDA deve garantir a sua confidencialidade (prevenindo que pessoas não autorizadas acedam a informações confidenciais), integridade (impedindo a modificação imprópria dos dados, mantendo a informação incorrupta, inviolável e consistente) e a sua disponibilidade. Esta última é assegurada através de mecanismos de recuperação de falhas, bem como mecanismos que toleram falhas de componentes [14–16].

1.1 Motivação

A AIDA deve possuir um elevado nível de disponibilidade e um bom funcionamento 24 horas por dia. Um pequeno período de paragem poderá trazer graves consequências para a qualidade dos serviços prestados. Neste contexto, e devido a tal elevada importância, é necessário tomar medidas de prevenção de falhas. Deste modo, é possível tomar ações antes da ocorrência da falha, evitando danos e graves consequências.

Uma vez que a prevenção de falhas em base de dados já ter sido objeto de estudo e implementação na AIDA [17–19], surge também a necessidade de prevenir falhas das máquinas onde os agentes da AIDA são executados, bem como prevenir situações de risco dos próprios agentes.

Para a implementação de um sistema de prevenção de falhas numa plataforma de interoperabilidade hospitalar é imprescindível recorrer ao processo de monitorização dos seus componentes. Este facto justifica as duas vertentes que constituem este estudo: a monitorização e a prevenção de falhas. A monitorização das máquinas e agentes envolvidos permite aos administradores do sistema conhecer aprofundadamente o comportamento e carga normal de trabalho dos componentes envolvidos. Assim, para além de possibilitar a implementação de um sistema de prevenção de falhas, a monitorização permite também aos administradores do sistema gerir da melhor forma o balanceamento/aproveitamento dos recursos.

1.2 Objetivos

Para o desenvolvimento e implementação destes sistemas propostos, é crucial que estes sejam testados em ambiente real. Para tal, a plataforma de interoperabilidade hospitalar utilizada foi a AIDA implementada no CHP.

As seguintes questões representam os objetivos que são pretendidos para este projeto. Ao longo de toda esta dissertação, e de forma geral no Capítulo 6, estas questões serão respondidas e exploradas.

Questão 1 Quais são os indicadores de desempenho das máquinas e agentes que devem ser recolhidos para o processo de monitorização?

(secção 3.1)

Questão 2 De que forma esses indicadores podem ser recolhidos? (secção 3.2)

Questão 3 Qual é o modelo de previsão de falhas que pode ser adaptado a este contexto? (secção 3.3)

Questão 4 Quais são as ferramentas a utilizar para a apresentação de dados e para a administração dos sistemas a serem implementados? (secção 3.4)

Questão 5 Quais são as características da carga de trabalho normal das máquinas e agentes envolvidos? (secção 5.2)

Questão 6 Quais são os resultados obtidos dos sistemas de prevenção de falhas? (secção 5.4)

Questão 7 Quais são as medidas que podem ser tomadas para melhorar os sistemas no futuro? (secção 6.2)

1.3 Estrutura do Documento

Esta dissertação encontra-se estruturada em 6 capítulos. Além deste primeiro capítulo introdutório que enquadra o leitor no contexto deste estudo e onde é apresentado as motivações e objetivos, segue-se os seguintes capítulos e apêndices:

Capítulo 2: É apresentado os conceitos e tecnologias envolvidas para a realização deste projeto, ou seja, o Estado da Arte. É explorado o conceito de interoperabilidade assim como a tecnologia de um SMA incorporado numa plataforma de interoperabilidade hospitalar. São também apresentados, de uma forma breve, os SIH. O capítulo aborda os conceitos de monitorização, previsão e prevenção de falhas e culmina com a apresentação de um modelo de previsão de falhas utilizado na medicina denominado *Modified Early Warning Score* (MEWS).

Capítulo 3: Neste capítulo é exposto as metodologias e ferramentas utilizadas para atingir os objetivos deste projeto. Começa pela apresentação da plataforma onde os sistemas foram desenvolvidos - *Microsoft .Net*. No seguimento do capítulo, em cada secção, é dado resposta às quatro primeiras questões exibidas em 1.2 que estão diretamente relacionadas com a metodologia de investigação deste projeto.

Capítulo 4: É apresentada e explorada a plataforma AIDA, com uma perspetiva particular para a AIDA implementada no CHP. Os sistemas de monitorização e prevenção são apresentados neste capítulo, quer a forma como estes foram desenvolvidos, quer as funções que desempenham. Três sistemas são explorados: sistema de monitorização e prevenção dos agentes, das máquinas e sistema de monitorização dos agentes.

Capítulo 5: Aqui são apresentados e interpretados vários resultados deste projeto. Este capítulo encontra-se dividido em 4 secções. Na primeira é apresentado o *website* desenvolvido que permite ter acesso aos *dashboards* de monitorização e a uma página de administração. De seguida são apresentados e interpretados resultados sobre a carga de trabalho das máquinas e agentes da AIDA do CHP. Na terceira e quarta secção são analisados *e-mails* de alerta enviados pelos sistemas desenvolvidos, relativamente aos erros registados pelos agentes e às situações consideradas críticas de máquinas e agentes.

Capítulo 6: Neste capítulo final encontram-se sumarizadas as principais conclusões obtidas através de todo o trabalho desenvolvido. Encontram-se também propostas que podem ser aplicadas neste projeto no futuro para melhorar o desempenho dos sistemas desenvolvidos.

Apêndices

A: Excertos dos *dashboards* de monitorização apresentados na subsecção 5.1.1.

B: Apresentação de trabalhos científicos elaborados em paralelo com este projeto. É apresentado as informações básicas, o resumo de cada trabalho e é também referido a ligação existente entre cada trabalho e esta dissertação.

Capítulo 2

Estado da Arte

Neste capítulo serão abordados vários pontos fundamentais para a realização do projeto. São apresentados vários conceitos e tecnologias tal como interoperabilidade, monitorização, previsão e prevenção de falhas, entre outros. Para começar, é fundamental conhecer os SIH para ser possível desenvolver os sistemas pretendidos.

2.1 Sistemas de Informação Hospitalar

Os SIH proporcionam uma melhor coordenação entre os profissionais de saúde e os equipamentos de uma unidade de saúde. Desta forma, é possível combater a taxa de erros médicos, bem como reduzir custos associados e aperfeiçoar a gestão do sistema informático de uma unidade hospitalar [6].

A nível hospitalar em Portugal, os Serviços Partilhados do Ministério da Saúde (SPMS) (instituto público integrado na administração indireta do Estado) é a entidade detentora dos principais SIH dominantes nos hospitais de Portugal. Entre eles destaca-se o Sistema de Gestão de Doentes Hospitalares (SONHO), desenvolvido no final da década de 80 e responsável pela gestão dos dados administrativos dos doentes. O sistema de gestão de base de dados utilizado pelo SONHO é o *Oracle* [1] e as suas principais funcionalidades são a identificação do utente, agendamento, validação e registo de consulta,

[1] http://www.oracle.com/

registo de vacinação, entre outras. Outros sistemas à margem da SPMS que têm como base a informação registada no SONHO são o Sistema de Apoio ao Médico (SAM) e o Sistema de Apoio à Prática de Enfermagem (SAPE). O SAM é um sistema orientado para a atividade do médico e detém funcionalidades como as prescrições de medicamentos, agenda médica, o Processo Clínico Eletrónico (PCE), identificação do utente e ainda registo/consulta da informação na perspetiva do médico. Por outro lado, o SAPE tem o objetivo de informatizar os registos de enfermagem desenvolvidos e visa o tratamento e a organização da informação processada na documentação de enfermagem. Este sistema possui funcionalidades como o o registo de intervenções (quer as de enfermagem, quer as que resultam das prescrições médicas), utilização da Classificação Internacional para a Prática de Enfermagem (CIPE), entre outras [20].

No entanto, outros SIH (externos à SPMS) são utilizados nas unidades hospitalares, segue-se alguns exemplos [20, 21]:

- *Radiology Information System* (RIS): sistema de informação responsável pela aquisição e análise de imagens médicas. O armazenamento de imagens médicas é garantida através da ligação com o *Picture Archive and Communication System* (PACS);

- *Laboratory Information System* (LIS): sistema de informação relacionado às análises clínicas;

- Sistema de Gestão de Recursos Humanos: sistema de informação responsável pela gestão dos profissionais do hospital de forma a obter o melhor rendimento possível;

- Sistema de Anatomia Patológica: sistema de informação relacionado a biópsias, exames peroperatórios e autópsias;

- Sistema de Gestão de Inventário: sistema de informação responsável pela gestão de matérias-primas, produtos e sobressalentes necessários para manutenção;

- *Sirurgical Information System* (SIS): sistema de informação de cirurgia responsável pela gestão dos blocos operatórios, registo de documentação clínica, entre outras funcionalidades;

Os SIH têm vindo a adquirir cada vez mais influência no campo da saúde e têm crescido quer em qualidade, quer em quantidade. Com essa sobrecarga de informação, investigadores têm vindo a focar-se na qualidade da informação que é registada no sistema [7, 13].

2.2 Interoperabilidade

Interoperabilidade é a capacidade de acessar e utilizar os dados de forma confiável e rápida de diversos sistemas e fontes, a fim de operá-los juntos sem a ocorrência de erros. Segundo o *Institute of Electrical and Electronics Engineers* (IEEE), interoperabilidade é a "capacidade de um sistema ou produto funcionar com outros sistemas ou produtos sem nenhum esforço adicional por parte do utilizador". De acordo com a *Internacional Organization for Standardization* (ISO), outra definição de interoperabilidade, é a capacidade de sistemas independentes trocar informações importantes e iniciar ações uns nos outros, com o objetivo de funcionarem em conjunto para benefício mútuo [10, 22].

Em 1999, Sheth separou o conceito de interoperabilidade em dois tipos [23]:

Interoperabilidade sintática: permite a troca de informação entre diferentes sistemas ou aplicações através de uma gramática. A entidade que envia a informação, codifica-a perante regras sintáticas de uma determinada gramática. Por sua vez, a entidade que recebe a informação, descodifica-a utilizando as mesmas regras de sintaxe.

Interoperabilidade semântica: é garantida quando a informação tem a mesma interpretação para ambas as entidades: a que envia e a que recebe a informação. Não pode haver espaço para ambiguidades.

Existem vários modelos que visam classificar sistemas de informação quanto ao nível de interoperabilidade. No entanto há alguns modelos que se destacam perante os outros, uma vez que utilizam um conjunto de atributos para classificar o sistema quanto à troca de informações, o nível tecnológico e ainda o grau de interoperação [10]. Um dos modelos mais completo e de fácil compreensão é o proposto por Tolk e Muguira [24].

Este modelo é denominado *Levels of Conceptual Interoperability Model* (LCIM) e apresenta mais tipos de interoperabilidade para além da semântica e sintática definidas por Sheth em 1999 [23]. Na Figura 2.1 pode-se visualizar este modelo e os respetivos 7 níveis de interoperabilidade.

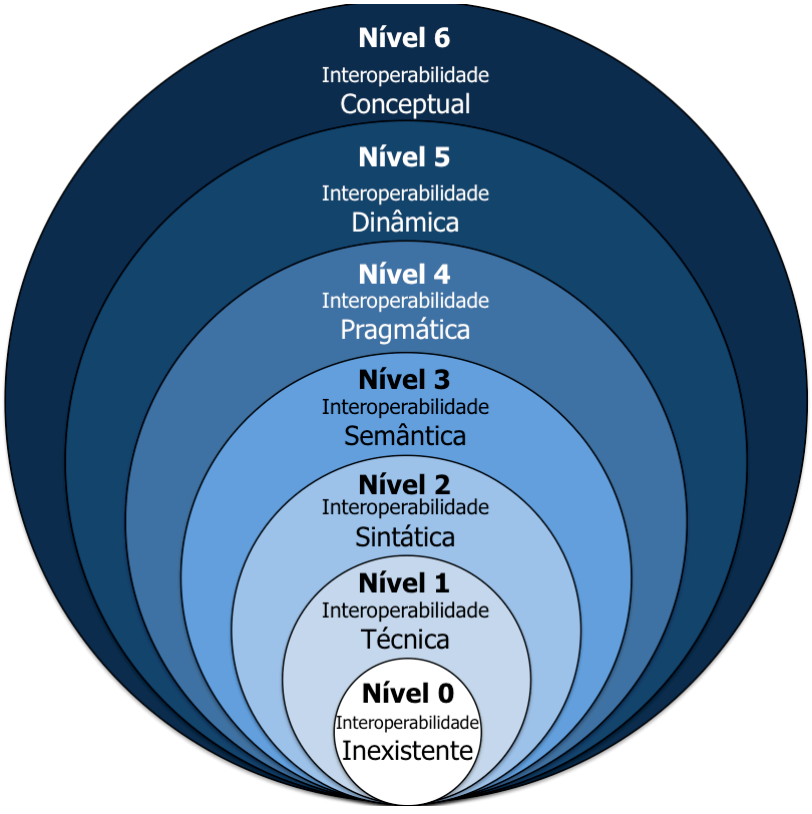

Figura 2.1: Modelo LCIM. Adaptado de Tolk e Muguira [24].

Segue-se as principais características de cada nível do modelo LCIM apresentado na Figura 2.1 [24]:

Nível 0: Sistemas autónomos com **interoperabilidade inexistente**;

Nível 1: Na **interoperabilidade técnica** existe um protocolo de comunicação entre os sistemas, que permite a troca de informações pela rede através de protocolos bem definidos;

Nível 2: Tal como já apresentado, quando se atinge um nível de **interoperabilidade sintática** os sistemas possuem uma estrutura comum, bem definida, no que diz respeito ao formato da informação que é trocada;

Nível 3: Também já apresentada, a **interoperabilidade semântica** é alcançada quando o conteúdo da informação trocada é interpretado da mesma forma em todos os sistemas;

Nível 4: Para obter a **interoperabilidade pragmática** é necessário que os sistemas intervenientes estejam conscientes dos métodos e procedimentos que cada sistema efetua. O contexto onde a troca de informação é realizada é compreendido por todos os sistemas participantes;

Nível 5: Os sistemas que operam sob dados ao longo do tempo consequentemente mudam de estado. Se for atingida a **interoperabilidade dinâmica**, os sistemas são capazes de perceber e tirar proveito das mudanças de estado. O efeito causado pelas trocas de informação é compreendido por todos os sistemas intervenientes;

Nível 6: Para atingir o maior nível segundo o modelo LCIM - **interoperabilidade conceptual** - é imprescindível os sistemas estarem de acordo com as premissas e restrições de cada ambiente real. Para atingir este propósito é necessário documentar os modelos conceptuais através de métodos utilizados na engenharia, para que qualquer engenheiro seja capaz de os compreender. Quando a interoperabilidade conceptual é alcançada, os sistemas participantes podem ser aplicados a ambientes distintos onde as premissas e restrições são diferentes.

Ainda relativamente à Figura 2.1, é importante referir que à medida que o nível sobe, maior é a capacidade de interoperação entre os sistemas. Outro facto que apura-se, tal como sugere a Figura 2.1, é que o nível 6 engloba todos os níveis apresentados, o nível 5 engloba todos os níveis abaixo dele e assim por diante. Posto isto, pode-se afirmar que quando a interoperabilidade conceptual é atingida, todas as características dos outros níveis estão também incorporadas no sistema. Analogamente verifica-se o mesmo para os outros níveis. O nível 3 por exemplo, possui as características dos níveis 1 e 2 (o nível 0 não possui qualquer característica de interoperabilidade) [24].

A interoperabilidade nas últimas décadas tem sido objeto de estudo de vários investigadores nas mais diversas áreas. Na área da saúde vários contributos têm vindo a emergir, nomeadamente a criação de normas para a comunicação entre os sistemas das unidade de saúde. A HL7, formulada por uma organização sem fins lucrativos em 1987, é uma norma que visa fornecer uma *framework* de fácil compreensão e normas relacionadas com o intuito de trocar, integrar, compartilhar e recuperar informação médica eletrónica para habilitar a prática clínica e a gestão dos sistemas, assim como a execução e avaliação dos serviços de saúde. A HL7 constitui um grande contributo para a interoperabilidade nas unidades de saúde, sendo as suas principais contribuições a melhoria da prestação de cuidados e do fluxo de trabalho diário, a redução de informação ambígua e facilita a troca de conhecimento entre profissionais de saúde [8, 25].

Outra norma utilizada em serviços de saúde é a *Digital Imaging and Communication in Medicine* (DICOM). Esta norma promove a comunicação de imagens digitais sem levar em conta a sua origem, ou seja, a imagem pode ser oriunda de qualquer modalidade. Esta norma estandardiza as imagens que são arquivadas e assim o desenvolvimento das PACS é facilitado, bem como a criação de base de dados para o apoio ao diagnóstico. Estas base de dados podem ser interrogadas por utilizadores distribuídos geograficamente desde que tenham permissão para tal. A norma atualmente está estruturada com base na orientação de objetos em que a informação e os métodos estão agrupadas em pacotes, o que permite uma fácil manipulação. Isto trás vantagens como uma fácil manutenção e desenvolvimento de aplicações me-

nos complexas que visam apresentar soluções. O objetivo desta norma é a estabilização em ambiente de rede, a implementação dos requisitos mínimos de conformidade e ainda a permissão da interoperabilidade entre os equipamentos de diferentes fabricantes o que trás enormes vantagens económicas bem como a fácil partilha de informação. A interoperabilidade nas imagens médicas assegura aos profissionais de saúde uma garantia de disponibilidade dessas imagens, bem como informações relacionadas [26, 27].

Apesar de, Carr e Moore terem concluído em 2003 que o nível de interoperabilidade entre os sistemas na maioria das instituições de saúde era muito pouco significante [28], muitos projetos têm sido desenvolvidos a favor da implementação da interoperabilidade nos SIH. Para além do progresso em novas linguagens e normas [29], a tecnologia que abrange os SMAs tem provado uma forte potencialidade no âmbito de prover interoperabilidade nas unidades de saúde [6, 9, 30, 31].

Os agentes inteligentes e a programação orientada a agentes constituem uma tecnologia baseada numa arquitetura distribuída. Os sistemas baseados em agentes têm vindo a alterar o paradigma conceptual da análise de problemas e a contornar obstáculos como a complexidade, distribuição e interatividade dos sistemas. Embora não exista uma definição concreta de um agente inteligente, pode-se afirmar que estes são artefactos computacionais dotados das seguintes propriedades [13, 32]:

- **Autonomia**: atuam de forma independente, sem a intervenção direta de humanos;
- **Reatividade**: são capazes de se aperceberem do ambiente que os rodeia e consequentemente, reagirem a estímulos captados por sensores;
- **Pro-atividade**: possuem capacidades inteligentes para resolver problemas e atingirem os seus objetivos a curto ou longo prazo;
- **Comportamento social**: têm a capacidade de interagir com outros agentes, podendo até mudar o seu comportamento ao responder a uma interação. Os agentes cooperam entre si de forma a atingir um objetivo comum, bem como os objetivos individuais de cada agente.

Um SMA consiste num conjunto de agentes inteligentes que interagem entre si para atingirem os seus objetivos. A autonomia e a pro-atividade dos agentes de um SMA permitem planear e realizar tarefas necessárias para alcançar os objetivos do sistema. Já o comportamento social permite cooperarem e interagirem entre si no SMA, também a fim de atingir os objetivos do sistema. Para usufruir de um SMA de forma interoperável, é imprescindível seguir determinadas normas e desenvolver o sistema da forma mais simples possível.

A utilização dos SMAs com o propósito de tornar os SIH interoperáveis tem vindo a ser cada vez mais explorada, revelando que esta é uma tecnologia com muita potencialidade na área da interoperabilidade [9]. Um SMA é capaz de gerir todo o ciclo de vida de um agente e tornar todos os SIH como um todo. Novos agentes podem ser criados em prol das necessidades que o sistema apresenta. O uso de agentes inteligentes no campo da saúde apresenta um forte potencial para o desenvolvimento de novas práticas de análise e metodologias que não distinguem agentes e humanos. Assim, um SMA pode vir a substituir tarefas que geralmente são realizadas por seres humanos, prosperando os sistemas de informação de uma unidade de saúde.

2.3 Monitorização

Monitorização significa "acompanhar e avaliar", "controlar mediante acompanhamento" ou "olhar atentamente, observar ou controlar com um propósito especial" [33]. Trata-se de um processo contínuo, composto por três componentes: recolha de dados, análise regular dos dados e disseminação dos dados a todas as partes intervenientes [33].

No que toca aos sistemas computacionais das unidades de saúde, a monitorização envolve a avaliação do desempenho dos serviços providenciados por computadores ou sistemas de comunicação (inclusive agentes inteligentes). Para a monitorização de um componente é necessário definir primeiro os objetivos a serem alcançados, para isso é fundamental quantificar a carga que se encontram submetidos e selecionar as métricas apropriadas para os objetivos traçados [34].

Uma característica fulcral destes componentes é que o seu desempenho depende significativamente da carga de trabalho que estão sujeitos. Portanto, o conhecimento e controlo da carga que o componente está submetido é basilar para o seu processo de monitorização. É importante referir que na avaliação do desempenho de um sistema (relativamente à carga de trabalho) não depende apenas da intensidade, mas também da sua natureza. Por exemplo, num servidor *web* nem todos os pedidos são equivalentes: um pedido para o método *get* pode ter um bom desempenho para objetos frequentemente usados e por outro lado, o mesmo servidor *web* pode ter um pobre desempenho ao aceder às bases de dados. Noutros servidores *web* a história pode ser outra. É preciso estar ciente das tarefas de cada componente para os avaliar corretamente [34].

A seleção das métricas a serem consideradas para um processo de monitorização, requer um elevado conhecimento do sistema. Através da união das métricas e dos indicadores da carga de trabalho de um determinado componente, são encontrados todos os parâmetros aos quais é necessário recolher informação [34].

Através de um processo de monitorização eficiente, é possível caracterizar um sistema e reconhecer as suas vulnerabilidades e ameaças a fim de determinar a probabilidade de ocorrer uma falha. Para além disto, os administradores do sistema conseguem implementar sistemas de apoio à decisão para efeitos de balanceamento de recursos [35].

Analisando estas informações, é possível criar uma "pirâmide" de processos a serem seguidos para alcançar a prevenção de uma falha. Na base desta "pirâmide" encontra-se o processo de monitorização, que trata da recolha de todos os dados fundamentais do componente onde se pretende prevenir falhas e permite conhecer as suas principais características. No nível intermediário da "pirâmide" encontra-se a previsão da falha, que pode ser baseada em diversos modelos de previsão e/ou métodos estatísticos. Finalmente, no cume da "pirâmide" encontra-se a prevenção da falha, neste nível os mecanismos e medidas de proteção apropriados são executados.

É de se realçar a potencialidade que a monitorização possui para a previsão e prevenção na área da saúde. A monitorização de indicadores de qua-

lidade de produtos consumo humano, de tecnologias médicas, do exercício profissional na área médica e de riscos ambientais constitui um instrumento que pode contribuir para a identificação de fatores de risco [33].

2.4 Modelo de Previsão de Falhas - MEWS

Pode-se afirmar que a previsão consiste na extração de informação do futuro com base no passado [34]. Os modelos de previsão de falhas têm como objetivo detetar, com antecedência, falhas ou situações anormais que possam vir a desencadear uma falha. A previsão e prevenção de falhas é um tema que tem sido explorado por vários investigadores com o propósito de aumentar a disponibilidade dos sistemas de informação [17–19, 36]. Na área da saúde, a previsão da condição de saúde de um paciente pode ser um fator decisivo para a sua vida. Desta forma, estes modelos auxiliam os médicos na tomada de decisões e permite uma antecipada prestação dos cuidados de saúde adequados [37].

Um destes modelos é o *Modified Early Warning Score* (MEWS). Este modelo toma como ponto de partida perceber que problemas graves de saúde são antecedidos por deteriorações fisiológicas. Dessa forma, procede-se à monitorização dos sinais vitais do paciente, atribuindo *scores* (valores) consoante o desvio aos valores considerados normais. Após medir os sinais vitais pertinentes para este modelo e atribuído os respetivos *scores*, é procedido a soma de todos esses valores. A partir do total obtido o nível de risco do paciente é determinado. Assim, é possível prever um problema grave, como por exemplo, a falência de um órgão [38–40].

Em 1997, surgiu o conceito de *Early Warning Scoring* e os parâmetros analisados nessa altura eram os seguintes [39, 41]:

1. **Temperatura**: grau de calor no organismo humano medido através de termómetros, o seu valor normal é 37°C;

2. **Frequência cardíaca**: medida da atividade cardíaca, normalmente expressa em batimentos por minuto (*bpm*);

3. **Pressão arterial sistólica**: valor mais elevado da pressão arterial de um ciclo cardíaco, é atingida quando ocorre a sístole do ventrículo esquerdo, medida em $mmHg$;

4. **Estado da consciência**: estado obtido através de um teste neurológico que consiste na resposta de vários fatores como a voz e a dor, por parte do paciente;

5. **Frequência respiratória**: Número de inspirações e expirações por unidade de tempo, medida em min^{-1}.

Contudo em 1999, investigadores após estudos realizados acrescentaram alguns sinais vitais a este modelo para o aperfeiçoar. Estava assim definido o modelo MEWS. Os sinais vitais adicionados foram [39, 42, 43]:

6. ***Output* urinário**: Medição do *output* de urina em $ml.kg^{-1}.h^{-1}$, a diminuição deste valor pode indicar sérios problemas de saúde;

7. **Oxigénio no sangue**: concentração de oxigénio no sangue. O nível de concentração de oxigénio no sangue está diretamente relacionado com o bom funcionamento do coração e dos pulmões.

Na Tabela 2.1 encontram-se os *scores* associados a cada sinal vital do MEWS.

Para a utilização deste modelo ter sucesso, a recolha dos valores dos sinais vitais deve ser contínua seguido do seu armazenamento. Deste modo, é possível analisar o histórico do paciente ao longo do tempo [41, 42].

Para classificar o estado do paciente extrai-se os 7 *scores* relativos a cada parâmetro do MEWS. Após esta extração, somam-se esses 7 valores, obtendo-se o *score* total. Este valor representa o estado do paciente no momento em que a leitura dos parâmetros foi feita. Para interpretar e classificar este estado, são seguidas as seguintes diretrizes [40, 41]:

- Quando um dos parâmetros possui o *score* de 2, o paciente deve ser alvo de observação frequente;

Tabela 2.1: MEWS *scores*. Adaptado de [41].

MEWS *scores*	**3**	**2**	**1**	**0**	**1**	**2**	**3**
Temperatura ($^{\circ}C$)		< 35.0	35.1-36.0	36.1-38.0	38.1-38.5	> 38.6	
Frequência cardíaca (bpm)		< 40	41-50	51-100	101-110	111-130	> 131
Pressão arterial sistólica ($mmHg$)	< 70	71-80	81 - 100	101 - 199		> 200	
Frequência respiratória (min^{-1})		< 8		8-14	15-20	21-29	> 30
Oxigénio no sangue (%)	< 85	85-89	90-93	> 94			
Output urinário ($ml.kg^{-1}.h^{-1}$)	0	< 0.5					
Estado da consciência		Agitação Confusão		Alerta	Responde à voz	Responde à dor	Não responde

- Se o *score* total (soma dos *scores* de todos os sinais vitais do paciente) for igual a 4, ou então existir um aumento de 2 valores em relação ao estado anterior, o paciente requer atenção do médico urgente;
- Se o *score* total for maior que 4, o paciente corre risco de vida.

Este modelo é dotado de várias vantagens no processo de monitorização e prevenção de problemas graves de saúde do paciente, nomeadamente:

- Permite definir prioridades em relação às intervenções a realizar [39];
- O processo de monitorização e observação do paciente é enriquecido pois fornece informações sobre as tendências fisiológicas do organismo [41];
- Por ser baseado em critérios quantitativos, auxilia na tomada de decisão médica [42];
- Prevê situações em que o paciente necessita de internamento nos cuidados intensivos [41].

Nos dias de hoje, o MEWS é utilizado em hospitais em alguns países como a Austrália, Reino Unido e Espanha. No caso de Portugal, encontra-se

implementado no Centro Hospitalar do Barlavento Algarvio e é objeto de estudos de investigação no CHP [17–19, 37, 39].

Capítulo 3

Metodologia de Investigação

Este projeto é desenvolvido para plataformas de interoperabilidade hospitalar baseadas num SMA. Assim, em seguimento do projeto que envolve bases de dados hospitalares [17–19], os alvos de monitorização e prevenção deste projeto são os agentes, bem como as máquinas envolvidas nestas plataformas.

Uma plataforma de interoperabilidade hospitalar possui grandes responsabilidades no que diz respeito às comunicações entre os SIH. Portanto, esta deve possuir um alto nível de disponibilidade. Este projeto propõe a integração de sistemas de prevenção de falhas nas máquinas da plataforma, assim como a prevenção de falhas em agentes. Desta forma, estes sistemas de monitorização e prevenção contribuem para obter este nível de disponibilidade da melhor forma possível. Por outras palavras, estes sistemas permitem vigiar e reparar antecipadamente os componentes com atividade anormal propícios a falhas, prevenindo graves danos no fluxo de trabalho de uma plataforma de interoperabilidade hospitalar e também possíveis consequências graves na qualidade dos serviços de saúde prestados.

A metodologia de investigação utilizada neste projeto foi a *action research*[1]. Este tipo de metodologia é baseada na pesquisa e resolução de problemas de forma progressiva, por outras palavras, à medida que o trabalho é desenvolvido para determinado problema, analisa-se os resultados temporá-

[1] pesquisa ação

rios e aplica-se as ações necessárias para atingir os objetivos pretendidos [44].

Os sistemas relativos a este projeto foram desenvolvidos na plataforma *Microsoft .NET*. Esta *framework* visa interligar todos os aspetos que envolvem uma aplicação: informação, sistemas, dispositivos, ferramentas para o desenvolvimento de aplicações e a *web* [45]. *.NET framework* é constituída por duas componentes [46]:

- *Common Language Runtime* (CLR): mecanismo reponsável pela execução das aplicações;
- Biblioteca *.NET framework*: inclui um vasto leque de classes testadas e reutilizáveis. O utilizador pode facilmente usufruir destas classes aquando o desenvolvimento de suas aplicações.

Através da *.NET framework* é possível desenvolver diversas aplicações e serviços para o sistema operativo *Windows*. Esta plataforma é dotada de várias características úteis que o utilizador pode usufruir [46]:

- Gestão de memória;
- Interoperabilidade entre várias linguagens: todas as linguagens são compiladas para uma linguagem intermédia que por sua vez é compilada pelo CLR quando a aplicação é executada;
- Vasta biblioteca de classes;
- Desenvolvimento de *frameworks* e tecnologias distintas: aplicações e serviços *web*, *Service-Oriented Architectures* (SOA), aplicações *Graphical User Interface* (GUI), entre outras.

Para além das vantagens já evidenciadas para escolha da plataforma *.NET* para desenvolvimento destes sistemas de monitorização e prevenção, outra razão que levou a esta opção foi o facto de todo o sistema computacional onde este projeto foi testado e implementado ser baseado no sistema operativo *Windows*. Na secção 4.1 será apresentado o caso de estudo deste projeto.

Para armazenar toda a informação consequente do processo de monitorização, foi utilizado uma base de dados *Oracle* [47]. É dos sistemas de base

de dados mais populares e mais utilizados em todo o mundo e também é utilizado no CHP, local onde este projeto foi desenvolvido e implementado (secção 4.1).

Este capítulo dará resposta às questões de 1 a 4 (ver secção 1.2), levando em conta tudo que já foi dito até aqui.

3.1 Indicadores de Desempenho (Questão 1)

Como mencionado na secção 2.3, para criar um processo de monitorização é necessário primeiramente selecionar os parâmetros que devem ser quantificados relativos aos componentes que se pretende monitorizar. Entre estes parâmetros, deve conter a carga a que estes estão submetidos e ainda as métricas necessárias para atingir os objetivos pretendidos [34].

Pretende-se monitorizar as máquinas e os agentes de uma plataforma de interoperabilidade hospitalar. Relativamente às **máquinas** foram selecionados três parâmetros relacionados com a carga a que estão submetidas:

- **Percentagem de CPU livre**: percentagem do tempo que o processador (CPU) gasta ao executar o *thread* do *system idle process* [48];
- **Percentagem de memória livre**: percentagem da memória física disponível para os processos que se encontram em execução. A memória livre é resultado da soma da lista de modo de suspensão, lista livre e lista zero [49];
- **Percentagem de disco livre**: rácio entre o espaço disponível e o espaço total utilizável de todas as unidades de disco lógico instaladas na máquina [50].

As máquinas onde os agentes são executados são a fonte de recursos para que estes realizem as suas tarefas. Portanto, é fundamental monitorizar estes componentes. Para além de prevenir e detetar falhas dos agentes, este processo de monitorização permite também aos administradores do sistema

controlar e conhecer melhor o comportamento do sistema. Deste modo, é possível balancear e tirar maior partido dos recursos disponíveis. Por exemplo, quando um novo agente é criado, é preciso escolher em qual máquina este será colocado. Através do processo de monitorização, o administrador do sistema conhecerá o comportamento normal de cada máquina, e sabendo que tipo de recurso o novo agente consumirá mais, irá colocá-lo na máquina mais apropriada.

Em relação aos **agentes**, pode-se classificá-los em dois tipos quanto ao tempo que se encontram na lista de processos das máquinas a realizar as suas tarefas: **contínuos** e **momentâneos**. Por outras palavras, agentes momentâneos são aqueles que são executados durante um curto período de tempo (o processo relativo ao agente é iniciado, executado durante alguns segundos e finalmente o processo é terminado). Por outro lado, os agentes contínuos estão permanentemente na lista de processos das máquinas (longo tempo de execução). No caso de estudo deste projeto (secção 4.1), este tipo de agente desencadeia um ou mais processos (dependentemente das suas necessidades), enquanto que os agentes momentâneos apenas desencadeiam um processo de cada vez.

Assim, quatro parâmetros correspondentes à carga foram selecionados para o processo de monitorização dos agentes. O quarto parâmetro varia levando em conta o que foi dito no parágrafo anterior:

- **Percentagem de CPU**: percentagem do tempo usado por todos os *threads* do(s) processo(s) gerados pelo agente [51];
- **Memória**: conjunto de trabalho total do(s) processo(s) gerados pelo agente. O conjunto de trabalho é o total de páginas de memória que o(s) processo(s) utilizaram recentemente [51];
- ***Input/Output* (I/O) de dados por segundo**: quantidade de dados, quer de entrada quer de saída, gerados pelo(s) processo(s) em questão [51];
- **Tempo de execução ou número de processos**: aos agentes momentâneos é medido o intervalo de tempo em que o processo foi executado.

Quanto aos contínuos é medido o número de processos existentes em dado momento.

Para além dos indicadores de desempenho relacionados com a carga que os agentes estão submetidos, para este processo de monitorização e posteriormente de previsão e prevenção, é também fundamental recolher informações acerca da **frequência de atividade**. Isto significa a frequência com que cada agente regista a sua atividade (normalmente o registo é efetuado em ficheiros *log*), pode também ser traduzido como o intervalo de tempo que normalmente o agente fica em repouso até atuar. É importante referir que para esta medição é necessário conhecer o sistema, para ser possível desencadear o processo de monitorização [34].

Eis a resposta para a **Questão 1**:

- Máquinas: percentagem de CPU, memória e disco livre;
- Agentes: percentagem de CPU, memória, I/O de dados por segundo, tempo de execução ou número de processos e ainda a frequência de atividade.

3.2 Recolha dos Indicadores de Desempenho (Questão 2)

Após a seleção dos indicadores de desempenho, surge a necessidade de encontrar e eleger as ferramentas mais apropriadas para o processo de recolha de dados.

Uma vez ter sido optado desenvolver estes sistemas na *framework .NET*, é uma escolha sábia explorar todas as classes e propriedades que esta plataforma dispõe. Visto que estes sistemas de monitorização e prevenção abrangem diversas máquinas de uma plataforma de interoperabilidade hospitalar, o acesso remoto é fundamental para o processo de recolha dos indicadores de desempenho pretendidos. Por esta razão e também pelo facto da vasta coleção de classes que possui, foi elegido a tecnologia *Windows Management*

Instrumentation (WMI) para a recolha da maioria dos indicadores de desempenho.

Com o objetivo de promover normas para a gestão de sistemas através de redes heterogéneas, a organização *Distributed Management Task Force, Inc.* criou a arquitetura *Web-Based Enterprise Management* (WBEM) [52]. WBEM é um conjunto de normas de gestão que foram desenvolvidas para unificar o processo de gestão dos sitemas computacionais distribuídos. A WBEM fornece um poderoso conjunto de ferramentas baseadas em normas que facilita a troca de informação entre tecnologias distintas [53]. O WMI é a abordagem da *Microsoft* para a WBEM. Esta tecnologia usufrui da norma *Common Information Model* (CIM) para representar componentes como sistemas, aplicações, redes, dispositivos e até mesmo ficheiros. Para fornecer uma *framework* comum, a norma CIM possui um conjunto básico de classes que definem uma série de objetos. Estes objetos WMI podem ser acedidos por programas na máquina local ou então remotamente, se o programa possuir as credenciais necessárias. Esta tecnologia providencia um vasto conjunto de serviços, incluindo a recuperação de informação e um sistema de notificação de eventos. Para além destas vantagens, o WMI é de fácil utilização pois utiliza uma linguagem baseada em *queries*: *Windows Query Language* (WQL). Esta linguagem é um subconjunto da linguagem *Structured Query Language* (SQL) utilizada em bases de dados [52, 54, 55].

Para a recolha de informação dos indicadores de desempenho relativos às máquinas e de quase todos indicadores relativos aos agentes (exceto o tempo de execução, número de processos e frequência de atividade), foi utilizado *queries* WQL. Segue-se um exemplo de uma *query* utilizada.

```
1 SELECT PercentIdleTime FROM
      Win32_PerfFormattedData_PerfOs_Processor
  WHERE name='_total'
```

Código 3.1: WQL *query* para a percentagem de CPU livre nas máquinas.

No caso dos indicadores de desempenho tempo de execução e número de processos, estes foram quantificados nos programas desenvolvidos. Tendo em consideração o caso de estudo deste projeto (secção 4.1), para a recolha da

frequência de atividade dos agentes foi usada a classe *Directory* da biblioteca disponibilizada pela plataforma *.NET*. A partir desta classe, e com conhecimento do caminho físico dos ficheiros *log* (localizados nos discos de cada máquina e onde se encontram registados os eventos realizados pelos agentes), é possível recolher informações acerca da última data de modificação do ficheiro. Contudo, é fundamental proceder à criação de *mapped drives* na máquina onde o programa de monitorização e prevenção é executado. Para tal, um ficheiro *batch* cria as respetivas *mapped drives* apontadas para os discos de todas as máquinas que se pretende que sejam monitorizadas. Finalmente, a frequência de atividade é calculada a partir da diferença entre os dois intervalos mais recentes.

Em síntese, a resposta para a **Questão 2** é a tecnologia WMI e a classe *Directory* da *framework .NET* com o auxílio de um simples ficheiro *batch*.

3.3 Modelos de Previsão de Falhas (Questão 3)

A construção de um sistema de prevenção de falhas só é possível após a elaboração de um modelo que possibilite a previsão de falhas. Para este projeto foi elaborado dois modelos de previsão de falhas, ambos baseados no modelo MEWS já apresentado na secção 2.4. Contudo, apesar de terem o mesmo ponto de partida, a abordagem feita para os agentes e máquinas foi distinta.

3.3.1 Previsão de Falhas dos Agentes

Tendo como base o caso de estudo (a ser apresentado na secção 4.1) e toda a estrutura que possui relativamente à gestão e registo dos ficheiros *log*, a previsão de falhas dos agentes foi baseada na frequência de atividade de cada agente. Uma vez que este parâmetro encontra-se em minutos, não faz sentido manter a parte negativa da tabela MEWS (Tabela 2.1). A segunda adaptação foi acrescentar um *score* (valor 4), uma vez que estamos a levar em conta para efeitos de previsão um parâmetro. Esta decisão permite ao sistema

categorizar melhor o estado da atividade do agente. A terceira e última adaptação foi definir os limites dos *scores* baseados nos percentis 85 a 97,5. Inicialmente foi optado pelos percentis 80 a 95, atribuindo o máximo *score* ao percentil 95. Esta decisão foi tomada com base no método do percentil 95, bastante utilizado na faturação dos ISPs e *websites* [56]. Contudo, após uma fase de testes inicial, chegou-se à conclusão que seria melhor adaptar os percentis para 85 a 97,5 de forma a aperfeiçoar a deteção de situações de risco. Como se é de esperar para o cálculo de percentis, antes de classificar a situação do estado do agente é imprescindível recolher primeiro um conjunto de valores significativos acerca da frequência de atividade de cada agente. Na Tabela 3.1 encontra-se o modelo de previsão de falhas dos agentes.

Tabela 3.1: *Scores* do Modelo de Previsão de Falhas dos Agentes

Scores	**0**	**1**	**2**	**3**	**4**
Frequência de atividade (min)	$\leq p85$	$]p85, p90]$	$]p90, p95]$	$]p95, p97,5]$	$> p97,5$

Quanto à gravidade das situações pode-se classificá-las como normal, pouco grave, grave, muito grave e crítica, respetivamente ao *score* atribuído de zero a quatro. Se o *score* for menor que quatro, um aviso aparecerá na *dashboard* de monitorização (ver secções 3.4 e 5.1.1). No caso de o *score* for igual a quatro, o agente encontra-se num estado crítico e será necessário tomar ações preventivas. Neste caso um *e-mail* de alerta é enviado ao administrador deste sistema.

3.3.2 Previsão de Falhas das Máquinas

No caso da previsão de falhas das máquinas, a abordagem foi diferente. Desta vez, três parâmetros foram selecionados: percentagem de CPU, memória e disco livre. A primeira adaptação também foi a remoção da parte negativa do modelo MEWS, uma vez que se trata de percentagens. Desta vez como se irá recorrer à soma de todos os *scores* obtendo um total, a escala de *scores* permaneceu com o máximo de três. Por se tratar de percentagens

livre, é normal que as situações com menores valores representem situações de maior perigo. Inicialmente também elaborou-se um modelo baseado em percentis, no entanto os resultados obtidos não correspondiam ao estado de gravidade que a máquina realmente estava. Por exemplo, o percentil mais grave em relação à percentagem de CPU livre rondava 90-95%. Isto acontece pois o comportamento normal das máquinas do caso de estudo é de baixo consumo de CPU e assim inúmeras situações eram identificadas como críticas quando, na realidade, não o eram. Para resolver esta situação foi decidido estabelecer intervalos fixos para efeitos de cálculo dos *scores*. Como já mencionado na secção 2.3, a avaliação de um sistema não depende apenas da intensidade, mas também da sua natureza. Portanto, foram debatidos com especialistas da área e do CHP, quais os intervalos fixos que deveriam ser impostos, tendo consciência da natureza de cada máquina. Este modelo de previsão deve permitir ao administrador alterar os intervalos fixos quer para cada métrica, quer para cada máquina individualmente sempre que necessário. Na Tabela 3.2 encontram-se os intervalos fixos por defeito para todas as máquinas, podendo estes serem alterados.

Tabela 3.2: *Scores* do Modelo de Previsão de Falhas das Máquinas

Scores	**0**	**1**	**2**	**3**
CPU livre (%)	≥ 50	$]50, 25]$	$]25, 10]$	< 10
memória livre (%)	≥ 25	$]25, 15]$	$]15, 10]$	< 10
disco livre (%)	≥ 25	$]25, 15]$	$]15, 10]$	< 10

Neste modelo, recorre-se à soma de todos os *scores* para cada máquina. Tal como no modelo MEWS, se o *score* total for quatro ou então ocorrer um aumento de dois valores do *score* total em relação à última medição, a máquina encontra-se num estado que requer observação atenta do administrador. Este acontecimento será devidamente identificado nos *dashboards* de monitorização (ver secções 3.4 e 5.1.1). Numa situação crítica, quando o total é superior a quatro, um *e-mail* é enviado para o administrador do sistema para que este tome as ações preventivas necessárias. Após um período de testes, também resolveu-se atribuir o estado crítico quando um dos parâme-

tros se encontra no *score* máximo. Visto que se tratam de recursos básicos e imprescindíveis para o bom funcionamento de computadores, chegou-se à conclusão que esta medida faz todo o sentido.

Resumidamente, a resposta à **Questão 3** passa pela adaptação do modelo MEWS conforme o caso do componente que se pretende prever falhas. No caso dos agentes o parâmetro utilizado é a frequência de atividade e os limites são baseados em percentis. Por outro lado, para as máquinas os limites são fixos (sujeitos a alteração de forma individual ou geral) e são utilizados três parâmetros: percentagem de CPU, memória e disco livre.

3.4 Ferramentas para Apresentação de Dados e Administração dos Sistemas (Questão 4)

De forma a facilitar o acesso à apresentação de dados recolhidos do processo de monitorização, bem como à administração dos sistemas desenvolvidos, optou-se pela criação de um *website*. Para a apresentação de dados é fundamental o uso de gráficos de forma a facilitar a leitura e ajudar a compreender melhor o comportamento dos componentes monitorizados. Para tal, foi utilizado uma ferramenta de *Business Intelligence* (BI) - *Pentaho Community* [2]. Quanto à administração dos sistemas foi desenvolvido uma página *web* na plataforma *.NET*. Por último, para efeitos de apresentação e discussão de resultados da vertente da prevenção deste projeto, foi recorrido a programas escritos em *Google Apps Script* para analisar todos os *e-mails* de alerta recebidos. Nas próximas secções cada uma destas ferramentas serão apresentadas e será explicado a importância destas neste projeto.

[2] http://community.pentaho.com/

3.4.1 Administração dos Sistemas

Depois de desenvolver os sistemas de monitorização e prevenção, surgiu a necessidade de administrá-los, de preferência de uma forma que possibilite um fácil acesso e um fácil manuseamento para o utilizador. Portanto, foi desenvolvido um *website* na plataforma *.NET* que visa monitorizar os próprios sistemas de monitorização e prevenção desenvolvidos neste projeto. Deste modo, o utilizador pode saber se estes sistemas se encontram operacionais. No caso do modelo de previsão de falhas dos agentes, a página informa o administrador quais são os agentes que estão a ser vigiados. Já no caso do modelo em relação às máquinas é possível alterar os limites fixos para cada parâmetro ou máquina, tal como foi referido na secção 3.3.2.

3.4.2 Ferramenta de *Business Intelligence*

O processo de monitorização desencadeia um elevado volume de dados que necessitam de ser interpretados corretamente, o que não consiste numa tarefa fácil. Para manipular e visualizar os dados referentes aos sistemas desenvolvidos neste projeto, recorreu-se a uma ferramenta de BI.

Este termo - *Business Intelligence* - foi criado por Howard Dresner no ano de 1989 com o objetivo de descrever um conjunto de componentes (desde conceitos, métodos, processos e ferramentas) que através das novas tecnologias melhoram a tomada de decisões no mundo do negócio [57]. O processo de BI decorre ao longo de vários passos. O primeiro é adquirir e integrar todos os dados oriundos de diversas fontes numa *data warehouse*. Após esta integração, é possível efetuar análises, compreender a dinâmica empresarial e ainda obter informações cruciais para a tomada de decisões [57–59]. Estas análises podem ser interpretadas através de várias ferramentas: produção de relatórios, *dashboards*, *data mining* e ferramentas de planeamento e modelação. O uso de ferramentas de BI traz inúmeras vantagens para as organizações, entre as quais a consolidação de dados, mais quantidade e qualidade de informação e maior apoio para tomadas de decisões [58].

Em relação a este projeto era fundamental encontrar uma ferramenta de BI *open-source*. Averiguou-se que a ferramenta que deveria ser escolhida

era a *Pentaho Community* [60, 61]. Esta ferramenta é constituída por várias aplicações baseadas na linguagem *Java*, o que permite a sua utilização em sistemas operativos distintos. Para este projeto para além do módulo principal desta ferramenta - *bi-platform* - foi adicionado um módulo extra: *Community Dashboard Editor* (CDE), dos vários que podem ser acrescentados. O CDE foi desenvolvido e encontra-se a cargo da *Webdetails* [3]. Este módulo baseado em *HyperText Markup Language* (HTML) permite ao utilizador através de uma plataforma de fácil uso, editar *dashboards* que tornam a análise de dados mais clara e completa. Este módulo ainda apresenta a vantagem de preencher os gráficos e tabelas dos *dashboards* a partir de várias fontes, nomeadamente as mais importantes: ficheiros *eXtensible Markup Language* (XML) e *queries* SQL.

Deste modo, o *Pentaho Community* juntamente com o módulo CDE formam a solução para a construção de *dashboards* de monitorização dos sistemas que foram desenvolvidos. As *dashboards* elaboradas serão apresentadas na subsecção 5.1.1 e alguns excertos encontram-se no Apêndice A.

3.4.3 Análise dos *E-mails* de Alerta

O sistema de alerta adotado para os sistemas de monitorização e prevenção desenvolvidos é baseado em *e-mails*. Futuramente este sistema pode ser aperfeiçoado como será discutido na secção 6.2. Quando se iniciou a fase de testes dos sistemas desenvolvidos, vários *e-mails* de alerta foram enviados para uma conta *Google* criada para este projeto. De imediato emergiu a necessidade de analisar toda a informação alusiva aos *e-mails* recebidos.

Através da poderosa ferramenta *Google Apps Script* [4] foi possível atingir esse propósito. *Google Apps Script* é uma linguagem baseada em *JavaScript* que contém um vasto leque de classes que permitem criar aplicações *web* que conseguem aceder e manipular diversas aplicações *Google* (calendário, agenda, *e-mail*, folhas de cálculo, documentos de texto, mapas e *sites*). As aplicações são desenvolvidas no *browser* e armazenadas e executadas nos

[3] http://www.webdetails.pt/
[4] http://www.google.com/script/start/

servidores da *Google* [62, 63].

Deste modo, foi desenvolvido um programa para analisar a quantidade de *e-mails* de alerta, bem como o conteúdo de cada um. Toda essa informação foi transferida para uma folha de cálculo onde foi possível efetuar a análise dos resultados obtidos do sistema de alerta e consequentemente proceder-se à construção de gráficos. No capítulo 5 serão mostrados os resultados obtidos.

De forma sucinta, as ferramentas *Google Apps Script*, a plataforma *.NET* e ainda a ferramenta de BI *Pentaho Community* juntamente com o módulo CDE, são a resposta para a **Questão 4** que visa apresentar e administrar toda a informação gerada pelos sistemas de monitorização e prevenção desenvolvidos neste projeto.

Capítulo 4

Sistemas de Monitorização e Prevenção

Neste capítulo será inicialmente apresentado o caso de estudo onde este projeto foi desenvolvido. Após a compreensão do "terreno" onde os sistemas de monitorização e prevenção são executados, são apresentados cada sistema desenvolvido individualmente, expondo as suas aptidões, limitações e os seus objetivos.

4.1 Caso de Estudo: AIDA - CHP

A AIDA [30, 64, 65] é uma plataforma criada por investigadores da Universidade do Minho e do CHP e já se encontra implementada em várias instituições de saúde portuguesas.

A AIDA pode ser descrita como uma agência que fornece trabalhadores electrónicos inteligentes, mais conhecidos por agentes, que para além de terem um comportamento pró-ativo, providenciam diversos serviços tais como: a comunicação entre vários sub-sistemas através do envio e receção de informações provenientes de relatórios médicos, imagens médicas, prescrições de medicamentos, entre outras; armazenamento e gestão da informação; resposta corretas a pedidos em tempo real [66].

Para garantir a disseminação e integração da informação gerada nas uni-

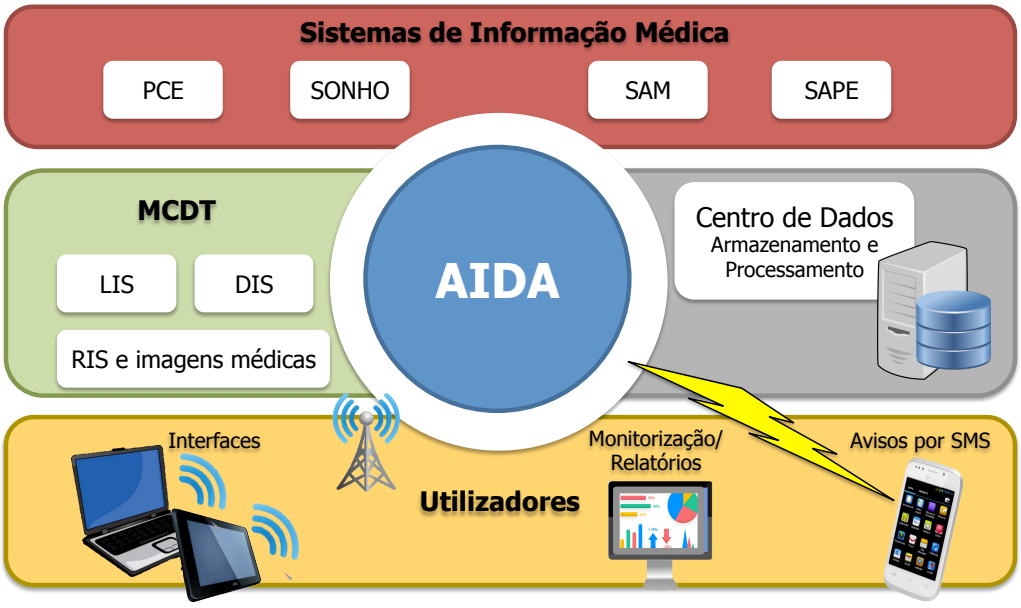

Figura 4.1: Ilustração do modelo conceptual da plataforma AIDA. Adaptado de [66].

dades de saúde, esta plataforma utiliza ferramentas de integração como o SOA e SMAs [10]. O objectivo da AIDA, tal como o nome indica, é integrar, disseminar e arquivar um grande conjunto de dados provenientes de várias fontes (departamentos, serviços, unidades, computadores, equipamentos médicos, entre outros) dos diferentes sistemas inseridos nas unidades hospitalares [7, 66].

Na Figura 4.1 podemos analisar todo o modelo conceptual da plataforma AIDA. Esta permite a comunicação entre todos os sistemas de informação médica (PCE, SONHO, SAM, SAPE) e todos sistemas dos Meios Complementares de Diagnóstico e Terapêutica (MCDT) (LIS, *Department Information System* (DIS), RIS e imagens médicas) através de mensagens entre agentes inteligentes. Para além disto, a AIDA possui ferramentas que possibilita a comunicação dos seres humanos com todo o sistema através de serviços *web* (interfaces, relatórios, avisos por SMS) tal como evidenciado na Figura 4.1 [67].

4.1.1 Sistema Multi-Agente da AIDA - CHP

Todos os agentes da AIDA comunicam entre si através de mensagens. Na Figura 4.2 é apresentado os diferentes tipos de agentes que constituem a arquitetura do SMA da AIDA.

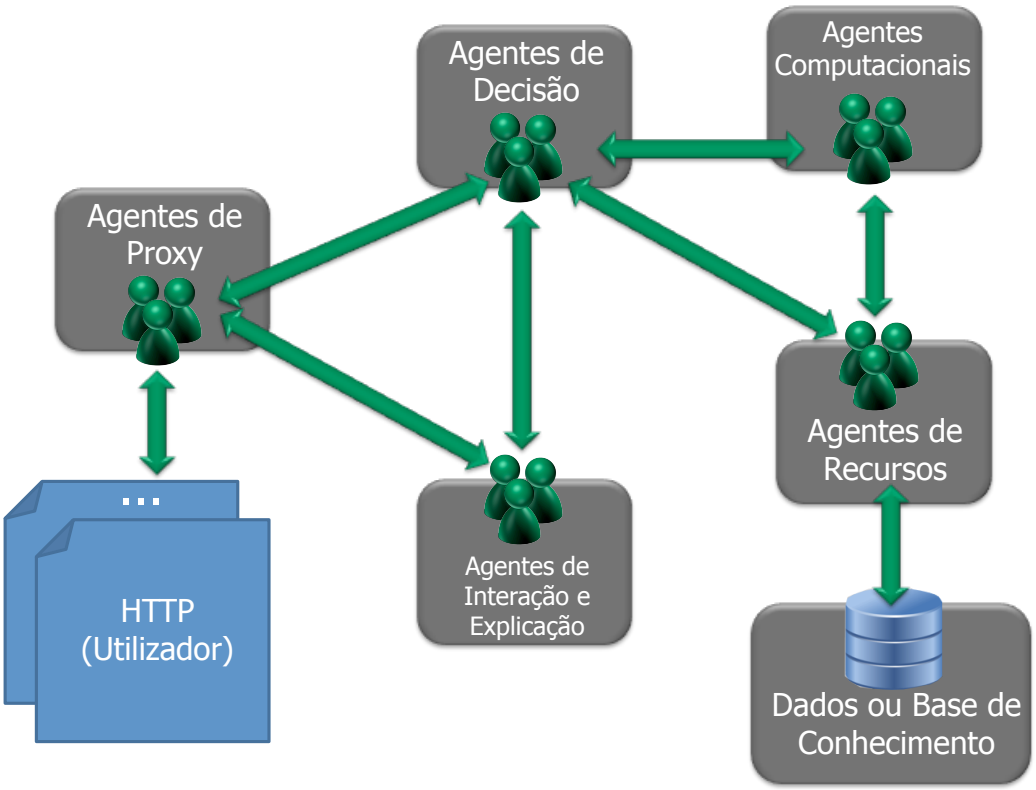

Figura 4.2: Arquitetura do SMA da AIDA. Adaptação de [13].

- **Agentes de *proxy***: tem o papel de estabelecer a ligação entre os utilizadores e o sistema. Através deste agentes, os utilizadores conseguem obter informações/explicações pretendidas, tomar decisões e visualizar resultados. A interface utilizada para estabelecer esta ligação são páginas *web* sob o protocolo *HyperText Transfer Protocol* (HTTP);
- **Agentes de decisão**: funciona como mediador ao aceitar as tarefas provenientes dos agentes de *proxy*. Têm a capacidade de desagrupar

essas tarefas em sub-tarefas, enviando-as para os agentes computacionais para serem processadas e ainda recebendo depois os respetivos resultados;

- **Agentes computacionais**: processam as tarefas requeridas pelos agentes de decisão e enviam para estes os respetivos resultados;
- **Agentes de recursos**: possuem o conhecimento necessário para aceder a dados específicos ou bases de conhecimento;
- **Agentes de interação e explicação**: baseados em processos argumentativos, recebem dados e respondem aos pedidos dos agentes de *proxy* e de decisão.

4.1.2 Máquinas e Agentes da AIDA-CHP

Para a realização deste projeto foi imprescindível compreender as funções de cada agente e em que máquinas estes são executados. A partir de informações armazenadas nas base de dados da AIDA e também através de reuniões com um dos administradores do sistema, foi possível fazer o levantamento das principais funções de cada grupo de agentes. O nome dos agentes são baseados em apenas algarismos e os seus grupos são distintos pelo primeiro algarismo de cada agente (por exemplo, o agente 60 pertence ao grupo de agentes começados pelo algarismo 6). Na Tabela 4.1 pode-se constatar as principais funções, de forma generalizada, de cada grupo de agentes bem como as máquinas em que são executados e ainda a respetiva classificação dos agentes como contínuos ou momentâneos (ver secção 3.1).

Ao analisar a Tabela 4.1, constata-se que cada grupo de agentes corresponde apenas a um tipo de funções. Comprova-se também que cada grupo possui exclusivamente agentes contínuos ou agentes momentâneos. Também é possível retirar a conclusão que existe uma certa organização na distribuição dos agentes pelas máquinas, por exemplo, a máquina hsa-aida01 apenas executa agentes do tipo contínuo. Contudo, o facto de vários grupos utilizarem várias máquinas, é para que haja um bom balanceamento na distribuição dos agentes pelas máquinas. O agente 59 é uma exceção no seu grupo, é o

Tabela 4.1: Categorização dos agentes ativos na AIDA - CHP.

Grupo	Agentes	Máquina(s)	Funções	Contínuos/ Momentâneos
1	12, 13, 15, 17, 18, 101	hsa-aida01	Partilha de ficheiros	Contínuos - SILAgAna.exe
3	31, 35, 36, 37, 38, 39, 391	hsa-aida02 hsa-aida04 hsa-aida08 hsa-aida09	Processamento (Verificação, produção de relatórios)	Momentâneos
4	46, 460, 49, 490, 491, 492	hsa-aida02 hsa-aida08	Faturação para o SONHO	Momentâneos
5	50-57, 59*, 501-...	hsa-aida02 hsa-aida08*	Pedidos, Processamento*	Contínuos - SILAgPed.exe, momentâneo*
6	60-65, 609, 620	hsa-aida04 hsa-aida08 hsa-aida09	Processamento	Momentâneos
7	701-730	hsa-aida01	Anexos	Contínuos - SILAgAna.exe
9	90-97, 901-914	hsa-aida01 hsa-aida02 hsa-aida08 hsa-aida09	Transferências	Contínuos - SILAgTrn.exe

único do grupo 5 que é momentâneo, executado na máquina hsa-aida08 e também o único que desempenha funções de processamento. Este agente deveria estar incluído no grupo 6. Na última coluna da Tabela 4.1 é também possível visualizar o nome do executável que todos os agentes contínuos utilizam para cada grupo. Relembrando o referido na secção 3.1, os agentes contínuos têm o mesmo nome do ficheiro executável (consoante o grupo) e os agentes momentâneos tem o nome do executável como "SILAgXX.exe", sendo "XX" o número do agente. Um último facto que deve ser referido acerca da Tabela 4.1, é que nesta apenas constam os agentes considerados ativos pelo sistema de monitorização e prevenção dos agentes baseados nos ficheiros *log* (secção 4.2).

Desta forma, este projecto abrange um total de cinco máquinas da AIDA

implementada no CHP. Nas próximas secções será explicado como foram desenvolvidos os sistemas de monitorização e prevenção dos agentes e máquinas.

4.2 Sistema de Monitorização e Prevenção dos Agentes (com base nos *logs*)

Como já mencionado na subsecção 3.3.1, o modelo de prevenção de falhas de cada agente é baseado na frequência de atividade. Este parâmetro é medido em minutos e significa o intervalo de tempo entre a data de registo da atividade atual do agente, no seu respetivo ficheiro *log*, e a data da atividade anterior a essa.

Todos agentes da AIDA que se pretende que sejam monitorizados têm uma tarefa em comum quando terminam de efetuar todas as outras: registar a sua atividade nos respetivos *logs*. Estes ficheiros encontram-se em todas as máquinas onde os agentes são executados e estão todos na mesma diretoria. Nesta diretoria de *logs* encontram-se várias pastas onde cada uma pertence a um agente. Dentro de cada uma das pastas encontram-se os ficheiros *log* associados a cada agente.

Cada agente pode ter até dois tipos de ficheiros *log*: um com registo da atividade normal do agente e outro com possíveis erros que podem ter ocorrido durante a sua execução. Os nomes dos ficheiros são confinados a uma expressão regular consoante o seu tipo:

- ***[0-9]{6}.log*** - expressão regular para o nome dos ficheiros *log* de registo da atividade;
- ***[0-9]{6}e.log*** - expressão regular para o nome dos ficheiros *log* de erros.

Os seis dígitos permitidos por ambas expressões regulares dizem respeito à data do ficheiro, sendo que os dois primeiros dígitos são referentes ao ano, os próximos dois ao mês e finalmente os últimos ao dia. Por exemplo, o ficheiro

log do dia 19 de setembro de 2013 tem como nome "130919.log". Desta forma, é possível distinguir os dois tipos de ficheiros *log* e selecionar o ficheiro que corresponde ao dia atual.

O sistema de monitorização e prevenção dos agentes apresentado nesta secção é denominado *MoniLogs* e encontra-se apresentado na Figura 4.3.

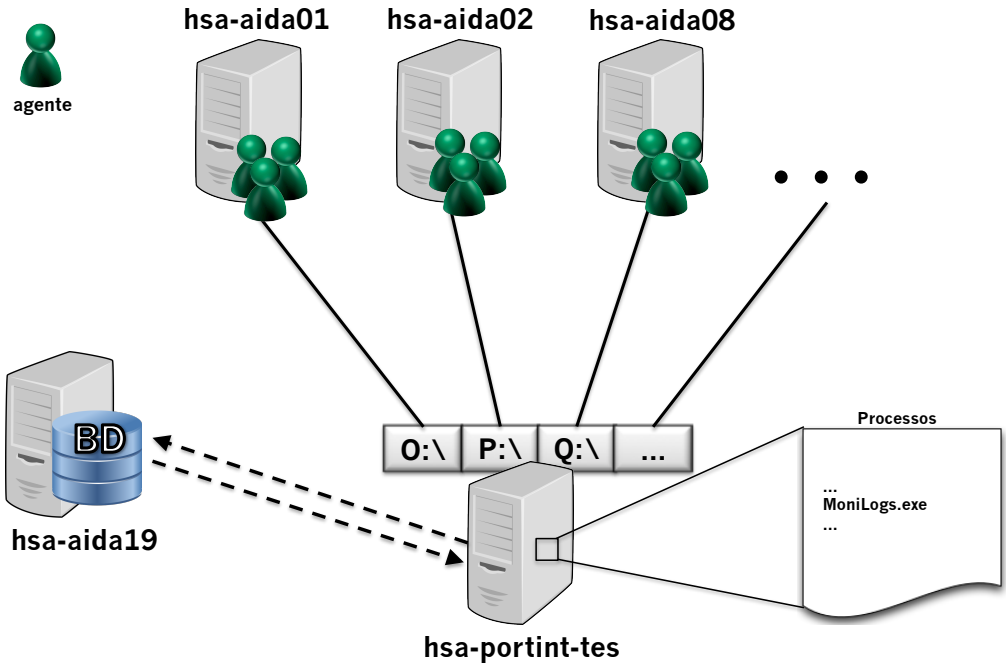

Figura 4.3: Arquitetura de *MoniLogs*.

Tal como sugere a Figura 4.3, a aplicação *MoniLogs* é executada na máquina hsa-portint-tes onde se encontra várias *mapped drives*. A partir de cada uma destas é possível aceder aos ficheiros *log* que se encontram nas máquinas onde os agentes são executados. Atualmente *MoniLogs* analisa *logs* nas máquinas hsa-aida01, 02, 04, 08 e 09. Para guardar toda a informação recolhida é utilizado uma base de dados situada na máquina hsa-aida19, independente da base de dados da AIDA. A decisão de utilizar esta base de dados passa pelo facto de não se pretender sobrecarregar a base de dados da AIDA e também, se esta última falhar, este sistema de monitorização e prevenção continua ativo.

A aplicação *MoniLogs* atua de 10 em 10 minutos para verificar o estado da frequência de atividade de cada agente. Uma vez que os *logs* de erros podem existir, esta aplicação também deteta quando novos erros ocorrem. Para registar a frequência de atividade, este sistema calcula a diferença entre a última data de modificação do ficheiro *log* e data de modificação anterior, que se encontra registada na base de dados.

Apesar de existir vários agentes que atuam em intervalos inferiores a 10 minutos, este facto não afeta o bom funcionamento deste sistema. A frequência de atividade destes agentes rondará os 10 minutos ou pouco menos se estes agentes registarem realmente com muita frequência. O que acontece é que não é possível detetar todas as datas de modificação do ficheiro *log* e sendo assim, *MoniLogs* calcula o intervalo entre as duas últimas datas que se verificou quando foi executado 10 minutos atrás, e no momento que é executado. A realidade não é totalmente representada, mas como o objetivo deste sistema é detetar se o agente encontra-se ou não parado, este facto não interfere neste processo. Pois se o agente efetivamente estiver parado, tal acontecimento será detetado por *MoniLogs* após a frequência de atividade do agente ultrapassar o percentil 97,5. Neste tipo de caso onde os agentes registam com muita frequência, a frequência de atividade geralmente anda um pouco acima dos 10 minutos. No capítulo 5 este facto será constatado.

Quando a máquina hsa-portint-tes era reiniciada, *MoniLogs* desligava-se, portanto foi necessário implementar uma tarefa do sistema operativo *Microsoft Windows* para arrancar *MoniLogs* quando a máquina é ligada.

No esquema da Figura 4.4 está representado o processo que *MoniLogs* segue para os ficheiros *logs* de registo que estão na diretoria de cada agente.

Este processo inicia-se ao verificar se existe algum ficheiro *log* que coincida não só com a expressão regular que está na Figura 4.4 mas também com a data do dia atual. Caso a resposta seja negativa, a variável boleana "ativo" passa a zero e a aplicação passa para a próxima diretoria que diz respeito a outro agente. Esta variável classifica os agentes quanto à sua atividade nos sistemas de monitorização e prevenção desenvolvidos neste projeto com base nos registos dos ficheiros *log* por parte dos agentes. No caso positivo, é verificado se a última data de modificação do ficheiro é mais recente do

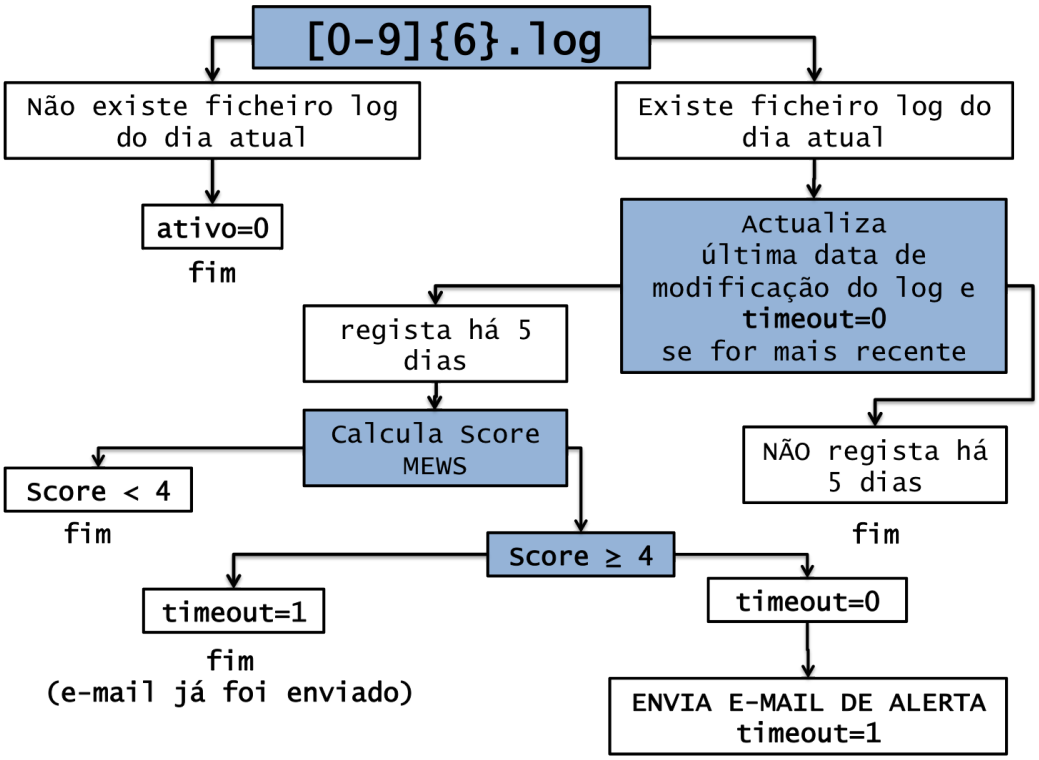

Figura 4.4: Processo seguido por *MoniLogs* relativamente aos ficheiros *logs* de registo.

que a que já se encontra na base de dados na máquina hsa-aida19. Se for diferente, a variável boleana "*timeout*" passa a zero se estiver a um. Esta variável representa o estado atual do agente quanto à sua atividade, se for um, significa que o agente está num estado crítico. Se a data for diferente significa que o agente voltou à sua atividade normal e é por isso que passa a zero neste momento. Para dar continuação ao processo de verificação se determinado agente encontra-se ou não numa situação crítica, é averiguado se o agente que está a ser analisado já regista a sua frequência de atividade há cinco dias. A razão pela qual é feita esta verificação é que, como já mencionado anteriormente, é imprescindível monitorizar antes de prever falhas em componentes. Desta forma, é recolhido o comportamento da frequência de atividade de cada agente durante cinco dias e apenas após esse período estes agentes encontram-se vigiados por este sistema de prevenção. Prosseguindo

este processo, e já com os ditos registos, procede-se ao cálculo do *score* da situação atual da frequencia de atividade do agente com base na Tabela 3.1 exibida na subsecção 3.3.1. Como já referido, se o *score* for maior ou igual a quatro o agente encontra-se num estado crítico. Para não haver réplicas de *e-mails* de alerta é averiguado a variável "*timeout*", se esta estiver a um, quer dizer que o agente ainda se encontra numa situação crítica e um *e-mail* de alerta já foi enviado para o administrador do sistema, caso ainda esteja a zero, o *e-mail* é enviado com o percentil 97,5 atual, a última data de modificação do ficheiro e ainda o seu conteúdo. Neste momento a variável boleana "*timeout*" passa a um. O processo é repitido para todas as diretorias de todos os agentes das máquinas hsa-aida01, 02, 04, 08 e 09.

Quanto aos *logs* de erros, *MoniLogs* segue o processo exibido na Figura 4.5. Este processo inicia de forma semelhante ao apresentado na Figura 4.4, é verificado se existe o ficheiro *log* de erros que coincida com a respetiva expressão regular e que seja do dia atual. Caso positivo, a aplicação compara a última data de modificação do ficheiro com a última data registada na base de dados referente aos erros. Se estas duas datas forem idênticas significa que nenhum erro novo surgiu, pois sempre que um erro novo surge este é registado no *log* de erros, alterando a última data de modificação do ficheiro. Por outro lado, se as datas forem diferentes ou então não existir ainda nenhuma data de erros na base de dados, significa que existe um novo erro e nesta situação um *e-mail* de alerta é enviado ao administrador do sistema com o conteúdo do ficheiro que contém o registo dos erros.

A aplicação *MoniLogs* para além de ser provida destes procedimentos de monitorização e prevenção, possui também mecanismos de persistência. Sempre que o envio de um *e-mail* de alerta falha, seja por que razão for, *MoniLogs* tenta novamente enviar o mesmo *e-mail* passado um curto período de tempo, se voltar a falhar, tenta novamente. Este mecanismo não perturba o funcionamento dos processos de análise dos ficheiros *log* pois é lançado num *thread*. Deste modo esta aplicação garante que o administrador recebe as notificações de alerta, apesar de poderem estar um pouco atrasadas. Outro mecanismo de persistência ainda mais importante, é o facto de *MoniLogs* ser uma aplicação que consegue manter a monitorização e prevenção dos agentes

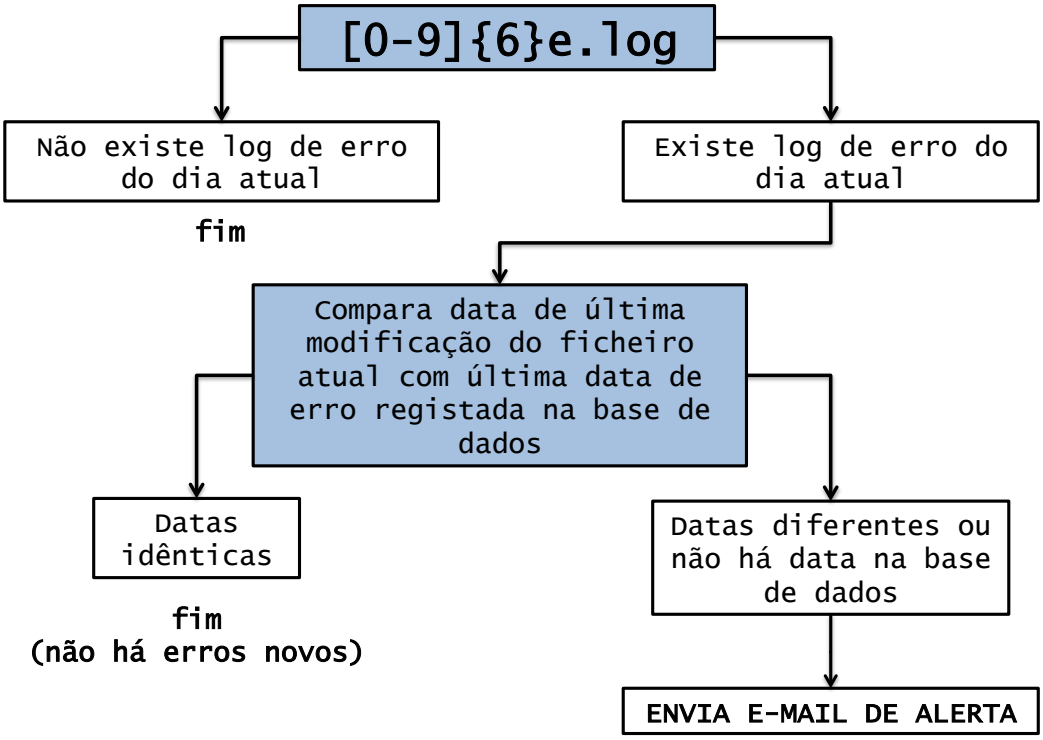

Figura 4.5: Processo seguido por *MoniLogs* relativamente aos *logs* de erros.

mesmo quando há uma falha na conexão à base de dados. Para tal, quando a aplicação se encontra em funcionamento normal com conexão à base de dados, são constantemente guardados e atualizados todos os percentis relativos a todos agentes. Quando a conexão falha, para além de avisar o administrador, a aplicação segue um procedimento sem base de dados baseado nos processos das Figuras 4.4 e 4.5. As variáveis boleanas são asseguradas pela aplicação e para efeitos do cálculo de *scores* são utilizados os percentis resultantes da última atualização que tiveram no momento em que *MoniLogs* possuía ligação à base de dados. Todos os registos que deveriam ser efetuados na base de dados são guardados num ficheiro e quando a conexão volta a estar ativa, todos esses registos são efetivados. Este mecanismo garante a disponibilidade do sistema de monitorização e prevenção dos agentes nos momentos de emergências. As desvantagens para o sistema quando se recorre a este mecanismo são os factos de os percentis não serem atualizados e não

ser possível monitorar a frequência de atividade dos agentes em tempo real nos *dashboards* de monitorização.

4.3 Sistema de Monitorização e Prevenção das Máquinas

A aplicação apresentada nesta secção foi denominada *MoniMaqs* por ser responsável pela monitorização e prevenção de falhas das máquinas. Na Figura 4.6 é possível visualizar o funcionamento desta aplicação.

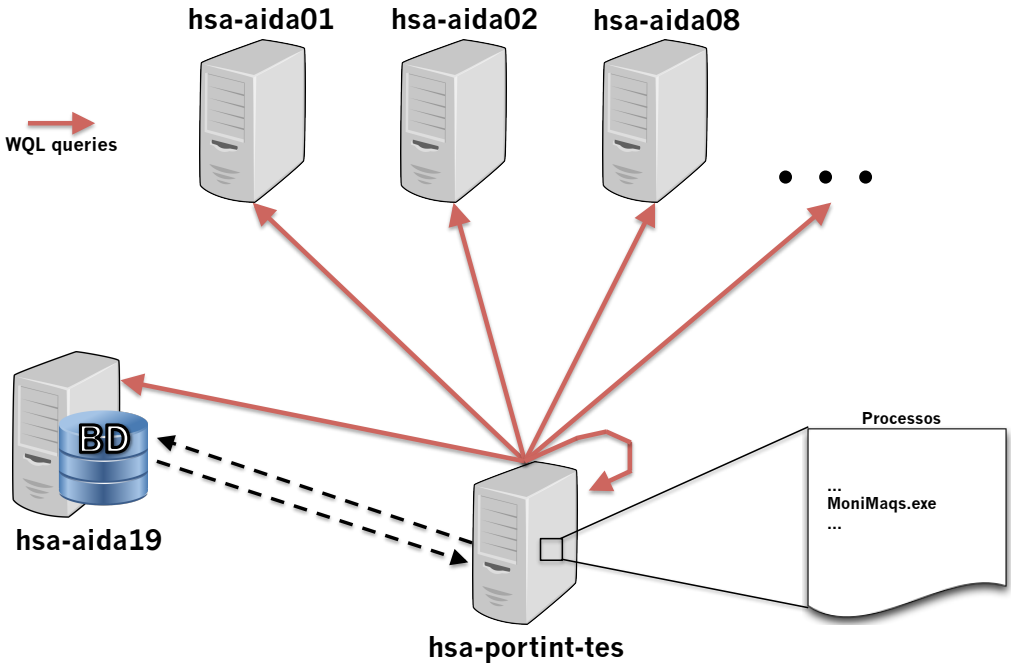

Figura 4.6: Arquitetura de *MoniMaqs*.

MoniMaqs é uma aplicação que também é executada na máquina hsa-portint-tes e utiliza a base de dados situada na máquina hsa-aida19, tal como mostra a Figura 4.6. As máquinas que este sistema monitoriza e previne falhas são: hsa-aida01, 02, 04, 08 e 09 onde se encontram os agentes da plataforma AIDA. É importante referir que ao prevenir falhas das máquinas

onde os agentes são executados, por inerência está-se também a prevenir falhas nos próprios agentes. Para além destas máquinas, *MoniMaqs* também monitoriza e previne falhas na hsa-aida19, por ser a máquina que alberga a base de dados deste projeto e possui as *dashboards* de monitorização, e a própria hsa-portint-tes, onde estão a ser executados todos os sistemas desenvolvidos neste projeto e também possui o *website* que exibe as *dashboards* de monitorização e a página de administração.

Esta aplicação é executada de 10 em 10 minutos sendo que nos primeiros cinco é recolhido sistematicamente valores dos três indicadores de desempenho em relação às máquinas, já mencionados na secção 3.1. Para recolher esses valores são utilizados WQL *queries*, tal como sugere a Figura 4.6, semelhantes ao código 3.1 (secção 3.2). Para todas as máquinas, com a exceção da hsa-portint-tes, é necessário que a aplicação conheça as credenciais necessárias para poder executar as WQL *queries* remotamente. Após a recolha de valores durante cinco minutos é calculado as respetivas médias para cada indicador, que são inseridas na base de dados e também são utilizadas para efeitos do cálculo do *score* total.

Com base no modelo de prevenção de falhas nas máquinas da Tabela 3.2 apresentado na subsecção 3.3.2, é calculado o *score* atual de cada máquina. Tal como em MEWS, se houver uma subida de dois valores em relação ao *score* total anterior, o *score* total é fixado no valor quatro que representa uma situação grave. Como já esperado, se a soma de todos os *scores* dos três indicadores for superior a quatro, um *e-mail* de alerta é enviado ao administrador do sistema a avisar que a máquina em questão encontra-se numa situação crítica. No *e-mail* de alerta para além de ter o *score* total, os valores atuais de CPU, memória e disco livre da máquina e os respetivos limites definidos na Tabela 3.2, tem também a lista de processos atuais com a respetiva utilização de CPU e memória. Este sistema também considera uma situação crítica quando um dos seu indicadores de desempenho encontra-se com o *score* máximo, ou seja, um *score* igual a três, também é enviado um *e-mail* de alerta nesta situação. Como já mencionado no Capítulo 3, esta decisão foi tomada por se tratar de recursos básicos e de papel fundamental para o bom funcionamento das máquinas. Se um destes indicadores estiver num estado

crítico deve-se considerar que a máquina também o está e medidas de prevenção devem ser tomadas. Tal como é feito em *MoniLogs*, todos os *e-mails* de alerta não são enviados mais que uma vez, evitando uma sobrecarga na caixa de correio do administrador com informação redundante. A aplicação *MoniMaqs* também é dotada do mesmo mecanismo de persistência no envio de *e-mails* de alerta de *MoniLogs*.

Antes de recolher qualquer valor dos indicadores de desempenho de cada máquina, *MoniMaqs* precisa de assegurar que estas estão ligadas. Para este fim, esta aplicação faz um teste *ping* para não só testar se as máquinas estão ligadas mas também para verificar se é possível estabelecer uma ligação pela rede. Se este teste der negativo, é enviado um *e-mail* de alerta ao administrador do sistema e não são recolhidos os valores dos indicadores de desempenho referentes àquela máquina. Tal como feito para *MoniLogs*, foi criada uma tarefa no sistema operativo *Microsoft Windows* para arrancar *MoniMaqs* sempre que a máquina hsa-portint-tes é reiniciada.

É importante referir que máquinas podem ser adicionadas e removidas a este sistema e os limites da Tabela 3.2 podem ser alterados através de uma página de administração que será introduzida no próximo capítulo.

4.4 Sistema de Monitorização dos Agentes

O sistema apresentado nesta secção é responsável pela monitorização dos indicadores de desempenho dos agentes. Tal como já apresentado em 3.1, os indicadores são: percentagem de CPU, memória, I/O de dados por segundo e tempo de execução ou número de processos (agente momentâneo ou contínuo, respetivamente). Esta aplicação denominada *MoniAgs* tem como função principal recolher os valores desses indicadores sistematicamente e encontra-se explicitada na Figura 4.7.

MoniAgs é executada na máquina hsa-portint-tes e utiliza a base de dados situada na máquina hsa-aida19, tal como as outras aplicações já apresentadas. Esta aplicação é executada uma vez por dia, dura 24 horas e monitoriza todos os agentes catalogados como ativos por *MoniLogs* nas máquinas hsa-aida01, 02, 04, 08 e 09. Sempre que a aplicação é executada, *MoniAgs*

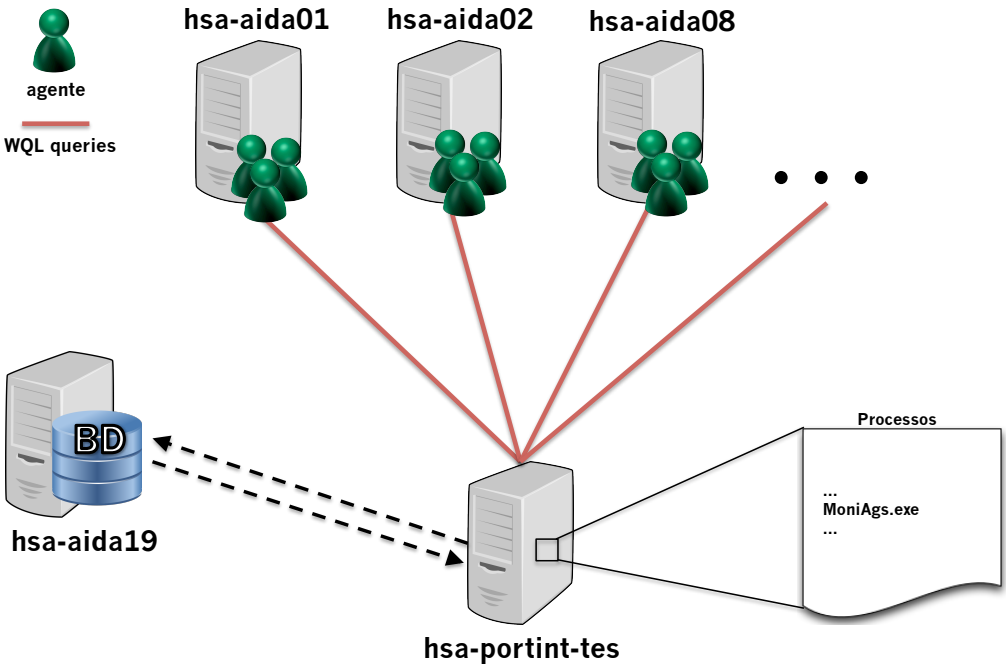

Figura 4.7: Arquitetura de *MoniAgs*.

atualiza os nomes dos ficheiros executáveis correspondentes a cada agente na base de dados e estes nomes são atribuídos com base na Tabela 4.1. Os nomes destes ficheiros são fundamentais pois é a partir deles que é possível distinguir executáveis na lista de processos de cada máquina.

A recolha dos indicadores de desempenho só é possível durante o período de tempo que os agentes estão a ser executados e é também realizada por WQL *queries* como evidencia a Figura 4.7. Como é imprevisível saber quando cada agente atua, implementou-se um sistema de notificação de eventos através da tecnologia WMI. No excerto de código 4.1 encontra-se um exemplo da *query* que é criada para que *MoniAgs* fique alerta à criação ou remoção do executável do agente 60.

```
SELECT * FROM __InstanceOperationEvent WITHIN 2 WHERE
    TargetInstance ISA 'Win32_Process' AND TargetInstance.Name
    = 'SILAg60.exe'
```

Código 4.1: WQL *query* para notificação de eventos em relação ao executável do agente 60.

Sempre que ocorre um evento da classe "__InstanceOperationEvent" é verificado se o evento trata-se da criação do processo ou então da sua remoção. Deste modo, é possível recolher os indicadores de desempenho de determinado agente enquanto este está a ser executado.

Como já mencionado, os agentes neste projeto foram divididos em dois grupos: contínuos e momentâneos. *MoniAgs* necessita de distingui-los para os tratar da forma mais apropriada, para tal foi possível recorrer a expressões regulares:

- ***[a-zA-Z]*([0-9]+)$*** - expressão regular para identificar executáveis de agentes momentâneos. No início do nome pode ter zero ou mais letras e termina com um ou mais algarismos.

- ***[a-zA-Z]+*** - expressão regular para o nome dos executáveis de agentes contínuos. Esta expressão regular não é usada para testar os nomes dos executáveis mas sim a expressão regular anterior. Se o nome que está a ser testado não der um resultado positivo, trata-se de um agente contínuo.

Uma vez que é possível separar os agentes a partir dos nomes dos respetivos executáveis, diferentes processos são seguidos quando ocorre um evento de criação ou remoção do processo. Começando pelos agentes momentâneos, encontra-se na Figura 4.8 o ciclo de funcionamento destes agentes em *MoniAgs*.

Após a identificação de que trata-se de um agente momentâneo através da expressão regular que está na Figura 4.8, é criado um objeto designado "*watcher*". Este objeto proveniente da classe "*Management*" da biblioteca da *framework .NET*, permite identificar quando um determinado processo

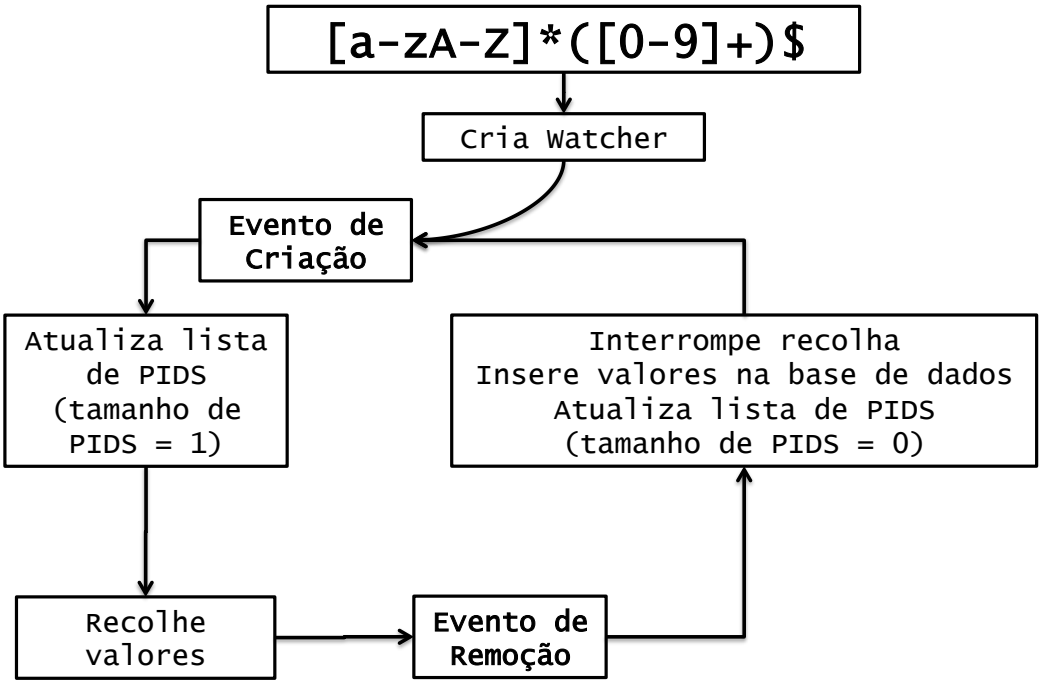

Figura 4.8: Ciclo de funcionamento de *MoniAgs* em relação aos agentes momentâneos.

é iniciado ou parado através de uma WQL *query* como a apresentada no código 4.1. Uma vez criado este objeto, *MoniAgs* fica à espera de um evento de criação. Quando este ocorre, é guardado numa lista todos os PIDs em relação aos executáveis do agente em questão. Como se trata de um agente momentâneo, esta lista apenas pode conter um PID ou então nenhum. Caso contrário trata-se de um situação irregular. Como acabou de ser criado um processo, a lista de PIDs possui um elemento. Nesse momento o agente está a ser executado e é recolhido os valores dos indicadores de desempenho sistematicamente até surgir o evento de remoção através de WQL *queries* semelhantes às usadas em *MoniMaqs*. Quando ocorre o evento de remoção, o processo de recolha é interrompido, a média de todos os valores é introduzida na base de dados e a lista de PIDs é atualizada, passando a conter nenhum elemento. *MoniAgs* fica à espera de um novo evento de criação para reiniciar o ciclo.

No que toca aos agentes contínuos o processo é mais complexo, pois pode existir vários executáveis com o mesmo nome e estes têm de ser tratados como um grupo. Na Figura 4.9 encontra-se esse processo que será explicado nas próximas linhas.

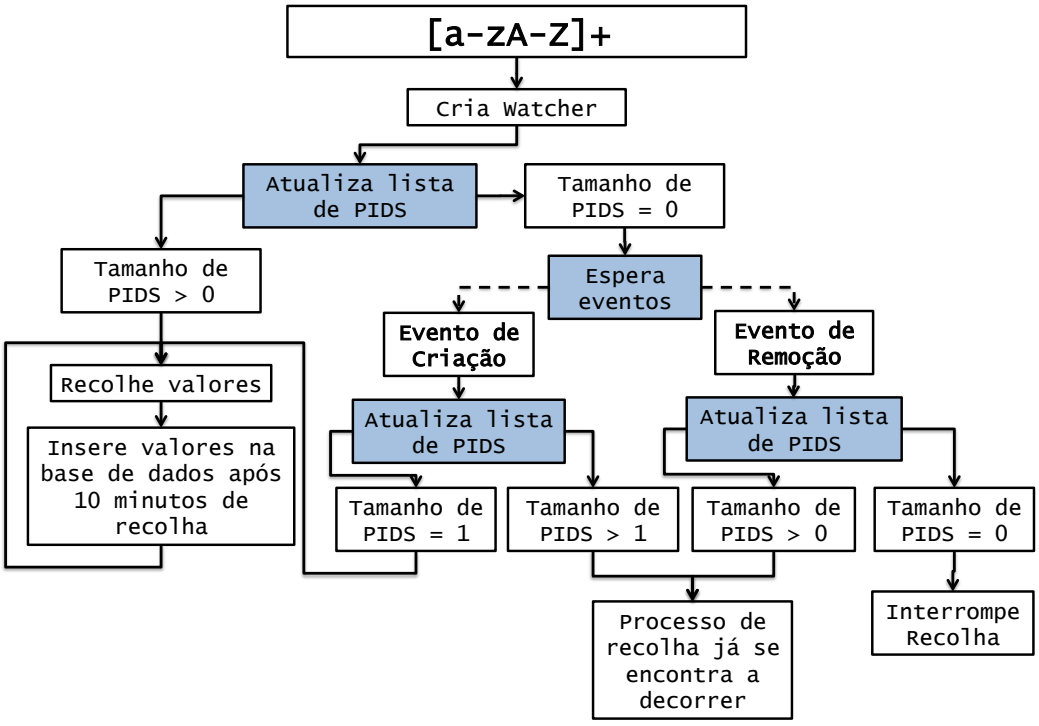

Figura 4.9: Funcionamento de *MoniAgs* em relação aos agentes contínuos.

Tal como a Figura 4.9 demonstra, após de *MoniAgs* identificar que se trata de um agente contínuo, é criado um "*watcher*" tal como no processo dos agentes momentâneos. Este "*watcher*" deteta quando determinado executável, com um nome específico, é iniciado ou parado.

Como neste caso trata-se de agentes que estão continuamente a ser executados, quando o "*watcher*" é criado, muito provavelmente os executáveis já se encontram na lista de processos a serem executados. Como sugere a Figura 4.9 é atualizada a lista de PIDs, desta vez esta lista pode conter vários elementos. Duas situações podem suceder: a lista de PIDs ou está vazia ou já contém elementos.

No caso de esta lista não estar vazia, significa que já se encontram executáveis com o nome que está a ser analisado a serem executados, procede-se assim ao processo de recolha através de WQL *queries* e após dez minutos de recolha, a média de cada indicador de desempenho é guardada na base de dados. O processo de recolha começa novamente e assim inicia-se um ciclo de recolha e introdução de dados.

Por outro lado se a lista de PIDs estiver vazia, a aplicação fica a espera de um evento de criação. Quando este ocorre, a lista é atualizada e como estava vazia naturalmente o tamanho da lista será de um. É iniciado um novo ciclo de recolha e introdução de dados.

É importante referir que os eventos de criação e remoção podem surgir a qualquer momento, influenciando ou não o processo de recolha que pode estar a ser realizado ou não. Abaixo estão as situações que podem surgir e como são processadas segundo o esquema da Figura 4.9.

- **Evento de criação:**
  - **Lista de PIDs = 1** → significa que está-se perante o único processo que acabou de ser criado. O ciclo de recolha e introdução de dados é iniciado;
  - **Lista de PIDs > 1** → o ciclo acima mencionado já se encontra em funcionamento.
- **Evento de remoção:**
  - **Lista de PIDs > 0** → o ciclo acima mencionado já se encontra em funcionamento;
  - **Lista de PIDs = 0** → visto que se trata de um evento de remoção e a lista já foi atualizada (ver Figura 4.9), está-se perante ao último processo que diz respeito a um nome de executável específico. O ciclo de recolha e introdução de dados é interrompido e *MoniAgs* ficará à espera de um novo evento de criação.

Para este sistema de monitorização foram criadas duas *dashboards* de monitorização, uma para cada tipo de agente (contínuo e momentâneo). Estas

serão apresentadas no próximo capítulo 5 bem como os resultados obtidos dos outros sistemas apresentados neste capítulo.

Capítulo 5

Apresentação e Discussão dos Resultados

Neste capítulo são apresentados e discutidos todos os resultados provenientes deste projeto. Os resultados estão divididos em quatro secções exibidas ao longo deste capítulo, as duas primeiras correspondem à vertente da monitorização, seguida de uma secção que diz respeito à deteção de erros dos agentes através da aplicação *MoniLogs* e a última à prevenção de falhas. Para efeitos de monitorização foi imprescindível desenvolver uma plataforma de monitorização (*website*) quer para controlar em tempo real os agentes e máquinas da AIDA, quer para analisar o historial de desempenho de cada componente. Na segunda secção é dada a resposta à **Questão 5** através de um estudo sobre a carga de trabalho das máquinas e agentes da AIDA. Na terceira secção são analisados os *e-mails* de alerta oriundos do sistema de monitorização e prevenção dos agentes baseado nos ficheiros *log* relativos aos erros de cada agnete. Finalmente, a resposta à questão **Questão 6** é esclarecida na última secção deste capítulo, onde são analisados os resultados oriundos dos sistemas de prevenção de falhas.

5.1 Plataforma de Monitorização

A plataforma de monitorização desenvolvida possui duas partes que serão exibidas nesta secção: as *dashboards* e a página de administração. Uma vez que esta plataforma é um *website*, torna a acessibilidade mais fácil por parte dos administradores. Este *website* encontra-se também localizado na máquina hsa-portint-tes e contém um menu inicial onde é possível aceder a todas as *dashboards* desenvolvidas e à página de administração. Neste mesmo menu, é informado ao utilizador se o servidor da ferramenta de BI do *Pentaho Community* se encontra ou não ativo na máquina hsa-aida19. Para além disto, uma simples caixa de pesquisa permite procurar em que máquina é executado determinado agente. Esta é uma ferramenta rápida e útil para os administradores, visto que a AIDA possui um elevado número de agentes.

5.1.1 *Dashboards* de Monitorização

Como já mencionado na subsecção 3.4.2, para o desenvolvimento destas *dashboards* foi utilizado a ferramenta de BI *Pentaho Community* juntamente com o módulo CDE. Para além das propriedades e componentes que esta ferramenta e módulo oferecem, foi necessário recorrer a alguns *scripts* de HTML e *JavaScript*.

Foram criadas quatro *dashboards*. A primeira é referente aos ficheiros *log* dos agentes monitorizados pela aplicação *MoniLogs*. A segunda e terceira são correspondentes à aplicação *MoniAgs* e foi necessário criar duas *dashboards* distintas devida às diferenças entre agentes contínuos e momentâneos. Por último, a quarta *dashboard* é referente às máquinas e à aplicação *MoniMaqs*.

Dashboard Agentes (*logs*)

A Figura 5.1 apresenta um excerto da *dashboard* referente aos *logs* dos agentes. No painel mais pequeno à esquerda é possível visualizar o primeiro filtro por máquina e ainda um gráfico circular a informar a quantidade de cada tipo de agentes ativos segundo a aplicação *MoniLogs* baseada nos ficheiros *log*. Neste mesmo painel ainda existe outro filtro referente aos agentes que

não está exibido na Figura 5.1. Este segundo filtro é dependente do primeiro, pois pretende selecionar os agentes individualmente que são executados em máquinas distintas.

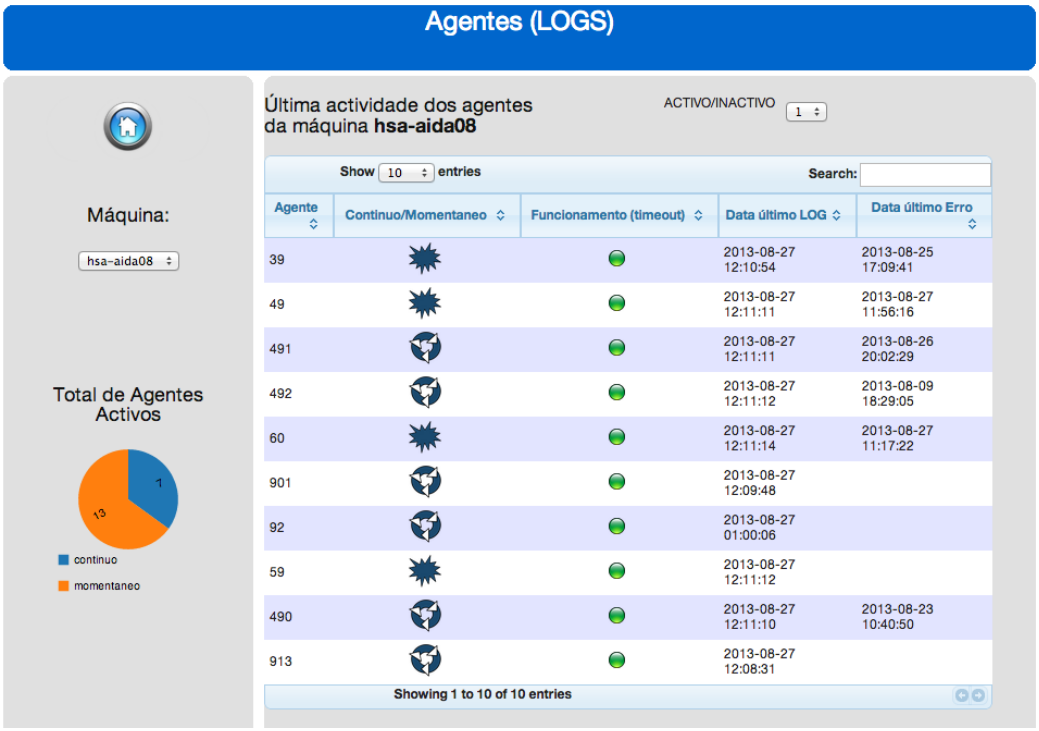

Figura 5.1: Excerto da *dashboard* Agentes (*logs*) retirado a 27-08-2013.

No maior painel à direita na Figura 5.1 encontra-se uma tabela com informações cruciais acerca dos registos de cada agente nos respetivos ficheiros *log*, quer os que registam a atividade, quer os que registam os erros. Nesta tabela ainda é possível saber se cada agente é do tipo momentâneo ou contínuo e se este atualmente encontra-se em *timeout* segundo a aplicação *MoniLogs*. Através do filtro "ativo/inativo" situado acima dessa mesma tabela, é possível visualizar os agentes que se encontram inativos e os seus últimos registos nos seus ficheiros *log* e o seu tipo.

Neste *dashboard* ainda é possível visualizar em tempo real o atual *score* de cada agente segundo a Tabela 3.1. Através do segundo filtro relativo aos agentes de cada máquina é possível visualizar um gráfico de linhas com os

percentis atuais (Tabela 3.1) e os últimos intervalos de cada agente. Utilizando o mesmo filtro juntamente com outro filtro que permite selecionar um período de tempo (data e hora de início e fim), o utilizador pode analisar o historial de um determinado agente selecionando o intervalo de tempo que lhe interessa investigar.

No apêndice A.1 encontram-se excertos desta *dashboard*.

Dashboard Agentes Cargas (Momentâneos)

Esta *dashboard* apresenta informações acerca dos indicadores de desempenho dos agentes momentâneos recolhidos pela aplicação *MoniAgs*. Tal como a *dashboard* acabada de apresentar, esta possui uma estrutura idêntica com dois painéis. No painel mais pequeno à esquerda, encontram-se os filtros de máquinas e agentes tal como a *dashboard* anterior, para além disto também possui legendas pormenorizadas dos indicadores de desempenho referentes aos agentes momentâneos (secção 3.1).

Quanto ao painel à direita são exibidos vários componentes:

- Gráfico de barras horizontal com os agentes que são executados numa determinada máquina selecionada no primeiro filtro. É mostrado a média do tempo de execução, CPU, memória e I/O de cada agente ao longo dos tempos, desde o registo mais antigo efetuado por *MoniAgs* na base de dados situada na máquina hsa-aida19. Ao clicar nas legendas deste gráfico pode-se filtrar a informação, sendo possível uma melhor visualização de alguns indicadores;

- Quatro gráficos de barras, cada um correspondente a um indicador de desempenho. Em cada gráfico encontram-se os 5 agentes que mais utilizam o respetivo indicador de desempenho. Estes gráficos não utilizam nenhum filtro desta *dashboard* e permite aos administradores da AIDA identificar quais os agentes que consomem mais recursos. Desta forma, os administradores sabem em quais agentes deve-se tomar medidas para obter uma melhor utilização dos recursos;

- Gráfico de barras que exibe a média de cada indicador de desempenho

de um determinado agente de uma determinada máquina (seleção efetuada pelos dois filtro do painel à esquerda) durante um determinado período de tempo (filtro de data e hora inicial e final). Para melhor visualização de cada indicador individualmente, basta clicar nas legendas do gráfico. Através deste gráfico, os administradores estão habilitados a analisar o historial do comportamento de cada agente momentâneo perante cada um dos seus indicadores desempenho, mais uma vez, uma melhor gestão dos recursos pode ser efetivada a partir destes dados.

No apêndice A.2 encontram-se excertos desta *dashboard*.

Dashboard Agentes Cargas (Contínuos)

Esta *dashboard* é semelhante à anterior pois também está relacionada com a aplicação *MoniAgs*. Exibe indicadores de desempenho dos agentes contínuos da AIDA. Uma vez que estes agentes possuem os ficheiros executáveis com o mesmo nome, consoante as suas funções (ver Tabela 4.1), os indicadores de desempenho são agrupados pelo nome do executável e não por agentes. Deste modo, a análise dos indicadores é feita a um grupo e não a cada agente individualmente.

No painel à esquerda, tal como a *dashboard* anterior, aparece uma legenda pormenorizada dos indicadores de desempenho e o filtro para as máquinas. Para além disto, existe um filtro dependente ao das máquinas com os nomes dos ficheiros executáveis. Uma pequena tabela imediatamente a baixo, exibe os agentes responsáveis pelo executável que está selecionado neste último filtro, que por sua vez é executado na máquina que está selecionada no primeiro filtro.

No painel maior que se encontra à direita, existe os seguintes componentes:

- Gráfico de barras que mostra a média da quantidade de ficheiros executáveis, CPU, memória e I/O de cada executável que é processado na máquina selecionada no primeiro filtro. Estas médias são calculadas com todos os valores que estão registados na base de dados na má-

quina hsa-aida19. Para melhor visualização, deve-se filtrar informação nas respetivas legendas;

- Quatro gráficos de barras, cada um correspondente a um indicador de desempenho. Em cada gráfico encontram-se os 5 agentes que mais utilizam o respetivo indicador de desempenho. Estes gráficos não utilizam nenhum filtro desta *dashboard* e permite aos administradores da AIDA identificar quais os agentes e respetivos executáveis que consomem mais recursos. Através de um balanceamento correto, uma melhor utilização dos recursos pode ser concretizada;

- Gráfico de linhas com os 4 indicadores de desempenho de um determinado executável de uma determinada máquina durante um determinado período de tempo (data e hora inicial e final). É possível identificar os momentos de picos de cada indicador individualmente e estudar o historial do comportamento de cada executável. Filtragem pelas legendas para melhor visualização é recomendado.

No apêndice A.3 encontram-se excertos desta *dashboard*.

Dashboard Máquinas Cargas

Esta *dashboard* funciona como uma plataforma de monitorização das máquinas da AIDA utilizadas neste projeto. Seguindo a mesma estrutura dos *dashboards* anteriores, no painel à esquerda existe um filtro para selecionar a máquina que se deseja analisar. Este filtro só influencia a partir da terceira secção de componentes desta *dashboard*.

À direita, temos os seguintes componentes no painel:

- Gráfico de barras horizontal com o *score* total atual de cada máquina (ver Tabela 3.2);

- Três gráficos circulares com a percentagem livre/ocupada atual, segundo *MoniMaqs*, de cada indicador de desempenho da máquina selecionada no filtro do painel à esquerda;

- Quatro gráficos de barras: percentagem de CPU, memória e disco livre e o número de agentes executados em cada máquina segundo a aplicação *MoniLogs* baseada nos ficheiros *log*. A partir destes gráficos os administradores facilmente detetam quais são as máquinas que consomem mais recursos, e desta forma podem proceder a um melhor balanceamento dos recursos. Por exemplo, selecionando a máquina apropriada aquando criam um novo agente. É importante referir que para esta análise, a informação exibida nos gráficos relativos aos indicadores de desempenho é filtrada por uma data inicial e final escolhida pelo utilizador. Por defeito é exibido a utilização do dia atual;

- Três gráficos de linhas, um para cada indicador, com os respetivos limites da Tabela 3.2 e os valores durante o período de tempo (data inicial e final) selecionado na segunda secção de gráficos desta *dashboard*. Isto habilita os administradores analisar o historial de cada máquina e verificar períodos críticos. Após identificar estes períodos, o utilizador pode ir procurar nas outras *dashboards* já apresentadas o comportamento dos agentes dessa máquina durante esse período. Desta forma, o utilizador saberá qual foi o(s) agente(s) que causaram tal situação crítica e poderá assim tomar medidas preventivas e reformular as tarefas de determinado(s) agente(s).

No apêndice A.4 encontram-se excertos desta *dashboard*.

5.1.2 Página de Administração

A principal função desta página é monitorizar as três aplicações desenvolvidas e apresentadas no Capítulo 4. No topo desta página encontra-se uma tabela a informar se as ditas aplicações encontram-se ou não a serem executadas na máquina hsa-portint-tes e as respetivas datas mais recentes registadas na base de dados. Assim, facilmente deteta-se se os sistemas de monitorização e prevenção estão ativos ou inativos, e neste caso, há quanto tempo.

Abaixo da tabela que traduz o estado de cada uma das aplicações, encontra-se um painel dividido em três separadores. Cada um destes diz respeito a uma das aplicações desenvolvidas.

Separador *MoniLogs*

Neste separador é exibido uma tabela com informações de cada agente no contexto da aplicação *MoniLogs*. Após selecionar a máquina num filtro junto à tabela, é exibido as seguintes informações: nome do agente, quantidade de registos que este agente tem na base de dados, a data do registo mais antigo (para saber se já regista há 5 dias, ver Figura 4.4), sinal verde ou vermelho caso o agente encontra-se sob vigia de *MoniLogs* ou não, respetivamente. Desta forma, o administrador facilmente sabe quais são os agentes que se encontram monitorizados por esta aplicação e quais não estão.

Este separador possui ainda dois botões. O primeiro apaga todos os registos efetuados pela aplicação *MoniLogs* na base de dados. Esta funcionalidade é útil quando o administrador dos sistemas deseja reiniciar todo o sistema de monitorização e prevenção de agentes baseado em *logs*. O segundo botão permite parar/iniciar a aplicação *MoniLogs*.

Separador *MoniMaqs*

Este separador possui funcionalidades muito importantes para o sistema de monitorização e prevenção de falhas das máquinas. Tal como mostra a Figura 5.2, neste separador é mostrado uma tabela com o estado atual de cada máquina e todos os seus limites atuais referentes a cada um dos indicadores de desempenho. Na Figura 5.2 ainda é possível visualizar dois botões em que ambos lançam uma janela sobre a página de administração para eliminar ou adicionar máquinas que se encontram sob vigia da aplicação *MoniMaqs* ou então editar os limites de cada máquina.

Tal como já mencionado na subsecção 3.3.2 e no final da secção 4.3, é extremamente importante os administradores terem a capacidade de modificar os limites do modelo apresentado na Tabela 3.2 aplicado no sistema de monitorização de prevenção de falhas das máquinas. Esta funcionalidade

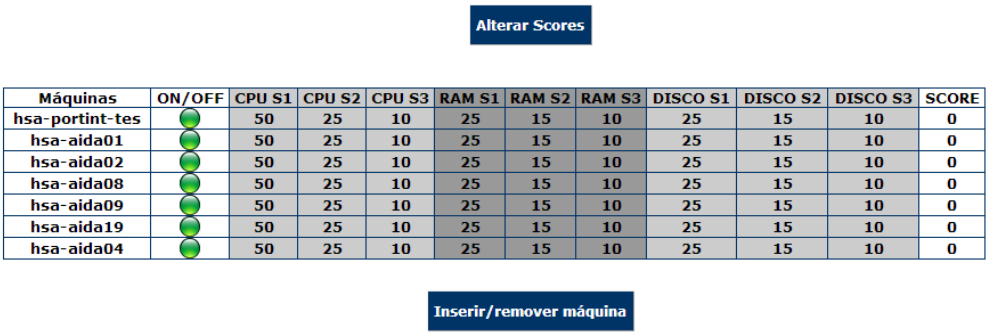

Figura 5.2: Excerto da página de administração (separador *MoniMaqs*), tabela com informações acerca do estado atual de cada máquina e seus respetivos limites atuais para efeitos do cálculo do *score* total baseado na Tabela 3.2.

encontra-se na Figura 5.3 onde o utilizador facilmente pode editar cada limite individualmente na respetiva caixa de texto.

Separador *MoniAgs*

Uma vez que a aplicação *MoniAgs* possui múltiplos sistemas de notificação, mais concretamente um para cada agente, esta aplicação necessita de ser vigiada para não causar danos na máquina hsa-portint-tes onde é executada. Para tal, além de exibir neste separador da página de administração a quantidade de CPU e memória que *MoniAgs* está no momento a consumir de hsa-portint-tes, foi criado um pequeno programa que vigia esta aplicação. Este programa é lançado periodicamente por uma tarefa do sistema operativo *Microsoft Windows* que foi criada. Neste programa é estabelecido limites para o consumo de CPU e memória, se estes limites são ultrapassados por *MoniAgs*, o administrador do sistema recebe um *e-mail* com informações sobre o sucedido e a aplicação é terminada.

Esta plataforma de monitorização foi o resultado mais imediato de todo este projeto. Nas próximas secções serão analisados resultados de casos quer retirados desta plataforma, quer obtido através dos *e-mails* de alerta enviados

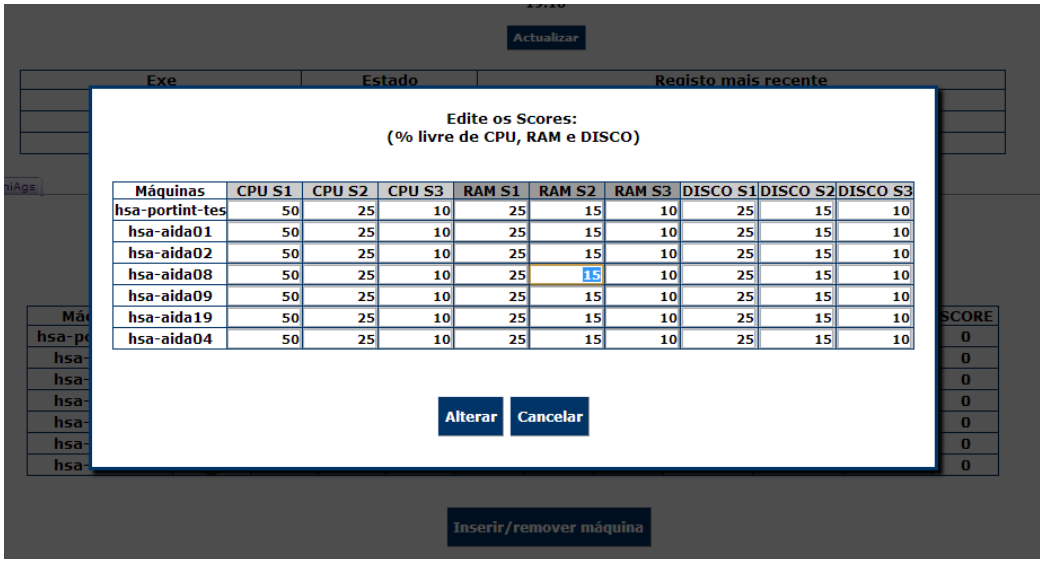

Figura 5.3: Janela lançada pelo botão "Alterar Scores" da página de administração (separador *MoniMaqs*).

pelos sistemas de monitorização de prevenção de falhas desenvolvidos.

5.2 Carga de Trabalho das Máquinas e Agentes (Questão 5)

Esta secção pretende responder à **Questão 5** colocada nos objetivos desta dissertação (secção 1.2). Serão apresentados e interpretados os dados recolhidos na AIDA do CHP ao longo do desenvolvimento deste projeto. No final da secção serão apresentadas novas interpretações de resultados que só são possíveis serem discutidas após a apresentação de resultados quer das máquinas, quer dos agentes, uma vez que as cargas de trabalho destes dois tipos de componentes estão estreitamente ligadas.

5.2.1 Máquinas

Em relação às máquinas, foi possível estudar as respetivas cargas de trabalho recolhendo e analisando dados relativamente aos indicadores de desempenho apresentados na secção 3.1. A Figura 5.4 permite visualizar a média de consumo de cada indicador de desempenho referente às máquinas e ainda a quantidade de agentes a serem executados em cada máquina no momento atual, segundo o sistema baseado nos *logs* de registo dos agentes. O excerto da *dashboard* apresentado na Figura 5.4 analisa os resultados entre o período de 27 de julho a 27 de agosto de 2013 para os 3 indicadores de desempenho.

Em relação ao consumo de CPU, o primeiro gráfico da Figura 5.4 apresenta a percentagem média de CPU livre durante o período da amostra selecionado. Pode-se retirar várias conclusões deste gráfico:

- A máquina hsa-aida08 foi a que consumiu mais CPU com uma média de 14,17%;
- Por outro lado, das máquinas que executam agentes da AIDA, é a hsa-aida01 que consumiu menos CPU com uma média de 5,09%;
- A máquina que consumiu menos CPU foi a hsa-portint-tes, média de 1,79%.

Quanto ao consumo de memória, a partir do segundo gráfico da Figura 5.4 pode-se retirar:

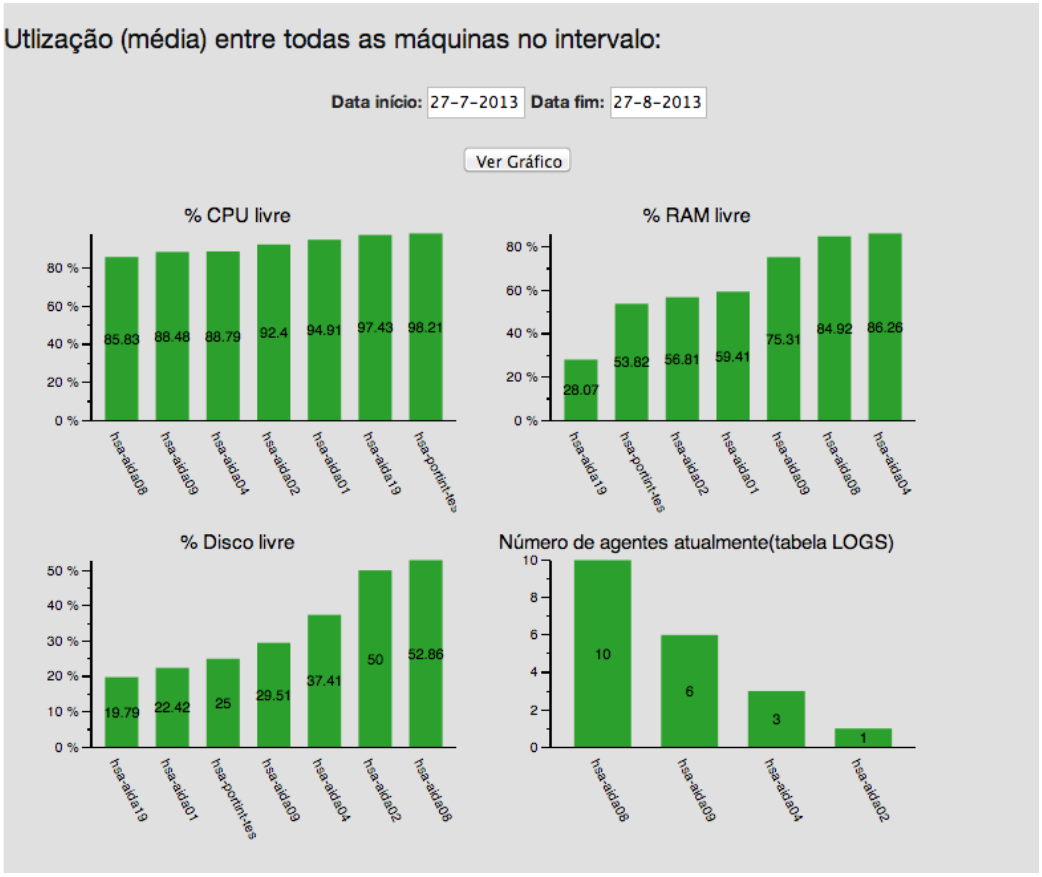

Figura 5.4: Excerto do *dashboard* referente às máquinas da AIDA do CHP. Retirado a 27-8-2013.

- A máquina hsa-aida19 é a que consumiu mais memória neste período de tempo, com uma média de 71,93%;
- Das máquinas que executam agentes da AIDA, foi a hsa-aida02 e hsa-aida01 que consumiram mais memória, médias de 43,19% e 40,59% respetivamente;
- As máquina que consumiram menos memória foram a hsa-aida08 e hsa-aida04, médias de 15,08% e 13,74% respetivamente.

Estes resultados obtidos referentes ao consumo de CPU e memória serão interpretados na subsecção 5.2.3, após apresentação dos resultados dos

agentes.

Em relação à percentagem de disco livre (terceiro gráfico da Figura 5.4), concluiu-se que a hsa-aida19 e hsa-aida01 são as que tem menos espaço disponível e a hsa-aida02 e hsa-aida08 são as que tem mais. Pode-se concluir que o espaço disponível de disco de todas estas máquinas ainda não comprometem o bom funcionamento destas. Apesar das máquinas hsa-aida19 e hsa-aida01 apresentarem baixos valores, ainda não chegaram ao limite mínimo estabelecido na tabela da Figura 5.2 (15%). Claro que se os administradores detetarem comportamentos inapropriados destas máquinas, uma das medidas a ser tomada será a "limpeza" de ficheiros destas máquinas e também um ligeiro aumento do limite mínimo da utilização dos discos destas máquinas na tabela da Figura 5.2.

A partir do último gráfico da Figura 5.4 chega-se à conclusão que a máquina hsa-aida08 é a que executa mais agentes segundo o sistema de monitorização baseado nos *logs* de registo dos agentes. A razão pela qual neste gráfico não se visualizar nenhum agente na máquina hsa-aida01 é devido ao facto dos agentes que são executados nesta máquina não escreverem nos seus ficheiros *log* regularmente. Ao rever a Tabela 4.1, sabe-se que na hsa-aida01 apenas são executados agentes contínuos que por sua vez não são executados regularmente e consequentemente não registam nos seus ficheiros *log* as suas atividades. No momento em que a Figura 5.4 foi retirada, no dia 27 de agosto de 2013, ainda não havia qualquer ficheiro *log* dos agentes da máquina hsa-aida01 modificado nesse dia. Algo semelhante acontece com a máquina hsa-aida02 que apenas apresenta um agente nesse mesmo gráfico.

CPU

A Figura 5.5 exibe excertos da *dashboard* das máquinas onde é possível ver o consumo de CPU das máquinas hsa-aida04 e 09 durante o período de 17 de julho a 13 de agosto de 2013 e da hsa-aida02 no dia 30 de julho de 2013. No primeiro gráfico da Figura 5.5 relativamente à máquina hsa-aida04 é possível constatar que consumo de CPU é constante com variações entre os 5-10%. Durante este período sucedeu-se duas situações em que o consumo

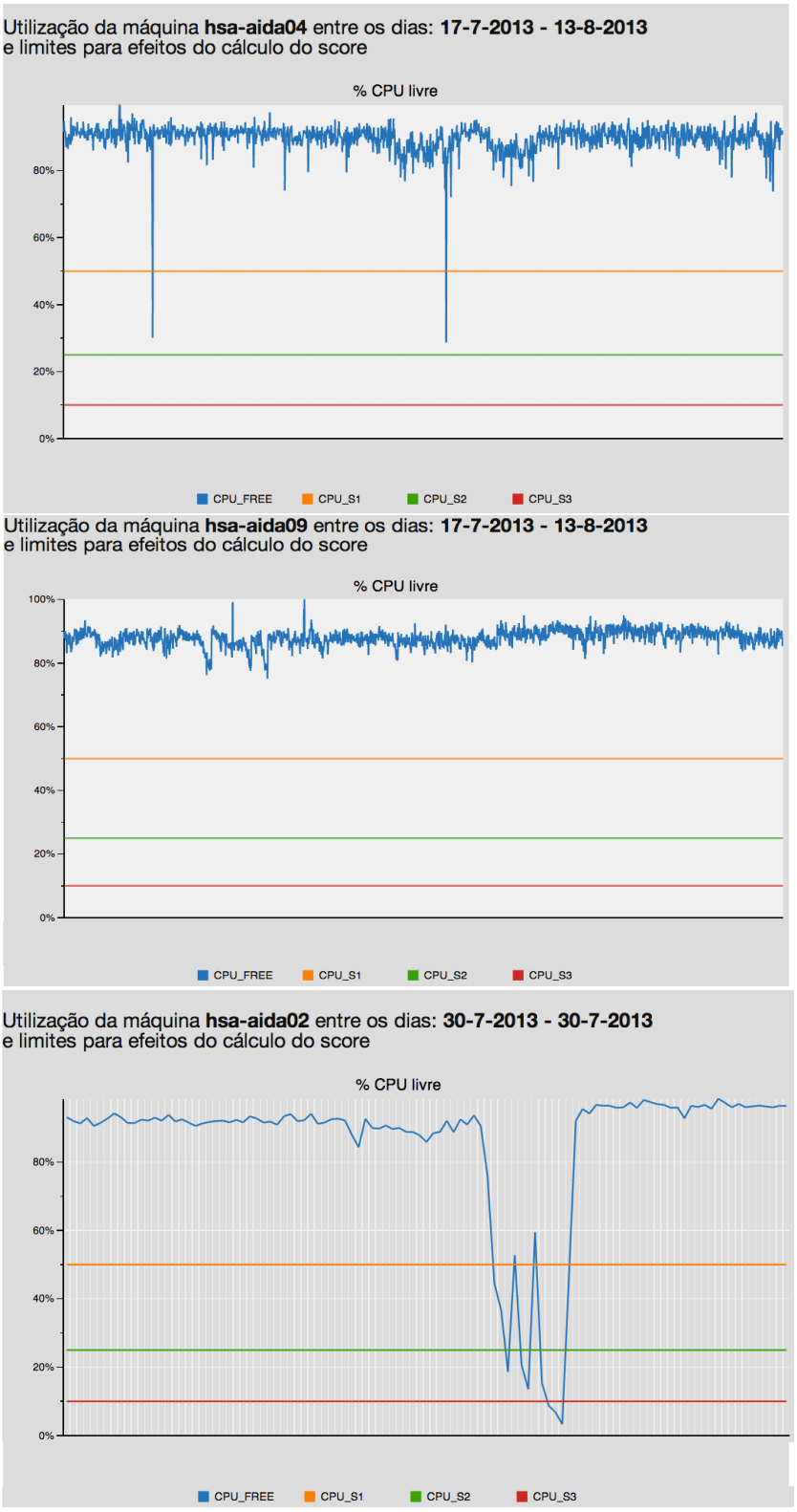

Figura 5.5: Excertos do *dashboard* referente ao consumo de CPU das máquinas da AIDA do CHP. Retirado a 13-8-2013.

de CPU ultrapassou o primeiro limite (50%).

No segundo gráfico da Figura 5.5 nota-se que a máquina hsa-aida09 tem uma variação de consumo de CPU ainda menos que hsa-aida04. As máquinas hsa-aida01 e 08 durante este período de tempo apresentaram um comportamento muito semelhante ao da hsa-aida09 no que toca ao consumo de CPU.

Finalmente, no terceiro gráfico da Figura 5.5, é exibido a percentagem de CPU livre da máquina hsa-aida02 num dia em que houve problemas nesta máquina. O gráfico mostra um excessivo consumo de CPU na parte da tarde com variações bastante acentuadas e com momentos onde a percentagem de processador livre excede os limites de 50%, 25% e 10%, causando uma contribuição para o cálculo do *score* total desta máquina de 1, 2 e 3 valores respetivamente. Esta situação foi detetada pelo sistema de monitorização e prevenção de falhas das máquinas (*MoniMaqs*) e *e-mails* de alerta foram enviados. Esta situação será explorada na próxima secção 5.4 alusiva à prevenção de falhas.

Quanto às máquinas monitorizadas que não executam os agentes intervenientes neste projeto, nomeadamente a hsa-aida19 e a hsa-portint-tes, verificou-se, durante o mesmo período, um nível de consumo muito baixo e bastante constante a rondar os 2% apenas com picos pontuais de de 10% e 40%, respetivamente. Este facto também pode ser constatado pelos valores exibidos na Figura 5.4. A razão para este comportamento baseia-se no facto destas máquinas não executarem agentes da AIDA que são constantemente executados e que realizam tarefas que consomem uma considerável quantia de CPU.

Memória

Relativamente ao consumo de memória nas máquinas da AIDA que executam os agentes, constatou-se que durante este período, 17 de julho a 13 de agosto de 2013, o consumo de memória, em geral, foi extremamente constante. No primeiro gráfico da Figura 5.6 encontra-se o consumo de memória da hsa-aida08 durante este período. A máquina hsa-aida04 teve um comportamento semelhante: a rondar o respetivo valor do gráfico referente à

utilização da memória na Figura 5.4. Observa-se que o consumo de memória da hsa-aida08 foi extremamente constante, com variações mínimas e o consumo de memória rondou sempre os 5-10%.

Quanto ao segundo gráfico da Figura 5.6, observa-se que o consumo de memória na máquina hsa-aida09 rondou os 40%, até que em dado momento, num dia nos inícios de agosto o consumo de memória desceu acentuadamente. O que aconteceu neste momento foi, provavelmente, um reinicio da máquina que consequentemente libertou bastante memória proveniente de algum processo que deveria estar a consumir uma quantidade razoável. É de se salientar que após este reinicio, o gráfico mostra que o consumo de memória aumenta gradualmente ao longo do tempo de forma muito lenta. Este facto requer alguma investigação sob os processos que são executados nesta máquina para descobrir qual o processo que causa este comportamento e identificar se trata-se ou não de uma situação de risco para o bom funcionamento da hsa-aida09. O comportamento do consumo de memória nas máquinas hsa-aida01 e 02 são semelhantes ao da máquina hsa-aida09 apresentado na Figura 5.6. No entanto foi possível apurar que a linha de consumo de memória na hsa-aida02 apesar de ter um declive muito próximo ao da hsa-aida09, possui variações mais bruscas. Quanto à hsa-aida01, as variações são semelhantes ao da hsa-aida09, o declive negativo é mais acentuado e também foi possível visualizar que os picos acentuados de libertação de memória foram mais frequentes.

Tal como mostrado para o CPU na Figura 5.5, no terceiro gráfico da Figura 5.6 encontra-se o consumo de memória da máquina hsa-aida02 no dia 30 de julho. Constata-se que o excessivo consumo de CPU na parte da tarde foi acompanhado por um excessivo consumo de memória. Neste gráfico encontra-se grandes variações de consumo de memória, ultrapassando os limites de 25%, 15% e até 10%. Este facto contribui ainda mais para o aumento do *score* da máquina nestes momentos.

As máquinas hsa-aida19 e hsa-portint-tes neste mesmo período de amostra, revelaram um consumo alto e constante de memória: 65% e 45% respetivamente. Facto que também pode ser verificado pela média de consumo no segundo gráfico da Figura 5.4. Para além disto, a máquina hsa-aida19 registou algumas situações graves a ultrapassar os limites estabelecidos na Tabela

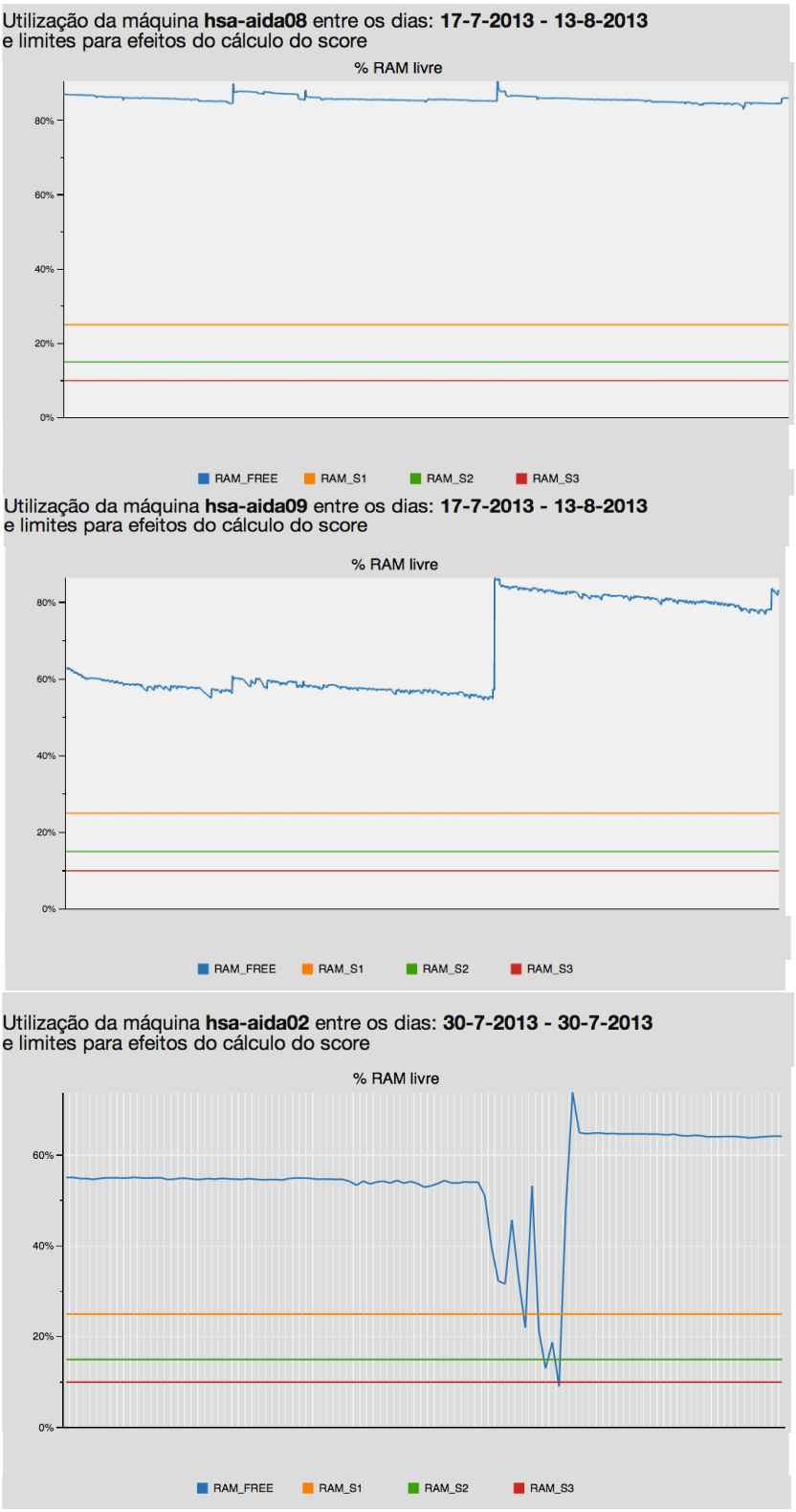

Figura 5.6: Excertos do *dashboard* referente ao consumo de memória das máquinas da AIDA do CHP. Retirado a 13-8-2013.

5.2. A razão principal para haver tanto consumo de memória na hsa-aida19 é devido à base de dados que lá se encontra implementada, acrescido do facto de estar permanentemente ligado o servidor da ferramenta *Pentaho Community*. Quanto à hsa-portint-tes, apesar das aplicações *MoniLogs, MoniMaqs e MoniAgs* serem executadas nessa máquina e não consomirem demasiada memória, contribuem para um aumento do consumo de memória juntamente com outras aplicações e serviços que se encontram na hsa-portint-tes.

Espaço livre no disco

Quanto ao espaço livre nos discos das máquinas analisadas da AIDA, constatou-se que este é bastante constante em todas as máquinas envolvidas neste projeto. Os valores que rondam são semelhantes aos apresentados no gráfico referente ao espaço livre de disco da Figura 5.4. Este facto mostra que existe bons mecanismos implementados nesta plataforma que visam a transferência de arquivos das máquinas para unidades próprias de armazenamento.

A função deste gráficos de linhas serve para observar detalhadamente o comportamento das máquinas individualmente e também detetar situações graves que tenham ocorrido. Embora não apareça nos gráficos o eixo das abcissas, para o utilizador saber qual é a data e hora de determinado ponto de qualquer um destes gráficos, basta colocar o ponteiro do rato sobre o ponto que uma pequena caixa de texto exibirá a informação pretendida.

5.2.2 Agentes

Relativamente aos agentes da AIDA monitorizados neste projeto, é fundamental, mais uma vez, separar os agentes momentâneos e contínuos para compreender melhor a carga de trabalho de cada agente. Sendo assim, nesta subsecção será apresentado primeiramente a carga de trabalho dos agentes momentâneos e posteriormente a dos contínuos.

Agentes Momentâneos

Na Figura 5.7, é possível visualizar os 5 agentes momentâneos que, em média, demoram mais tempo a serem executados, os 5 que consomem mais CPU, os 5 que consomem mais memória e finalmente, os 5 com mais atividade I/O.

Figura 5.7: Excerto do *dashboard* referente aos agentes momentâneos da AIDA do CHP. Retirado a 13-8-2013.

Quanto ao primeiro gráfico da Figura 5.7 referente ao tempo de execução de cada agente é possível apurar que, indiscutivelmente, o agente 609 situado na máquina hsa-aida04 é o que demora mais tempo a realizar as suas tarefas. Por norma, os agentes momentâneos tem um período de execução bastante curto a rondar os 10-20 segundos.

Relativamente ao consumo de CPU, o agente 36 localizado na hsa-aida04 é o que revela um maior nível de consumo (13,42% em média). Após este agente, o segundo gráfico da Figura 5.7 mostra os agentes da máquina hsa-

aida08 e 09.

O terceiro gráfico da figura em questão, demonstra que os 5 agentes momentâneos que utilizam mais memória possuem níveis muito semelhantes de consumo. Desta forma, torna-se difícil destacar qual o agente que consome mais memória, apesar de o agente 39 da hsa-aida08 estar à frente.

Quanto à atividade de I/O, é possível constatar que o agente com mais atividade é também o 39 da hsa-aida08 (219,27 *KBytes* por segundo), seguido dos agentes das máquinas hsa-aida04 e 09.

É de se salientar que o agente 39 da hsa-aida08 é o único que consta nestes quatro gráficos apresentados. Sendo o que consome mais memória e com maior atividade I/O, é o segundo que consome mais CPU e o terceiro com maior tempo de execução.

As Figuras 5.8 e 5.9 mostram alguns dos gráficos que são possíveis visualizar na *dashboard* referente à carga de trabalho dos agentes momentâneos. Optou-se por mostrar as cargas dos agentes momentâneos executados na hsa-aida04 visto que é nesta máquina que se encontra os agentes 609 e 36 que estão bem evidenciados na Figura 5.7. Através do gráfico e da respetiva ampliação na Figura 5.8, é possível apurar que o agente 609 apesar de ser o agente que demora mais tempo a ser executado, possui um consumo de CPU e atividade I/O praticamente nulos. Este baixo consumo destes dois indicadores também é possível verificar no agente 35. Desta forma, é o agente 36 o único desta máquina que consome um nível considerável de CPU. Este mesmo agente é o único da hsa-aida04 que apresenta uma elevada atividade I/O.

É importante lembrar que nesta máquina (hsa-aida04) apenas são executados estes três agentes momentâneos. Este facto deve ser levado em conta para o balanceamento de recursos futuros.

Relativamente ao gráfico apresentado na Figura 5.9 é possível visualizar a média dos indicadores de desempenho do agente 39 (hsa-aida08) no dia 2 de agosto de 2013. As médias elevadas de cada indicador para este dia confirmam a média geral já apresentado na Figura 5.7. Conclui-se que este comportamento é constante e que a média geral apresentada representa de forma correta a carga de trabalho do dia a dia do agente 39.

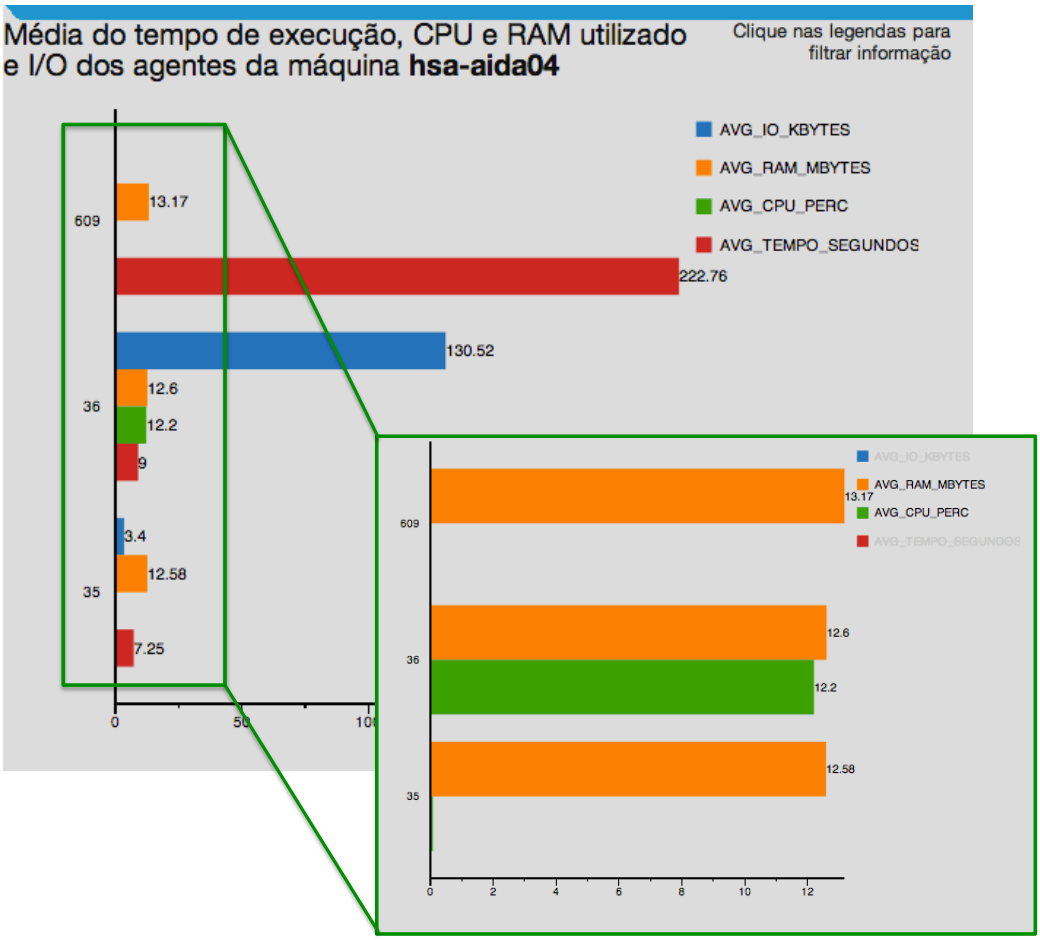

Figura 5.8: Excerto do *dashboard* referente à carga de trabalho dos agentes executados na máquina hsa-aida04 da AIDA do CHP. Retirado a 27-8-2013.

No que toca ao quinto indicador de desempenho dos agentes momentâneos, a frequência de atividade que por sua vez é o indicador único para a prevenção de falhas dos agentes, foi possível retirar alguns casos mostrados nas Figuras 5.10 e 5.11.

Na Figura 5.10 é apresentado os percentis atuais no momento em que a imagem foi criada e os últimos intervalos de registo da atividade do agente 39 no seu respetivo ficheiro *log*. É observado um comportamento repetitivo onde a aplicação *MoniLogs* regista tanto os intervalos superiores ao percentil 85 e 90, como regista intervalos significativamente inferiores. Como já men-

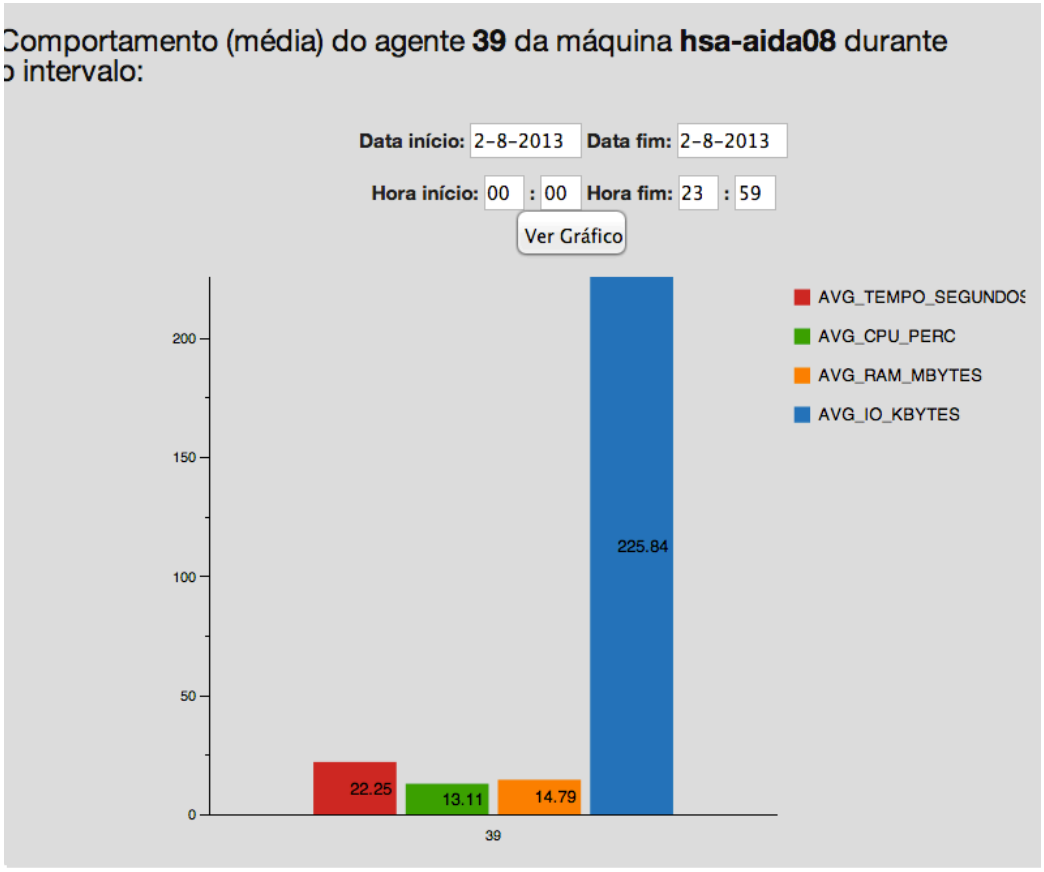

Figura 5.9: Excerto do *dashboard* referente à carga de trabalho do agente 39 executado na hsa-aida08 da AIDA do CHP. Retirado a 27-8-2013.

cionado no Capítulo 4 (secção 4.2), este gráfico não representa na totalidade a realidade e os valores apresentados estão de acordo com o mencionado na dita secção. O agente 39 regista frequentemente no seu ficheiro *log* em intervalos bem mais curtos que 5 ou 10 minutos tal como sugere a Figura 5.10. No entanto este facto não influencia a deteção de situações críticas. Quando estes agentes se encontram em situações críticas as barreiras dos percentis são quebradas e o sistema de prevenção de falhas deteta e informa a situação. Na leitura destes gráficos deve-se levar em conta os picos que ultrapassam o comportamento normal dos agentes e não os intervalos que estão abaixo desses limites que representam, acima de tudo, situações normais.

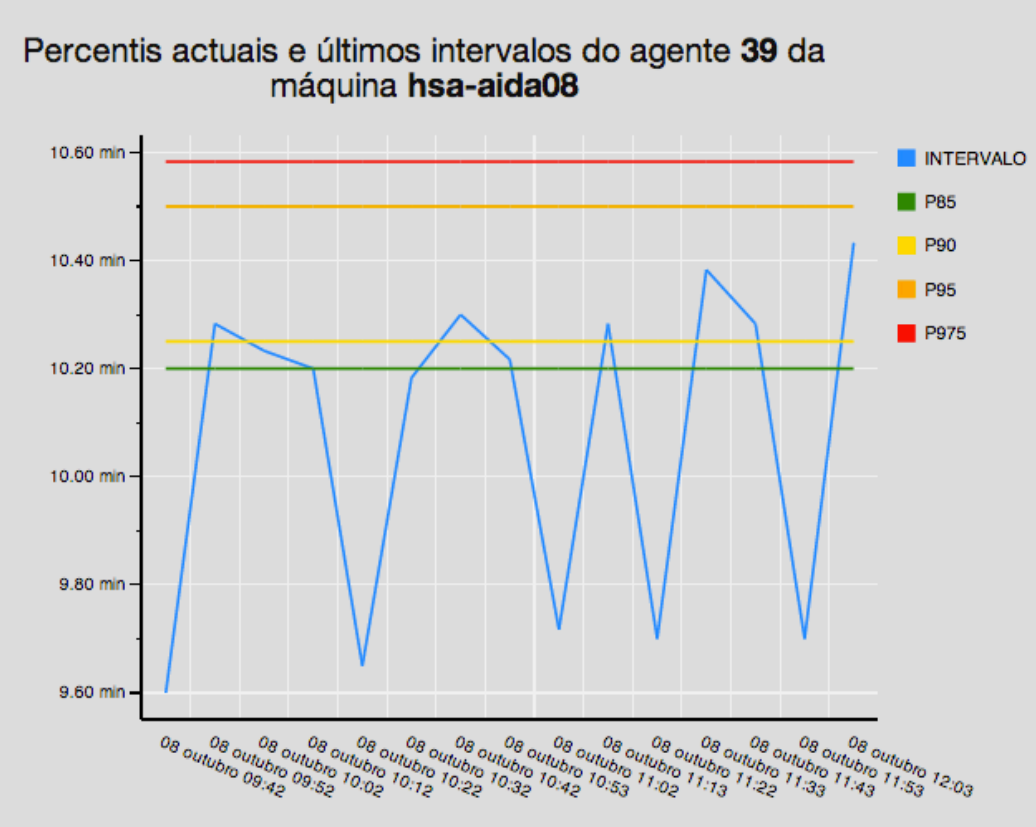

Figura 5.10: Excerto do *dashboard* referente ao *log* do agente 39 da AIDA do CHP. Retirado a 8-10-2013.

A Figura 5.11 representa exatamente o que acabou de ser dito. Nesta figura é possível visualizar os picos que romperam os limites dos percentis referentes à frequência de atividade do agente 609 no dia 5 de setembro de 2013 à tarde. Como mostra o gráfico, às 16:15 horas registou-se uma situação grave. Este momento foi detetado quer pelo sistema de prevenção de falhas dos agentes, quer das máquinas. Na próxima secção 5.4 esta situação será evidenciada.

Agentes Contínuos

Como já referido anteriormente, no que toca à carga de trabalho dos agentes contínuos estes são tratados como um grupo. Cada grupo é formado

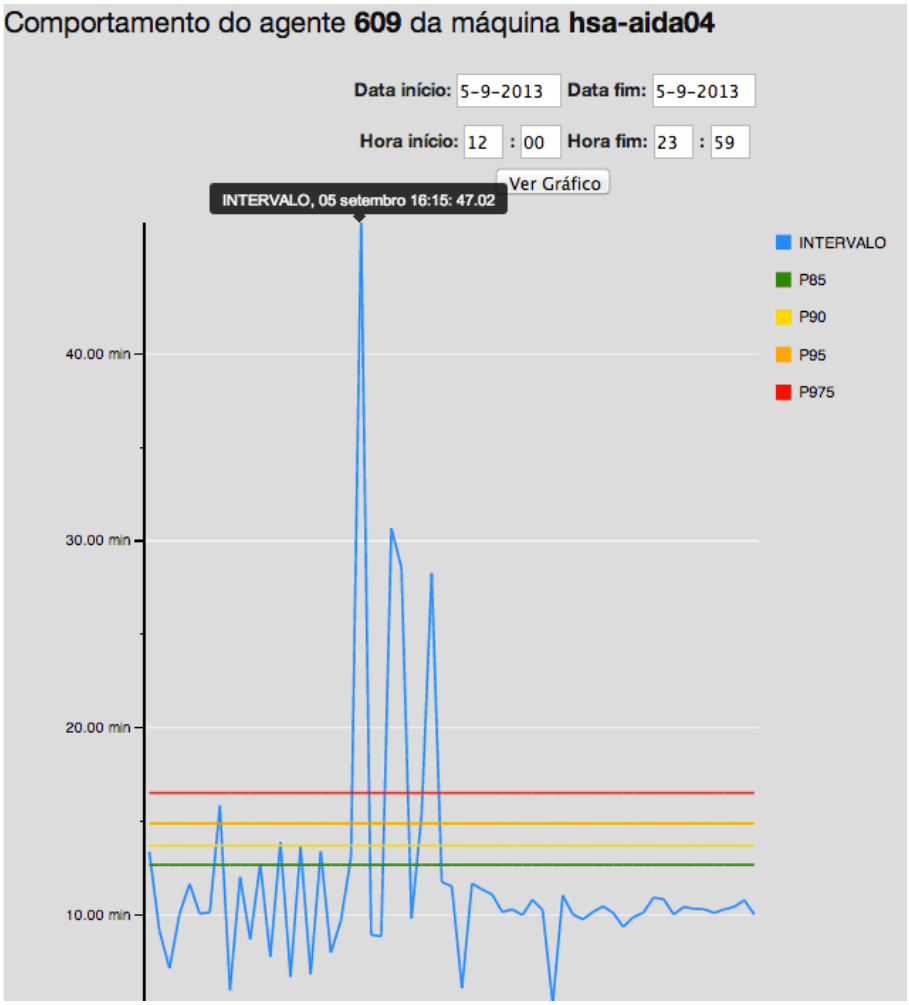

Figura 5.11: Excerto do *dashboard* referente aos *logs* do agente 609 da AIDA do CHP no dia 5 de setembro de 2013. Retirado a 8-10-2013.

por dois critérios: primeiramente, os agentes têm de ser executados na mesma máquina e em segundo lugar, os agentes têm o nome dos ficheiros executados gerados iguais. Deste modo, é apresentado na Figura 5.12 os 5 grupos de executáveis que, em média, geram mais processos (cada processo corresponde a um executável), os 5 que que consomem mais CPU, os 5 que consomem mais memória e os 5 com maior atividade I/O. Estes gráficos representam uma boa forma de os administradores analisarem quais agentes gastam mais recursos e assim definir prioridades no que toca à remoção/modificação das tarefas de cada agente. O mesmo acontece para os agentes momentâneos através da Figura 5.7.

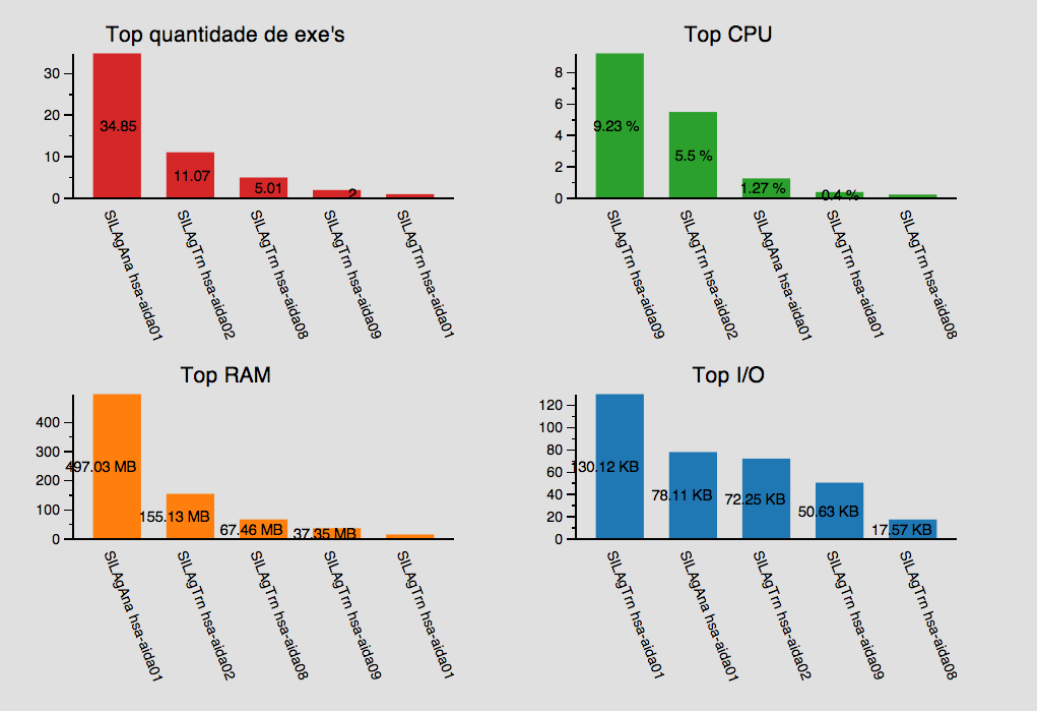

Figura 5.12: Excerto do *dashboard* referente aos agentes contínuos da AIDA do CHP. Retirado a 13-8-2013.

Ao analisar o primeiro gráfico da Figura 5.12, constata-se que o executáveis SILAgAna situados na hsa-aida01 são os que existem em maior número em média (aproximadamente 35).

Os executáveis SILAgTrn da hsa-aida09 e 02 são os que em média conso-

mem mais CPU.

A partir do terceiro gráfico da Figura 5.12 é possível apurar que os executáveis SILAgAna da hsa-aida01 são os que consomem mais memória, com uma diferença de quase 350 *MBytes* para os que se encontram em segundo lugar na lista: SILAgTrn da hsa-aida02. Este facto é esperado devido à grande discrepância do número de executáveis que se pode ver no primeiro gráfico da Figura 5.12. Mais executáveis, ou seja, mais processos, naturalmente consomem mais memória.

Finalmente, foi possível concluir que os executáveis SILAgTrn da hsa-aida01 são os que apresentam maior atividade I/O (130,12 *KBytes* por segundo).

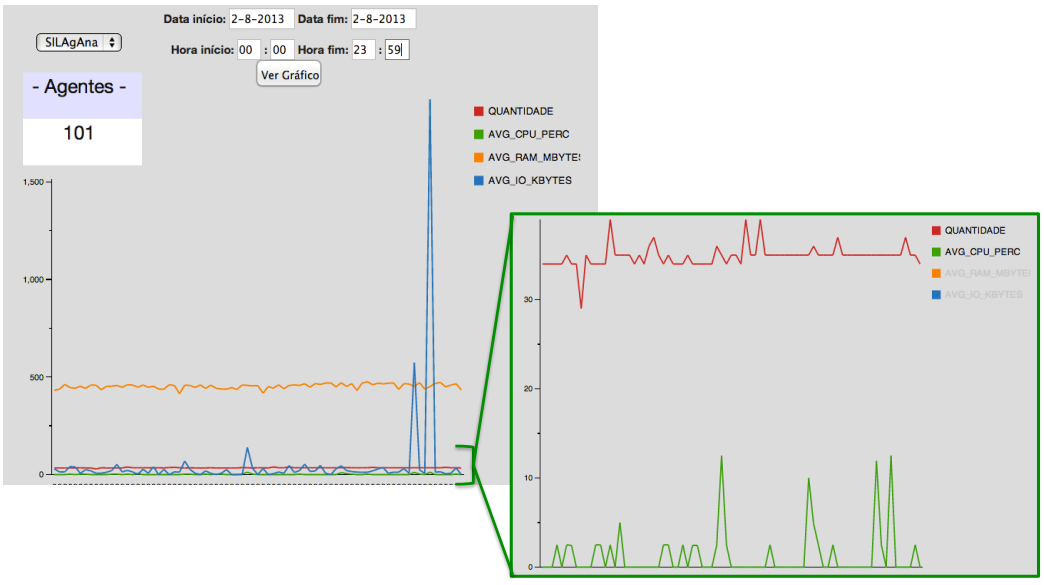

Figura 5.13: Excerto do *dashboard* referente ao agentes contínuo 101 da máquina hsa-aida01 da AIDA do CHP no dia 2 de agosto de 2013. Retirado a 13-8-2013.

Analisando alguns casos particulares, encontra-se na Figura 5.13 o comportamento dos executáveis SILAgAna que, como sugere a figura no canto superior esquerdo, apenas são gerados pelo agente 101 durante o dia 2 de agosto de 2013. Através do gráfico da esquerda da Figura 5.13 é possível apurar que o consumo de memória deste agente é elevado e constante, ron-

dando os 400-450 *MBytes*. Tal como já constatado, no gráfico da direita é possível confirmar que durante este dia, o número de executáveis foram cerca de 35. O consumo de CPU de todos estes executáveis mal excede os 10% e que a atividade I/O foi bastante constante durante o dia com a excepção de um pico elevadíssimo no final do dia. Facto que talvez se sucedeu devido à execução de alguma tarefa diária que este agente tem programado para o final de cada dia.

Relativamente à frequência de atividade dos agentes contínuos foi possível concluir que a maioria destes agentes devem ser excluídos/ignorados do sistema de prevenção de falhas de agentes. Esta conclusão foi possível retirar devido ao facto de ter sido enviados *e-mails* de alerta relativos a estes agentes de uma forma exagerada. Mais concretamente foram enviados um ou dois *e-mails* de alerta de situações críticas por dia. Acrescido a este facto, foi possível analisar gráficos da *dashboard* que diz respeito aos *logs* dos agentes. Na Figura 5.14 é possível observar que durante dois dias e meio o agente contínuo 901 teve um comportamento de registo no seu ficheiro *log* pouco constante. É possível apurar picos na madrugada do dia 25 e 27 de agosto e na hora de almoço/início de tarde do dia 25. Para além disto, foi possível apurar através dos *e-mails* de alerta recebidos que, geralmente, estes agentes acusam situações críticas de madrugada. Este facto tem um explicação simples.

Estes agentes, principalmente os do grupo 9 (ver Tabela 4.1) são responsáveis pelas transferências do sistema. Por sua vez, estes agentes apenas registam a sua atividade no seu ficheiro *log* quando efetivamente realizam uma transferência. O que acontece é que quando não há nada para transferir, o agente não regista essa atividade e isto acontece normalmente à noite. Esta é a razão para o comportamento visualizado no gráfico da Figura 5.14 e de todos os *e-mails* de alerta recebidos neste âmbito. Esta questão pode ser facilmente contornada ao modificar estes agentes e outros contínuos, de modo a que estes registem periodicamente nos seus ficheiros *log* mesmo que não efetuem alguma atividade.

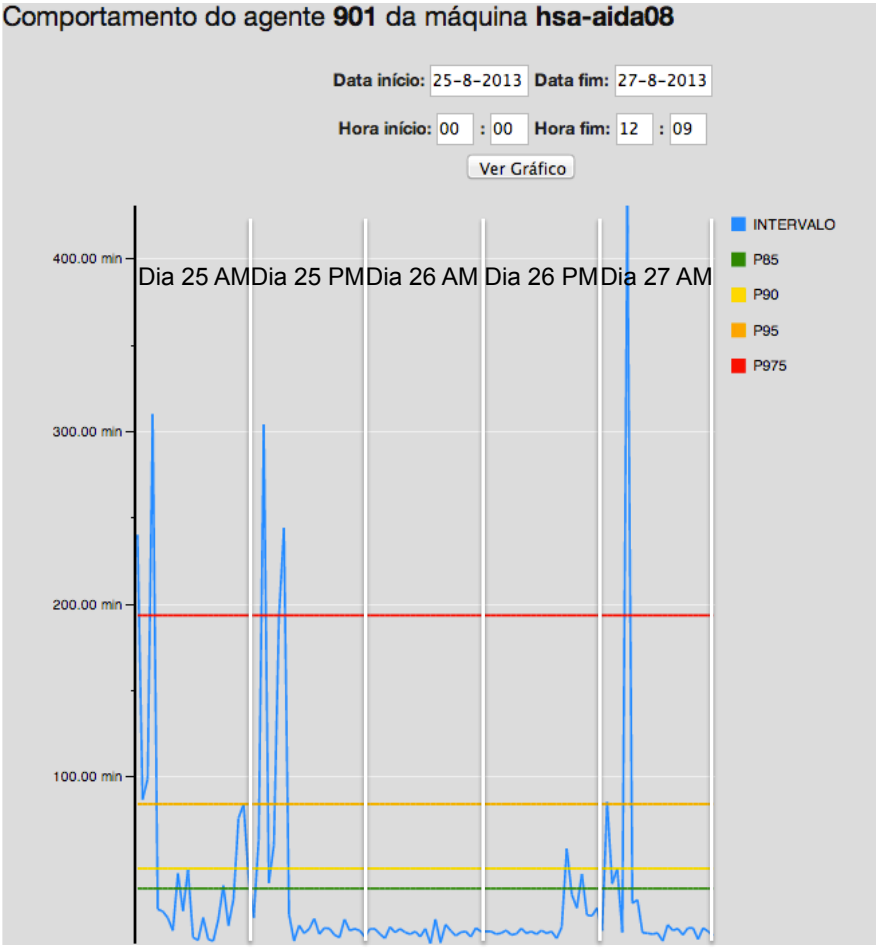

Figura 5.14: Excerto do *dashboard* referente ao agentes contínuo 901 da máquina hsa-aida08 da AIDA do CHP entre o dia 25 e 27 de agosto de 2013. A imagem foi modificada para melhor perceção. Retirado a 13-8-2013.

5.2.3 Interpretação dos Resultados

Alguns resultados já foram interpretados ao longo desta secção alusiva à carga de trabalho das máquinas e agentes da AIDA do CHP. Nas próximas linhas serão interpretados os resultados provenientes do cruzamento de informações da carga de trabalho dos agentes com a das máquinas.

Relativamente ao consumo de memória, é possível apurar que agentes contínuos, que são responsáveis por tarefas como transferência de arquivos e gestão de anexos, estão continuamente a serem executados e podem gerar

mais de que um executável na lista de processos, deste modo, naturalmente, consomem mais memória. Este facto é evidente ao comparar os gráficos referentes ao consumo de memória dos agentes momentâneos (Figura 5.7) e dos agentes contínuos (Figura 5.12). No primeiro a gama do consumo exibida encontra-se entre, aproximadamente, 12-15 *MBytes*, enquanto no gráfico relativo aos contínuos a gama situa-se entre os 37-500 *MBytes*, aproximadamente.

Desta forma, encontra-se justificado o porquê do alto consumo de memória de hsa-aida01 e 02 na Figura 5.4, pois estas máquinas na maioria executam agentes contínuos, como sugere a Tabela 4.1. Por outro lado, as máquinas hsa-aida04, 08 e 09 consomem menos memória pois executam, predominantemente, agentes momentâneos.

No que toca ao uso do processador, conclui-se que o facto que leva à hsa-aida08 a ter o maior consumo de CPU (14,17%) é a quantidade de agentes que esta máquina executa, acrescido do facto de grande parte deste agentes serem executados muito frequentemente. Através da tabela exibida na Figura 5.1, verifica-se que a máquina hsa-aida08, segundo o sistema baseado no registo nos ficheiros *log*, possui 10 agentes dos 20 que estavam ativos (gráfico circular à esquerda) no momento que a imagem foi capturada.

Voltando à análise dos gráficos da Figura 5.4 alusivos à utilização de CPU e memória das máquinas, é possível concluir que a máquina hsa-aida04 é a que dispões mais recursos disponíveis, nomeadamente os principais: processador e memória. Pode-se afirmar que novos agentes que venham a ser criados para esta plataforma AIDA, devem ser executados nesta máquina.

Em termos de balanceamento dos recursos principais das máquinas, pode-se afirmar que, segundo a Figura 5.4, a memória é a que apresenta maior diferença nos valores de consumo. Ignorando as máquinas que não executam agentes (hsa-aida19 e hsa-portint-tes), é visível uma diferença nítida de 15-20% de consumo de memória entre as máquinas hsa-aida01 e 02 e as máquinas hsa-aida04, 08 e 09. A primeira medida que ocorre de imediato perante este facto é transferir agentes das máquinas hsa-aida01 e 02 que consomem uma razoável quantia de memória para as outras três máquinas. Contudo esta pode não ser a melhor opção, antes de tomar essa medida é necessário con-

siderar várias variáveis. Parte dos agentes que são executados na hsa-aida01 e 02, por exemplo os que realizam transferências, não têm tanta importância como um agente responsável pela faturação ou processamento localizado na hsa-aida08. Por outras palavras, é mais conveniente tolerar uma falha de um agente do tipo SILAgTrn do que outro agente com funções fundamentais para o bom desempenho dos cuidados de saúde.

É importante referir que o número de máquinas que executam agentes da AIDA não deve ser exagerado, nem muito reduzido. Se houvesse muitas máquinas o sistema tornar-se-ia demasiado complexo para gerir ou monitorizar, para além do facto que haveria mais comunicações através da rede, o que tornaria o sistema mais lento no geral. Por outro lado, se houvesse um número de máquinas muito reduzido, para além de sobrecarregar as máquinas em questão, a importância da disponibilidade de cada máquina seria elevadíssima, sendo muito difícil assegurar essa disponibilidade. Por outras palavras, se uma desta máquinas falhassem, as consequências seriam mais graves.

A máquina hsa-aida19 encontra-se com uma memória livre de apenas 28,07% como mostra a Figura 5.4, devido à base de dados lá implementada e ao servidor da ferramenta *Pentaho Community* com já referido. No entanto, quanto à gestão de recursos desta máquina, recomenda-se que não lhe seja mais atribuída uma função que vá utilizar um gasto considerável de memória. É preciso ter em atenção também à quantidade de funções da hsa-portint-tes. Convém ao administrador estar atento a este facto.

De uma forma geral, pode-se concluir que este sistema encontra-se com os seus recursos bem balanceados. No entanto, pequenos reparos podiam ser feitos como por exemplo a transferência de um ou mais agentes da hsa-aida08 para a hsa-aida04. Deste modo, a elevada importância da alta disponibilidade da máquina hsa-aida08 seria descentralizada, ou seja, se esta máquina porventura falhasse, os danos não seriam tão grandes.

Para concluir esta secção sobre a carga de trabalho das máquinas e agentes da AIDA do CHP, encontram-se nos seguintes pontos um resposta sucinta à **Questão 5**:

- O sistema, em geral, encontra-se com os seus principais recursos das máquinas bem balanceados;
- A máquina hsa-aida08 provavelmente é a mais importante de todas as máquinas analisadas por executar muitos agentes e com funções importantes;
- As máquinas hsa-aida19 e hsa-portint-tes que não executam agentes, encontram-se com uma carga considerável. Convém não as sobrecarregar muito mais a fim de manter o bom funcionamento das mesmas;
- Agentes contínuos, de uma forma genérica, por gerarem mais executáveis e serem executados permanentemente, consomem mais memória;
- Os agentes momentâneos que se destacam mais é o 609 e o 39 pelo elevado tempo de execução e pelos elevados níveis de todos os indicadores de desempenho, respetivamente;
- O grupo de executáveis SILAgAna da hsa-aida01 são os que mais se destacam devido ao elevado consumo de memória oriundo da grande quantidade de executáveis;
- A frequência de atividade dos agentes contínuos é demasiada irregular ao ponto de não ser possível monitorizar este ponto nestes agentes;
- O espaço disponível nos discos de todas as máquinas é bem gerido através de mecanismos de transferências de arquivos para unidades apropriadas.

5.3 Análise da Deteção de Erros dos Agentes

Antes de analisar os resultados obtidos através dos sistemas prevenção de falhas, é apresentado nesta secção os resultados obtidos através dos *e-mails* de alerta enviados pela aplicação *MoniLogs* no que diz respeito à ocorrência de erros dos agentes. Como referido na secção 4.2, a aplicação *MoniLogs* para além de analisar a frequência com que cada agente regista no seu ficheiro *log*,

também analisa o ficheiro *log* dos erros que eventualmente possa existir na mesma diretoria.

Desta forma é possível apurar em quais agentes ocorrem mais erros, e assim o administrador, para além de ser avisado sempre que surge um novo erro, sabe qual o agente que deve ser modificado primeiramente. Deste modo é reduzido o máximo número de erros possíveis.

Esta análise aos *e-mails* foi automatizada através de programas desenvolvidos na linguagem *Google Apps Script*, assunto já mencionado na subsecção 3.4.3. Estes programas desenvolvidos têm como função contabilizar e registar informações contidas nos *e-mails* de alerta numa folha de cálculo do serviço *Drive* [1] que a *Google* disponibiliza. Para além de recolher e registar as informações em tabelas, estes programas também constroem gráficos a partir dos dados adquiridos.

As próximas figuras que se seguem, dizem respeito à análise dos *e-mails* de alerta de erros recebidos no período de 6 de setembro a 10 de outubro de 2013. É importante referir que mais *e-mails* foram recebidos antes deste período, contudo foi decidido apenas explorar este intervalo de tempo para não dificultar esta análise.

Na Figura 5.15 é possível visualizar a quantidade de *e-mails* de alerta enviados por agente no período de estudo. Cada *e-mail* contém um erro ocorrido, eventualmente pode conter dois ou mais erros se estes se sucederam num intervalo inferior a 10 minutos, pois a aplicação que deteta estes erros, *MoniLogs*, é executada de 10 em 10 minutos.

Através do gráfico da Figura 5.15 verifica-se que os agentes 36 e 491 são os que registam mais ocorrência de erros (258 e 272, respetivamente) seguido do agente 35 (150 erros). Conclui-se que a maioria dos agentes que registam erros são os da máquina hsa-aida08 e todos os agentes da máquina hsa-aida04 que estão presentes neste gráfico (agente 35 e 36 em destaque e ainda duas ocorrências do agente 609). Verifica-se também que não foi detetado nenhum erro dos agentes que são executados na máquina hsa-aida01 e apenas um na máquina hsa-aida02.

Na Figura 5.16 encontra-se um gráfico que representa a quantificação de

[1] https://drive.google.com

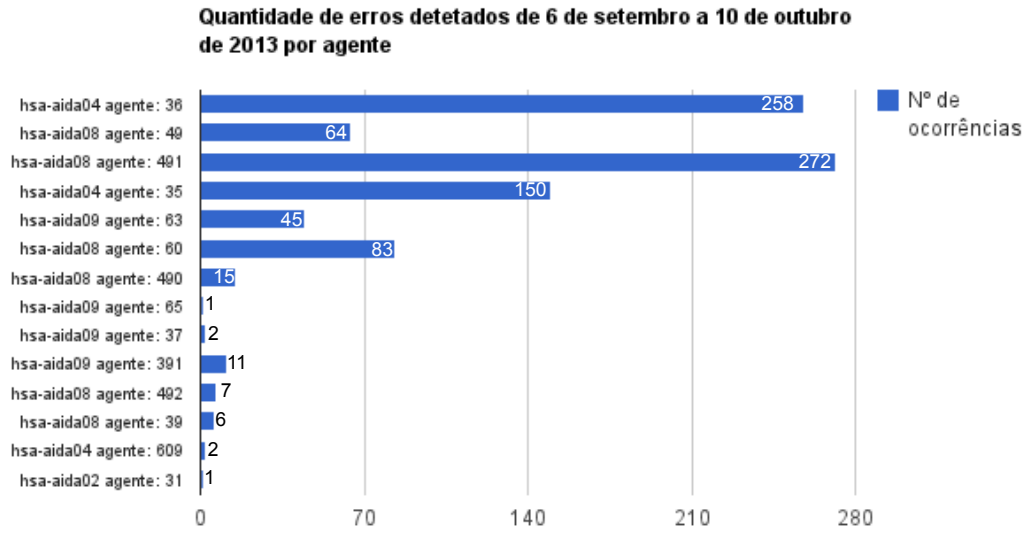

Figura 5.15: Quantidade de erros detetados no período de 6 de setembro a 10 e outubro de 2013 por agente.

erros detetados por máquina. Por outras palavras, neste gráfico é possível constatar a soma total de erros de todos os agentes de cada máquina. Facilmente rapara-se que as máquinas hsa-aida04 e 08 são as que possuem mais erros registados nos ficheiros *log*. Apesar de a hsa-aida08 ser líder neste gráfico, o valor de 410 erros registados na hsa-aida04 chama mais atenção devido ao facto de apenas 3 agentes serem executados nesta máquina, e um deles (agente 609) apenas ter registado 2 dos 410 erros. O elevado número de erros registados na hsa-aida08, apesar de ser elevado e merecer a devida atenção, é justificável devido ao elevado número de agentes que esta máquina executa.

Através dos programas desenvolvidos em *Google Apps Script* conseguiu-se guardar nas já mencionadas folhas de cálculo a data e hora dos *e-mails* de alerta recebidos. Desta forma, ordenando todas as data-horas de cada agente de forma descendente e calculando os intervalos de tempo de duas em duas ocorrências, foi possível calcular o tempo médio de ocorrência de erros em cada agente. O gráfico da Figura 5.17 exibe os resultados obtidos.

Para observar este gráfico em questão é preciso levar em conta os valores que se encontram no gráfico da Figura 5.15. Ao observar estes dois gráficos

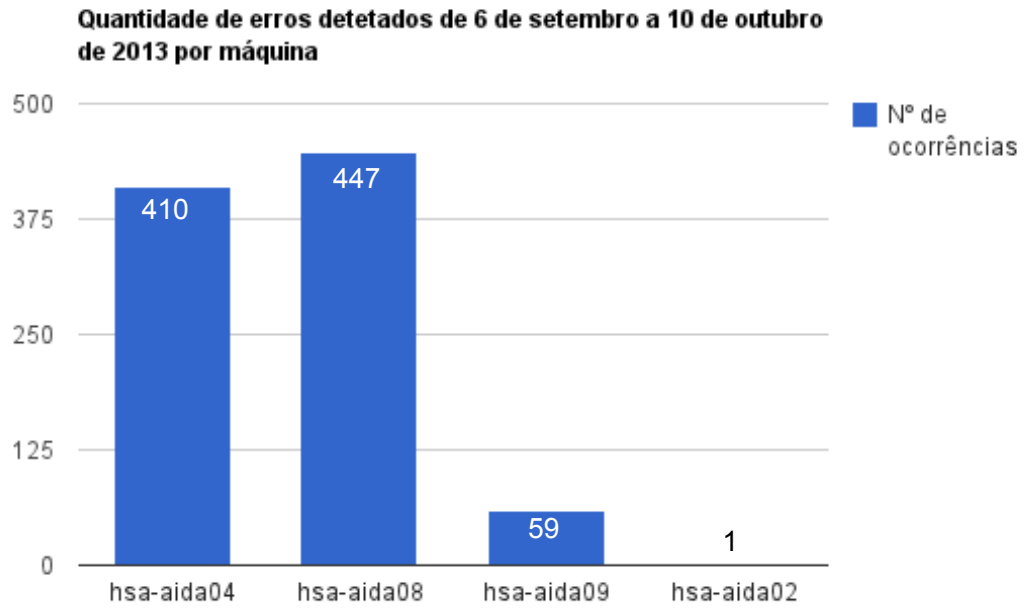

Figura 5.16: Quantidade de erros detetados no período de 6 de setembro a 10 e outubro de 2013 por máquina (soma dos erros de todos os agentes de cada máquina).

constata-se os seguintes factos:

1. Quanto maior o número de erros registados (Figura 5.15), maior é a frequência que se recebe *e-mails* de alerta;

2. Por outro lado, quanto menor o número de erros registados (Figura 5.15), menor é a frequência que se recebe *e-mails* de alerta;

3. Para os agentes 65 e 31 por ter-se registado apenas uma ocorrência durante o período de estudo, não foi possível calcular qualquer intervalo de tempo;

Os factos 1 e 2 são justificados pelo facto de quanto mais erros ocorrer num determinado período de tempo, é de se esperar que se receba mais frequentemente *e-mails* de alerta e vice-versa. Por exemplo, o agente 36 da hsa-aida04 registou um total de 258 erros no período de estudo (Figura 5.15), que ocorreram aproximadamente de 3 em 3 horas, mais concretamente 3 horas e 11

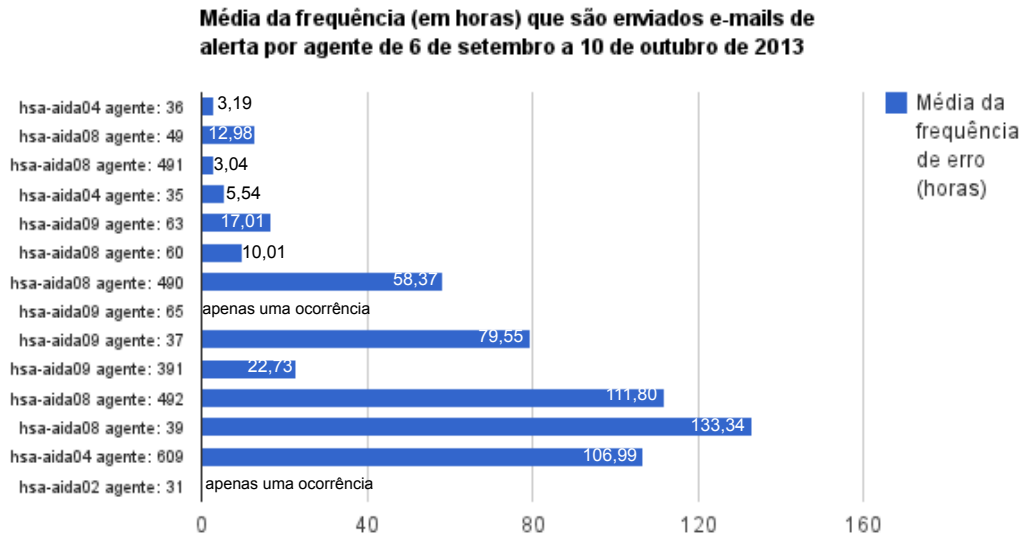

Figura 5.17: Média da frequência (em horas) que são enviados *e-mails* de alerta por agente. Período de 6 de setembro a 10 de outubro de 2013.

minutos (Figura 5.17). Por outro lado, o agente 39 que registou 6 erros que aconteceram, em média, num intervalo de 133 horas (aproximadamente 5 dias).

A conclusão que se retira de imediato é que deve-se atuar nos agentes que apresentam menores valores no gráfico da Figura 5.17, contudo, apesar dos agentes destes casos registarem mais erros, pode não significar que estes erros estão a prejudicar mais o bom funcionamento da AIDA do que os erros registados por agentes que raramente detetam erros. O único erro registado pelo agente 65, por exemplo, pode ser mais influente do que todos os erros do agente 36 para o correto funcionamento da plataforma de interoperabilidade hospitalar. A partir daqui, conclui-se que onde deve-se atuar primeiro não depende só da quantidade de erros registados. Apesar deste ser um fator imprescindível para a tomada de decisão por parte do administrador, é preciso identificar quais erros são mais graves e que comprometem mais o bom funcionamento da AIDA, para finalmente tomar as medidas corretivas necessárias.

5.4 Prevenção de Falhas (Questão 6)

Ao longo desta secção será abordada a **Questão 6** que consiste na apresentação e discussão dos resultados provenientes dos sistemas de prevenção de falhas desenvolvidos neste projeto e apresentados no Capítulo 4. Estes resultados são obtidos através de uma análise aos *e-mails* de alerta recebidos, quer das máquinas (*MoniMaqs*), quer dos agentes (*MoniLogs*). Esta análise aos *e-mails* é realizada por programas desenvolvidos na linguagem *Google Apps Script* e desempenham as mesmas funções dos programas utilizados para a análise de deteção de erros dos agentes (secção 5.3).

Primeiramente serão abordados os resultados do sistema de prevenção de falhas das máquinas e depois os provenientes do sistema de prevenção de falhas dos agentes baseados na frequência de atividade de cada um.

5.4.1 Máquinas

Na Tabela 5.1 é possível observar todas as ocorrências de *e-mails* de alerta produzidas pela aplicação *MoniMaqs*, bem como a quantificação de cada *score* por máquina. A ocorrência mais antiga apresentada é no dia 30 de julho de 2013 e a mais recente dia 25 de setembro. Quer a tabela, quer o gráfico foram obtidos através dos programas desenvolvidos em *Google Apps Script*.

Ao analisar a Tabela 5.1 é possível identificar 6 situações críticas, agrupando as ocorrências por máquina e dia (a ocorrência de 25 de setembro da hsa-aida02 pode ser agrupada com as ocorrências do dia 24 de setembro).

Relativamente à máquina hsa-aida02, pode-se apurar que em dois momentos esta máquina esteve num estado crítico. A 30 de julho de 2013 a aplicação *MoniMaqs* detetou um *score* total de 5 as 16:00 horas e após alguns minutos o estado da máquina alcançou o *score* total de 6. Através do *e-mail* de alerta recebido pode-se afirmar que a hsa-aida02 encontrava-se com 3% de CPU livre e 9% de memória livre. Nesta situação um dos administradores da AIDA foi informado e tomou as medidas necessárias, prevenindo que esta situação causa-se falhas. Quanto à situação do dia 24/25 de setembro,

Tabela 5.1: Data e hora de ocorrências de *scores* críticos por máquina.

Máquina	Data	Hora	Score
hsa-aida02	30 julho 2013	16:00	5
		16:26	6
	24 setembro 2013	22:40	4
		22:53	5
		23:05	4
		23:18	5
		23:31	5
		23:45	6
		23:58	6
	25 setembro 2013	00:12	6
hsa-aida04	27 agosto 2013	12:41	4
	5 setembro 2013	16:14	4
		17:31	4
hsa-aida19	22 agosto 2013	15:44	5
		16:19	5
	27 agosto 2013	11:03	5
		11:24	5

a máquina ficou demasiado tempo num estado crítico, desde as 22:40 horas até as 00:12 horas do dia 25.

Nas duas situações críticas que dizem respeito à máquina hsa-aida04, detetou-se que o parâmetro do CPU encontrava-se com o *score* máximo. Estas duas situações ocorreram devido à mesma causa: um processo que estava a ser executado nesta máquina estava a fazer uma gestão inapropriada dos seus *threads*, ocupando todo o processador. Este processo foi identificado e terminado. Relembrando o gráfico da Figura 5.11, constata-se que este programa foi a causa que fez com que o agente 609 parasse de funcionar na tarde do dia 5 de setembro.

Quanto às situações detetadas na máquina hsa-aida19, verificou-se através dos *e-mails* de alerta recebidos que o excessivo consumo de CPU foi devido a um processo semelhante ao que acabou de ser descrito nas situações críticas sucedidas na hsa-aida04. O facto de o *score* total ter-se fixado em 5 e não em 4 como na máquina hsa-aida04, foi porque a utilização de memória da hsa-

aida19 nestes momentos rondava o intervalo de 25 a 15 % livre de memória, o que acrescentou um valor ao *score* total.

Conclui-se que aconteceram duas situações críticas para cada uma das três máquinas apresentadas na Tabela 5.1 durante o intervalo de 30 de julho a 25 de setembro. Ao analisar a coluna referente aos *scores* totais da dita tabela, pode-se apurar que as situações críticas da máquina hsa-aida02 são as mais severas, pois apresentam *scores* totais de 5 e 6 maioritariamente, enquanto que a hsa-aida04 teve situações com um *score* total de 4, e a hsa-aida19 de 5.

A partir destes resultados chega-se à conclusão que é preciso estar atento ao comportamento da máquina hsa-aida02 e rever as tarefas de alguns agentes que tenham provocado estas situações. Quanto as situações críticas das máquinas hsa-aida04 e 19, os programas que estavam a fazer uma gestão inapropriada dos seus *threads* já se encontram corrigidos. Mesmo assim, estes resultados provaram a eficácia do sistema de prevenção de falhas das máquinas em situações críticas.

5.4.2 Agentes

No que toca aos *e-mails* de alerta recebidos relativos ao sistema de prevenção de falhas dos agentes, foi necessário filtrar alguns resultados para ser possível analisar o desempenho deste sistema. Mais concretamente, teve-se que ignorar *e-mails* de alerta oriundos de determinados agentes como o 901 e 913 pois estes enviam demasiados alertas por dia. Esta situação já foi debatida na parte dos agentes contínuos na subsecção 5.2.2.

Na Tabela 5.2 encontram-se situações críticas detetadas pelo sistema de prevenção de falhas de agentes durante o mês de julho e agosto de 2013. Durante este período de estudo, constatou-se através da Tabela 5.2 que nos dias 23, 25 e 31 de julho vários agentes pararam de funcionar durante um largo período de tempo (acima dos respetivos percentis 97,5). O que aconteceu realmente nestas três situações foi que o sistema encontrava-se em manutenção, e para tal, vários agentes foram interrompidos. Este facto prova que se algum agente realmente falhasse, a situação seria rapidamente detetada e re-

Tabela 5.2: Data e hora do último registo no ficheiro *log* por agente em situações críticas detetadas em julho e agosto de 2013.

Data	Hora	Máquina	Agente
22 julho 2013	11:52	hsa-aida09	391
23 julho 2013	15:18	hsa-aida08	39
			490
			491
			492
			59
			60
25 julho 2013	20:07	hsa-aida02	31
	20:10	hsa-aida08	39
			49
			490
			491
			492
			59
	20:11	hsa-aida08	60
		hsa-aida09	37
			38
			391
			61
			63
31 julho 2013	12:00	hsa-aida08	39
			49
			490
			491
			492
			59
			60
	12:01	hsa-aida09	37
5 agosto 2013	13:22	hsa-aida08	490

portada aos responsáveis para tomarem as medidas necessárias e prevenirem maiores danos.

Quanto à ocorrência do dia 22 de julho e 5 de agosto da Tabela 5.2, constatou-se que estes agentes, 391 e 490 respetivamente, ficaram demasiado tempo parados. Tratou-se de situações anormais detetadas pelo sistema de prevenção de falhas dos agentes. Nas duas situações referidas, ambos agentes já não escreviam há mais de 15 minutos nos seus ficheiros *log* de registo, quando o comportamento normal destes agentes é fazer um ou até dois registos por minuto.

Outro período de estudo foi realizado para analisar o desempenho do sistema de prevenção de falhas de agentes: 25 de agosto a 13 de outubro de 2013. No gráfico da Figura 5.18 encontra-se a quantidade de *e-mails* de alerta por agente nesse período.

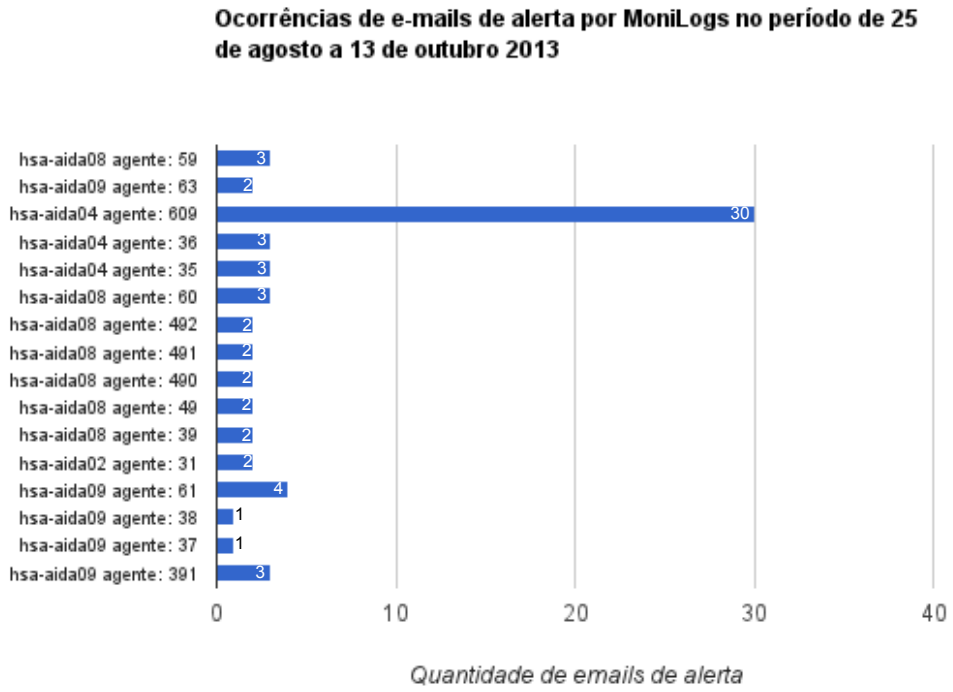

Figura 5.18: Quantidade de *e-mails* de alerta por agente em relação à frequência de atividade. Período de 25 de agosto a 13 de outubro de 2013.

Os resultados oriundos dos agentes do grupo 9 (Tabela 4.1) foram ignorados por motivos já explicados. Sendo assim, no gráfico da Figura 5.18 é possível constatar que o agente que acusou mais situações críticas segundo *MoniLogs* foi o 609. Estas situações, na maior parte das vezes, não se trataram de situações críticas. O que acontece é que, como já verificado na subsecção 5.2.2, o agente 609 tem um tempo de execução muito longo o que faz com que o seu registo no ficheiro *log* seja algo que é feito em intervalos pouco constantes (semelhante ao que acontece aos agentes de transferências). Deste modo, apesar de *MoniLogs* tentar ajustar os percentis de limites deste agente, não é suficiente para impedir que o percentil 97,5 seja ultrapassado várias vezes. Relembrando novamente o caso da Figura 5.11, um *e-mail* de alerta por *MoniLogs* também foi enviado por volta das 16:15 horas do dia 5 de setembro. O agente 609 realmente não estava a funcionar devido a razões já mencionadas.

Relativamente aos outros agentes, as situações detetadas foram reais, contudo de duas naturezas distintas: situações de manutenção onde vários agentes foram prepositadamente interrompidos e situações em que os agentes realmente não registavam a sua atividade há demasiado tempo. Segundo os *e-mails* de alerta foi possível apurar uma situação de manutenção no dia 10 de outubro. Entre aproximadamente as 18:20 e as 18:45 horas foram enviados *e-mails* de alerta por parte de 4 agentes que são executados na hsa-aida09, os 3 agentes da hsa-aida04, 7 agentes da hsa-aida08 e ainda o agente 31 da hsa-aida02.

A Figura 5.19 mostra um novo gráfico relativo aos *e-mails* de alerta enviados no mesmo período, mas desta vez foram removidos os casos referentes ao agente 609 e todos os casos da manutenção do dia 10 de outubro.

Nota-se claramente pelo gráfico da Figura 5.19 que as ocorrências desceram drasticamente com a remoção dos casos da manutenção e com os casos do agente 609. Através deste gráfico e comparando as datas de registo nos ficheiros *log* recebidas nos *e-mails* de alerta, verifica-se que as datas dos dois casos dos agentes 35 e 36 da máquina hsa-aida04 são iguais e que essas datas também estão presentes nos *e-mails* referentes ao agente 609. Não é possível apurar se estas situações trataram-se de uma falha ou então simplesmente de

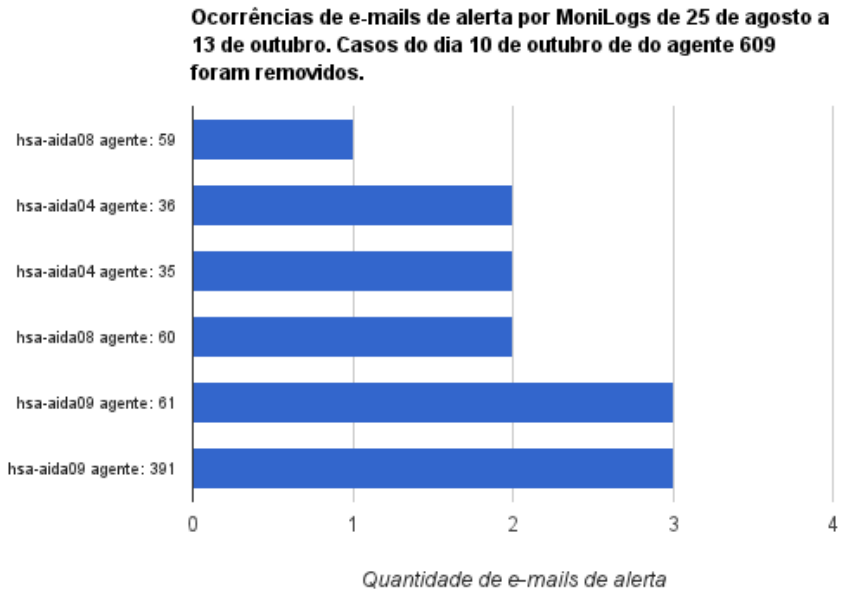

Figura 5.19: Quantidade de *e-mails* de alerta por agente de 25 de agosto a 13 de outubro de 2013. Foram removidos os casos da situação de manutenção de 10 de outubro e os referentes ao agente 609.

uma manutenção. Aos outros casos apresentados na Figura 5.19, trataram-se de situações em que os agentes ficaram demasiado tempo parados. Estas situações devem ser analisadas para apurar se é necessário modificar esses agentes ou então outros componentes que possam ter provocado essas situações anormais.

Em resposta à **Questão 6** pode-se dizer que de uma forma geral, concluiu-se que a prevenção de falhas das máquinas e agentes da AIDA por parte dos sistemas desenvolvidos foi alcançada. Situações críticas, ou situações de manutenção que simulam situações críticas, foram detetadas. No caso das máquinas detetou-se situações críticas na hsa-aida02, 04 e 19 durante o período de estudo. Analisando as causas destas falhas, conclui-se que a hsa-aida02 é a que merece maior atenção por parte dos administradores da AIDA. Relativamente aos agentes, verificou-se que algumas arestas têm de ser moldadas no sistema de prevenção de falhas dos agentes, nomeadamente

nos casos referentes aos agentes contínuos e ao agente 609 que possui uma atividade inconstante e um longo tempo de execução. As situações de manutenção mostraram a eficácia do sistema desenvolvido.

Antes de concluir este capítulo, é de elevada importância referir que a monitorização e prevenção de falhas de componentes da AIDA contribui para o aumento da integridade de todo o sistema, uma vez que o controlo sob as máquinas e agentes desta plataforma é amplo. Para além disto, os sistemas desenvolvidos neste projeto fazem aumentar a disponibilidade de todo o sistema através da deteção de situações críticas onde falhas podem ocorrer facilmente. A alta disponibilidade de um sistema como a AIDA é um aspeto de bastante relevância na segurança da informação hospitalar.

Capítulo 6

Conclusões

As plataformas de interoperabilidade hospitalar como a AIDA implementada no CHP têm uma grande importância na qualidade de serviços prestados, portanto esta deve possuir um alto nível de disponibilidade de forma a garantir um funcionamento apropriado de todo o sistema. Os agentes inteligentes, e consequentemente as máquinas onde são executados, são componentes fundamentais na AIDA. Ao assegurar um elevado nível de disponibilidade desses componentes, contribui-se para o aumento da disponibilidade global da AIDA. Para alcançar tal nível de disponibilidade nestes componentes e consequentemente garantir uma boa qualidade dos serviços prestados, é necessário prevenir possíveis falhas que possam ocorrer.

Conclui-se que antes da prevenção de falhas é imprescindível recorrer aos processos de monitorização dos componentes envolvidos e previsão de falhas dos mesmos. Monitorizar, prever e prevenir. Esta é a ordem dos processos para conseguir uma prevenção de falhas eficaz.

6.1 Contributos

Constatou-se que é possível dividir os agentes da AIDA em dois tipos: contínuos e momentâneos. Os primeiros estão permanentemente a serem executados (encontram-se sempre na lista de processos da máquina) e normalmente estes agentes geram mais do que um ficheiro executável. Quanto

aos agentes momentâneos, são executados pontualmente e geram apenas um único ficheiro executável. Apura-se também que os momentâneos apresentam funções de processamento e faturação, enquanto os contínuos são responsáveis por partilhas de ficheiros, pedidos, gestão de anexos e transferências.

Os indicadores de desempenho selecionados (**Questão 1**) para monitorizar foram percentagem de CPU, memória e disco livre para as máquinas e memória, percentagem de CPU, I/O de dados por segundo, tempo de execução ou número de processos (agentes momentâneos e contínuos respetivamente) para os agentes da AIDA. Outro indicador dos agentes muito importante para este projeto é a frequência de atividade, que representa a frequência com que determinado agente regista no seu ficheiro *log* a sua atividade. A tecnologia WMI através de *queries* WQL acrescido das potencialidades da plataforma *.NET* são ferramentas poderosas para a recolha dos indicadores propostos (resposta à **Questão 4**).

Ainda no campo da monitorização, foi possível estudar a carga normal de trabalho das máquinas e agentes (**Questão 5**). Conclui-se que, globalmente, os recursos componentes analisados da AIDA encontram-se balanceados. Considera-se a máquina hsa-aida08 a mais influente uma vez que executa muitos agentes e com funções importantes. Uma das medidas que podia ser implementada seria a transferência de um ou mais agentes da hsa-aida08 para a hsa-aida04, visto que esta máquina apresenta uma carga amena e desta forma, uma possível falha na hsa-aida08 não teria danos tão colossais. Verificou-se que o número de máquinas da AIDA não deve ser excessivo nem reduzido. Se o sistema tiver muitas máquinas o processo de monitorização seria mais complexo, a gestão por parte dos administradores seria dificultada e as comunicações pela rede seriam excessivas. Noutra perspetiva, com um número reduzido, as máquinas seriam sobrecarregadas e uma falha em qualquer uma das máquinas provocaria danos muito graves.

Alusivamente à carga dos agentes, comprovou-se que os agentes contínuos por gerarem mais executáveis e por estarem sempre a serem executados consomem mais memória. A máquina com maior consumo de CPU no período de estudo foi a hsa-aida08 devido ao elevado número de agentes que são executados nesta máquina e por estes agentes, na maioria, serem momentâneos

e serem executados frequentemente. No que diz respeito à frequência de atividade, certificou-se que esta é bastante irregular nos agentes contínuos, não sendo possível monitorizar este indicador neste tipo de agente. Através das informações recolhidas sobre o espaço de disco livre das máquinas, conclui-se que este espaço é bem gerido através de mecanismos de transferências de arquivos eficientes. Os gráficos que permitiram investigar a carga de trabalho destes componentes foram elaborados através da ferramenta de BI *Pentaho Community* com o módulo CDE instalado (também resposta à **Questão 4**). Esta ferramenta mostrou-se muito eficiente para este propósito.

Quanto à previsão de falhas, conclui-se que a adaptação do modelo MEWS ao desempenho das máquinas e agentes da AIDA é possível e detetou situações críticas propícias a falhas. Acredita-se que estas adptações com alguns ajustes se tornarão em ferramentas com uma elevada potencialidade para auxiliar os administradores do sistema e prosperar a qualidade dos serviços prestados no CHP. Na elaboração destes modelos (**Questão 3**) constatou-se que o modelo dos agentes, baseado na frequência de atividade, para efeitos do cálculo dos *scores* deve seguir os limites dos percentis 85, 90, 95 e 97,5. Deste modo, os limites são atualizados consoante o comportamento de determinado agente ao longo do tempo e as situações críticas são detetadas mais claramente. Quanto ao modelo das máquinas comprovou-se que o cálculo dos *scores* baseado nos percentis dos indicadores de desempenho não são uma solução eficaz. Para retificar este problema, optou-se por atribuir limites fixos a cada indicador que foram discutidos e cuidadosamente selecionados. É importantíssimo referir que estes limites podem ser alterados ao longo do tempo através de uma página de administração também desenvolvida neste projeto. Cada limite de cada parâmetro de cada máquina pode ser modificado, desta forma através do conhecimento que os administradores possuem acerca do sistema, este modelo pode ser aperfeiçoado.

Após a monitorização e a previsão, finalmente alcança-se a prevenção de falhas. Através dos sistemas desenvolvidos que monitorizam os componentes em questão e que aplicam os modelos de previsão criados, situações críticas são detetadas e *e-mails* de alerta são enviados para que os administradores rapidamente analisem a situação e tomem as ações preventivas necessárias.

Quanto aos resultados obtidos pelos sistemas de prevenção desenvolvidos, nomeadamente os *e-mails* de alerta, a ferramenta *Google Apps Script* revelou-se fundamental para analisar detalhadamente todos os *e-mails* enviados para uma conta *Google* criada para este projeto (também resposta à **Questão 4**).

O sistema de monitorização e prevenção de falhas dos agentes baseado nos *logs*, permitiu também analisar os *logs* de erros. Durante o período de estudo selecionado, detetou-se que ocorrem em média erros de 3 em 3 horas em determinados agentes. A conclusão que se tira é que apesar destes agentes registarem um maior número de erros, a prioridade da ação por parte dos administradores pode não ser estes agentes e sim os outros que revelaram erros muito pontuais. Estes erros que são mais raros provavelmente significam que causam danos mais influentes para o sistema. A quantidade de erros identificados deve ser apenas uma variável na tomada de decisão, é necessário investigar os erros ocorridos e identificar quais são os mais comprometedores para o funcionamento apropriado da AIDA.

Demonstrou-se que os sistemas de prevenção de falhas quer para as máquinas, quer para os agentes da AIDA desenvolvidos são capazes de prever falhas nos respetivos componentes (**Questão 6**). Situações críticas das máquinas hsa-aida02, 04 e 19 foram detetadas, os administradores foram devidamente informados e medidas preventivas foram tomadas. No caso da prevenção de falhas dos agentes, constatou-se que apesar deste sistema necessitar um aperfeiçoamento nos casos dos agentes contínuos, este revelou-se capaz de detetar situações anormais. As situações de manutenção que provocou a paragem de vários agentes contribuem para reforçar a capacidade do sistema de prevenção de falhas em agentes desenvolvido. Isto significa que, numa situação extrema, se vários agentes parassem o sistema de prevenção identificaria e reportaria a situação para os administradores do sistema.

A principal conclusão que se retira deste projeto é que os sistemas desenvolvidos contribuem para o aumento da integridade da AIDA implementada no CHP e sobretudo para um aumento da alta disponibilidade que esta plataforma deve possuir.

Foram produzidos dois artigos científicos com trabalho desenvolvido neste projeto. O primeiro, que é um capítulo de livro (Apêndice B.2), integra vá-

rias secções acerca da interoperabilidade na saúde e uma destas secções diz respeito ao trabalho elaborado neste projeto. Por outro lado, o conteúdo do outro artigo produzido, que se encontra no Apêndice B.4, foca-se principalmente no trabalho desenvolvido nesta dissertação. Contudo, este encontra-se em fase de revisão por pares. É importante referir que outros dois artigos científicos foram produzidos (Apêndices B.1 e B.3) e que serviram para ampliar o conhecimento da plataforma AIDA e de conceitos como interoperabilidade, SMAs e Ambiente Inteligente.

6.2 Trabalho Futuro (Questão 7)

Esta última secção desta dissertação tem como objetivo responder à **Questão 7**, última questão proposta na secção 1.2.

Uma vertente que pode ser explorada diz respeito aos erros identificados e registados nos ficheiros *log* de erros dos agentes. Ao detetar o erro, pode-se analisar o respetivo registo no *log* e desencadear mecanismos automáticos para corrigi-lo ou suavizar os seus danos. Obviamente que para atingir este objetivo é necessário um vasto estudo acerca dos erros que ocorrem e é preciso implementar um processo de categorização do erro, que o encaminhe para os mecanismos apropriados. Mecanismos semelhantes também podem ser aplicados a situações vulneráveis a falhas.

Pequenos aspetos que podem ser acrescentados aos sistemas desenvolvidos são: criar uma lista de *e-mails* de administradores e enviar os alertas apenas aos administradores disponíveis e consoante a falha e o componente em questão; incluir nos *e-mails* de alerta aviso de receção, caso a mensagem não tivesse sido lida após um tempo pré-definido e alterável, o *e-mail* seria enviado para outro administrador ou então uma *Short Message Service* (SMS) seria encaminhada (o procedimento de alerta a seguir podia ser definido pela gravidade da situação). Outros pequenos aspetos, relativamente ao *website* de monitorização desenvolvido, seria tornar possível ligar o servidor do *Pentaho Community* remotamente e visualizar os conteúdos dos ficheiros *log*, mas para estas funções é necessário redefinir os acessos às máquinas da AIDA. No *website* de monitorização ainda podia ser adicionado uma opção

para os administradores interromperem o módulo de alerta dos sistemas de monitorização e prevenção nos casos de manutenção. Desta forma, as situações de manutenção não seriam reportadas como situações críticas.

Mais parâmetros podem ser adicionados aos modelos de previsão e através de um próximo acompanhamento ao longo de um considerável período de tempo, seria possível eleger os parâmetros que são mais influentes para o bom funcionamento de cada componente. Relativamente aos limites destes modelos, podem-se ser apurados através de situações de falhas reais novos limites que representam se determinado componente encontra-se vulnerável ou não.

No entanto, mudanças mais drásticas podem ser feitas. Para o sistema de recolha de dados e processamento das informações para identificar situações críticas, podem ser usados agentes inteligentes integrados no SMA da AIDA. Apesar da boa prestação da adaptação do modelo MEWS neste projeto, outras técnicas de previsão podem ser utilizadas de forma a ajustarem-se melhor à natureza dos componentes. Nomeadamente ferramentas de *data mining* para reconhecer correlações e padrões nos comportamentos dos componentes e métodos estatísticos para efeitos de previsão. Através de um profundo estudo do comportamento de cada componente é possível implementar influentes sistemas de apoio à decisão para haver uma ótima gestão de recursos da plataforma de interoperabilidade hospitalar. Algumas normas podem ser definidas para serem seguidas na implementação de novos agentes, máquinas ou outros componentes. Estas mudanças drásticas podem também ser aplicadas ao trabalho desenvolvido em [17–19] acerca da monitorização e prevenção de falhas das base de dados.

Acredita-se que usando como base o trabalho apresentado nesta dissertação e levando em conta estas propostas, que são a resposta à derradeira **Questão 7**, é possível implementar um sistema capaz de monitorizar e prevenir sérias falhas dos "sinais vitais" (componentes) de uma plataforma como a AIDA e também capaz de enaltecer a integridade e, principalmente, a disponibilidade de uma plataforma de interoperabilidade hospitalar.

Bibliografia

[1] E. Ammenwerth, J. Brender, P. Nykänen, H.-u. Prokosch, M. Rigby, and J. Talmon, "Visions and strategies to improve evaluation of health information systems Reflections and lessons based on the HIS-EVAL workshop in Innsbruck," *International Journal of Medical Informatics*, 2004.

[2] L. Mota, "Sistemas de informação de enfermagem: um estudo sobre a relevância da informação para os médicos," Master's thesis, Mestrado de Informática Médica, Universidade do Porto, 2010.

[3] P. L. Reichertz, "Hospital information systems-past, present, future," *International Journal of Medical Informatics*, vol. 75, no. 3-4, pp. 282 – 299, 2006, electronic Health Record, Healthcare Registers and Telemedicine and European Potential for Building Information Society in Healthcare. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1386505605002091

[4] J. Weber-Jahnke, L. Peyton, and T. Topaloglou, "ehealth system interoperability," *Information Systems Frontiers*, vol. 14, no. 1, pp. 1–3, 2012.

[5] 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.

[6] L. Palazzo, P. Sernani, A. Claudi, G. Dolcini, G. Biancucci, and A. F. Dragoni, "A multi-agent architecture for health information systems," in *International Workshop on Artificial Intelligence and NetMedicine*, 2013, p. 41.

[7] H. Peixoto, M. Santos, A. Abelha, and J. Machado, "Intelligence in Interoperability with AIDA," in *20th International Symposium on Methodologies for Intelligent Systems*, vol. 7661, World Intelligence Congress, Macau. Lecture Notes in Computer Science - Springer, 2012.

[8] Health Level Seven International. About HL7. [Online]. Available: http://www.hl7.org/about/index.cfm?ref=nav

[9] 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.

[10] M. Miranda, J. Duarte, A. Abelha, J. Machado, and J. Neves, "Interoperability in healthcare," in *Proceedings of the European Simulation and Modelling Conference (ESM 2010)*, Belgium, 2010.

[11] J. Machado, M. Miranda, P. Gonçalves, A. Abelha, J. Neves, and J. A. Marques, "AIDATrace: interoperation platform for active monitoring in healthcare environments," 2010.

[12] J. Machado, V. Alves, A. Abelha, and J. Neves, "Ambient intelligence via multiagent systems in the medical arena," *Engineering intelligent systems for electrical engineering and communications*, vol. 15, no. 3, pp. 151–157, 2007.

[13] 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.

[14] E. Bertino and R. Sandhu, "Database security - concepts, approaches, and challenges," *Dependable and Secure Computing, IEEE Transactions on*, vol. 2, no. 1, pp. 2–19, 2005.

[15] A. Rodrigues, *Oracle 10g e 9i: Fundamentos Para Profissionais*. Lisboa: FCA, 2005.

[16] S. Kim, N. Cho, Y. Lee, S.-H. Kang, T. Kim, H. Hwang, and D. Mun, "Application of density-based outlier detection to database activity monitoring," *Information Systems Frontiers*, pp. 1–11, 2010.

[17] P. Silva, C. Quintas, J. Duarte, M. Santos, J. Neves, A. Abelha, and J. Machado, "Hospital database workload and fault forecasting," in *2012 IEEE EMBS Conference on Biomedical Engineering and Sciences*, Malaysia, 2012.

[18] P. Silva, C. Quintas, M. Santos, A. Abelha, and J. Machado, "Step towards fault forecasting in hospital information systems," in *5th IEEE International Conference on Biomedical Engineering and Informatics*, China, 2012.

[19] P. Silva, C. Quintas, P. Gonçalves, G. Pontes, M. Santos, A. Abelha, and J. Machado, "Intelligent Systems based in Hospital Database Malfunction Scenarios," in *4ht IEEE International Conference on Industrial Engineering and Engineering Management*, Hong Kong, 2012.

[20] Capgemini Portugal, Serviço de Consultoria e Informática, S.A., "Definição da arquitectura de interoperabilidade dos sistemas de informação integrada na saúde: Relatório de características, requesitos e práticas," 2009.

[21] A. Winter, R. Haux, E. Ammenwerth, B. Brigl, N. Hellrung, and F. Jahn, *Health Information Systems: Architectures and Strategies: 2nd Edition*. Springer, 2011.

[22] A. Jin and L. Ahlfors, "Interoperability in electronic medical records," Master's thesis, School of Economics and Management, Department of informatics, Lund University, June 2012.

[23] A. P. Sheth, "Changing focus on interoperability in information systems: from system, syntax, structure to semantics," in *Interoperating geographic information systems*. Springer, 1999, pp. 5–29.

[24] T. Andreas, D. Saikou, and T. Charles, "Applying the levels of conceptual interoperability model in support of integratability, interoperability, and composability for system-of-systems engineering," *Journal of Systemics, Cybernetics and Informatics*, 2007.

[25] M. Miranda, G. Pontes, P. Gonçalves, H. Peixoto, M. Santos, A. Abelha, and J. Machado, "Modelling intelligent behaviours in multi-agent based hl7 services," *Computer and Information Science 2010*, vol. 317, pp. 95–106, 2010.

[26] National Electrical Manufacturers Association. DICOM - Digital Imaging and Communications in Medicine. [Online]. Available: http://medical.nema.org/

[27] 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.

[28] C. D. Carr and S. M. Moore, "IHE: a model for driving adoption of standards," *Computerized Medical Imaging and Graphics*, vol. 27, no. 2, pp. 137–146, 2003.

[29] H. Peixoto, J. Machado, J. Neves, and A. Abelha, "Semantic interoperability and health records," in *E-Health*. Springer, 2010, pp. 236–237.

[30] A. Abelha, M. Machado, M. Santos, A. Sollari, F. Rua, M. Paiva, and J. Neves, "Agency for archive, integration and diffusion of medical information," *Proceedings of the Third IASTED International Conference - Artificial Intelligence and Applications(ISBN 0-88986-390-3), Malaga, Spain*, 2003.

[31] 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.

[32] G. Weiss, *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 2000.

[33] E. A. Waldman, "Usos da vigilância e da monitorização em saúde pública," *Informe Epidemiológico do SUS*, vol. 7, no. 3, pp. 7–26, 1998.

[34] J. Boudec, *Performance Evaluation of Computer and Communication Systems*, ser. Computer and communication sciences. EPFL Press, 2010.

[35] G. Stoneburner, A. Goguen, and A. Feringa, "Risk management guide for information technology systems," *Nist special publication*, vol. 800, no. 30, pp. 800–30, 2002.

[36] P. Zhang, H. Muccini, A. Polini, and X. Li, "Run-time systems failure prediction via proactive monitoring," in *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*. IEEE Computer Society, 2011, pp. 484–487.

[37] C. F. Portela, M. F. Santos, Á. Silva, J. Machado, and A. Abelha, "Enabling a pervasive approach for intelligent decision support in critical health care," in *ENTERprise Information Systems*. Springer, 2011, pp. 233–243.

[38] F. Portela, P. Gago, M. F. Santos, Á. Silva, F. Rua, J. Machado, A. Abelha, and J. Neves, "Knowledge discovery for pervasive and real-time intelligent decision support in intensive care medicine," in *KMIS 2011- International Conference on Knowledge Management and Information Sharing*, 2011.

[39] A. Albino and V. Jacinto, "Implementação da escala de alerta precoce - EWS," Centro Hospitalar do Barlavento Algarvio, EPE, Portimão, Tech. Rep., 2010.

[40] C. Subbe, M. Kruger, P. Rutherford, and L. Gemmel, "Validation of a modified early warning score in medical admissions," *Qjm*, vol. 94, no. 10, pp. 521–526, 2001.

[41] G. Devaney and W. Lead, "Guideline for the use of the modified early warning score (MEWS)," Outer North East London Community Services, Tech. Rep., 2011.

[42] J. Gardner-Thorpe, N. Love, J. Wrightson, S. Walsh, and N. Keeling, "The value of Modified Early Warning Score (MEWS) in surgical in-patients: a prospective observational study." *Annals of the Royal College of Surgeons of England*, vol. 88, no. 6, pp. 571–575, 2006.

[43] C. P. Subbe, R. G. Davies, E. Williams, P. Rutherford, and L. Gemmell, "Effect of introducing the Modified Early Warning score on clinical outcomes, cardio-pulmonary arrests and intensive care utilisation in acute medical admissions," *Anaesthesia*, vol. 58, no. 8, pp. 797–802, 2003.

[44] J. McNiff, *Action research: Principles and practice*. Routledge, 2013.

[45] A. Fairbanks, J. LeCam, and T. McCabe, "Microsoft .net: understanding the.net framework and developing applications using .net," *J. Comput. Sci. Coll.*, vol. 18, no. 5, pp. 291–292, May 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=771832.771909

[46] Microsoft Corporation. Getting Started with the .NET Framework. [Online]. Available: http://msdn.microsoft.com/en-us/library/hh425099.aspx

[47] L. Ashdown and T. Kyte, *Oracle Database Concepts, 11g Release 2 (11.2).* Oracle, 2013.

[48] Microsoft Corporation. (2013) Win32_PerfFormattedData_PerfOS_Processor class. [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa394271(v=vs.85).aspx

[49] Microsoft Corporation. (2013) Win32_PerfFormattedData_PerfOS_Memory class. [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa394268(v=vs.85).aspx

[50] Microsoft Corporation. (2013) Win32_PerfFormattedData_PerfDisk_LogicalDisk class. [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa394261(v=vs.85).aspx

[51] Microsoft Corporation. (2013) Win32_PerfFormattedData_PerfProc_Process class. [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa394277(v=vs.85).aspx

[52] S. Costa, F. Luiz, and Others, "Um Software de gerenciamento baseado no padrão WBEM-WMI," *Sistemas de Informação & Gestão de Tecnologia.*, vol. 5, 2010.

[53] Distributed Management Task Force, Inc. Web-Based Enterprise Management. [Online]. Available: http://www.dmtf.org/standards/wbem

[54] A. Boshier, "Windows Management Instrumentation: A Simple, Powerful Tool for Scripting Windows Management [J]," *MSDN Magazine*, vol. 4, 2000.

[55] Microsoft Corporation. (2013) WMI Overview. [Online]. Available: http://technet.microsoft.com/en-us/library/cc753534.aspx

[56] X. Dimitropoulos, P. Hurley, A. Kind, and M. P. Stoecklin, "On the 95-percentile billing method," in *Passive and Active Network Measurement.* Springer, 2009, pp. 207–216.

[57] M. Ghazanfari, M. Jafari, and S. Rouhani, "A tool to evaluate the business intelligence of enterprise systems," *Scientia Iranica*, vol. 18, no. 6, pp. 1579–1590, 2011.

[58] H. J. Watson and B. H. Wixom, "The current state of business intelligence," *Computer*, vol. 40, no. 9, pp. 96–99, 2007.

[59] S.-T. Li, L.-Y. Shue, and S.-F. Lee, "Business intelligence approach to supporting strategy-making of isp service management," *Expert Systems with Applications*, vol. 35, no. 3, pp. 739–754, 2008.

[60] M. Tereso and J. Bernardino, "Open source business intelligence tools for smes," in *Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on*. IEEE, 2011, pp. 1–4.

[61] C. Thomsen and T. B. Pedersen, "A survey of open source tools for business intelligence," in *Data Warehousing and Knowledge Discovery*. Springer, 2005, pp. 74–84.

[62] D. Meyer. Google Apps Script gets green light - CNET - August 20, 2009. [Online]. Available: http://news.cnet.com/8301-1001_3-10314002-92.html

[63] Google Inc. Google Apps Script. [Online]. Available: http://www.google.com/script/start/

[64] 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 *Proceedings of the Second IASTED International Conference-Artificial Intelligence and Applications, Benalmadena, Spain*, 2002.

[65] A. Abelha, "Sistemas multiagente como suporte a trabalho cooperativo em unidades hospitalares," Ph.D. dissertation, Departamento de Informática, Escola de Engenharia, Universidade do Minho, 2004.

[66] 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 *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on*, aug. 2010, pp. 201 –206.

[67] A. Abelha, C. Analide, J. Machado, J. Neves, M. Santos, and P. Novais, "Ambient intelligence and simulation in health care virtual scenarios," in *Establishing The Foundation Of Collaborative Networks*, ser. IFIP Advances in Information and Communication Technology, L. Camarinha-Matos, H. Afsarmanesh, P. Novais, and C. Analide, Eds. Springer Boston, 2007, vol. 243, pp. 461–468.

Apêndice A

Excertos dos *dashboards*

A.1 Agentes (LOGS)

Na Figura 5.1 já se encontra um excerto da parte inicial da *dashboard* Agentes (*logs*).

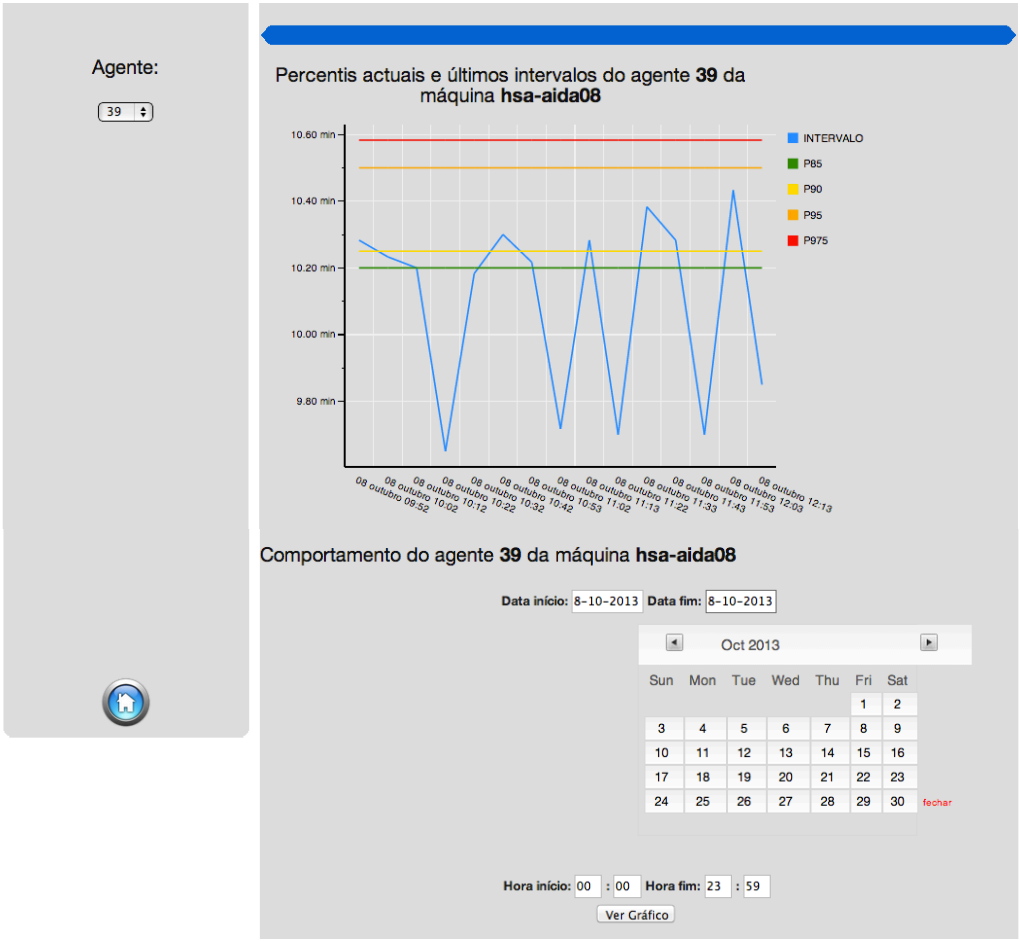

Figura A.1: Excerto da *dashboard* Agentes (*logs*) retirado a 8 de outubro de 2013. A imagem complementa ao excerto já apresentado na Figura 5.1.

A.2 Agentes Cargas (Momentâneos)

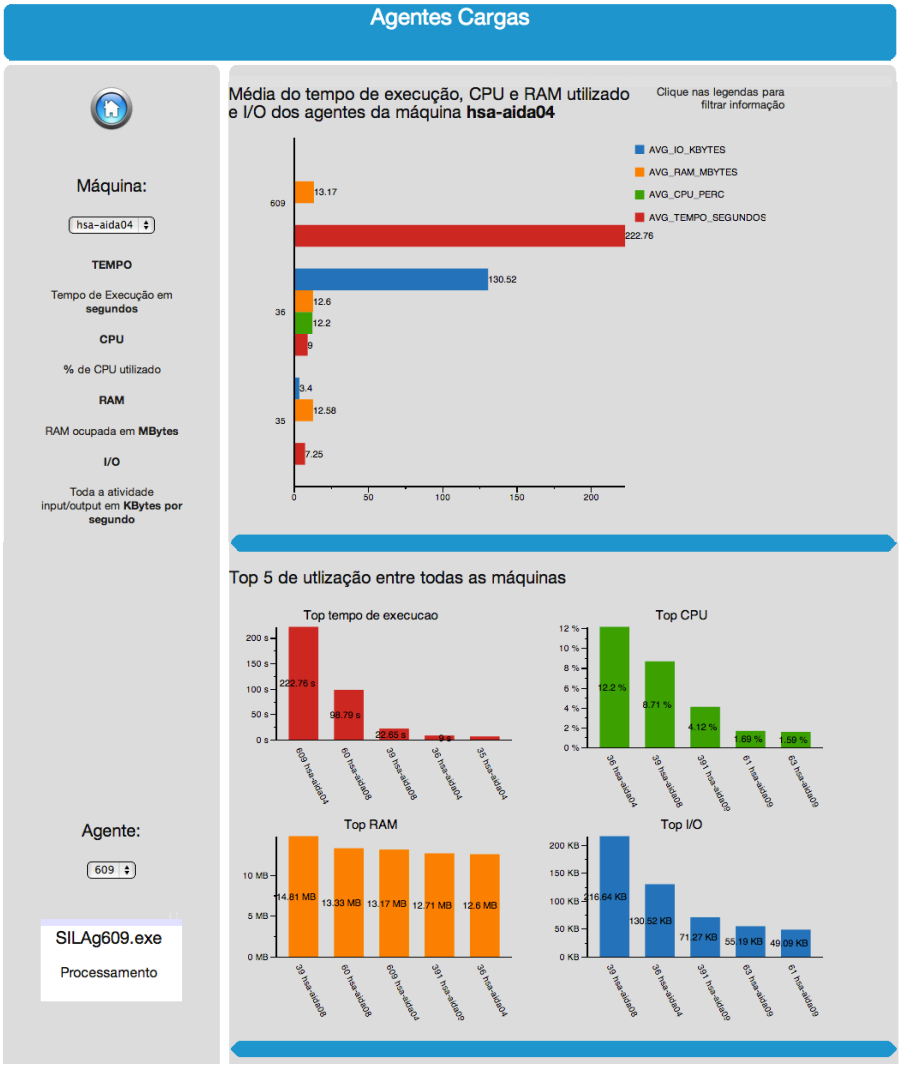

Figura A.2: Excerto da *dashboard* Agentes Cargas (Momentâneos) retirado a 8 de outubro de 2013. Na imagem não se encontra apresentado o último gráfico deste *dashboard*, contudo é do género do gráfico da Figura 5.9.

A.3 Agentes Cargas (Contínuos)

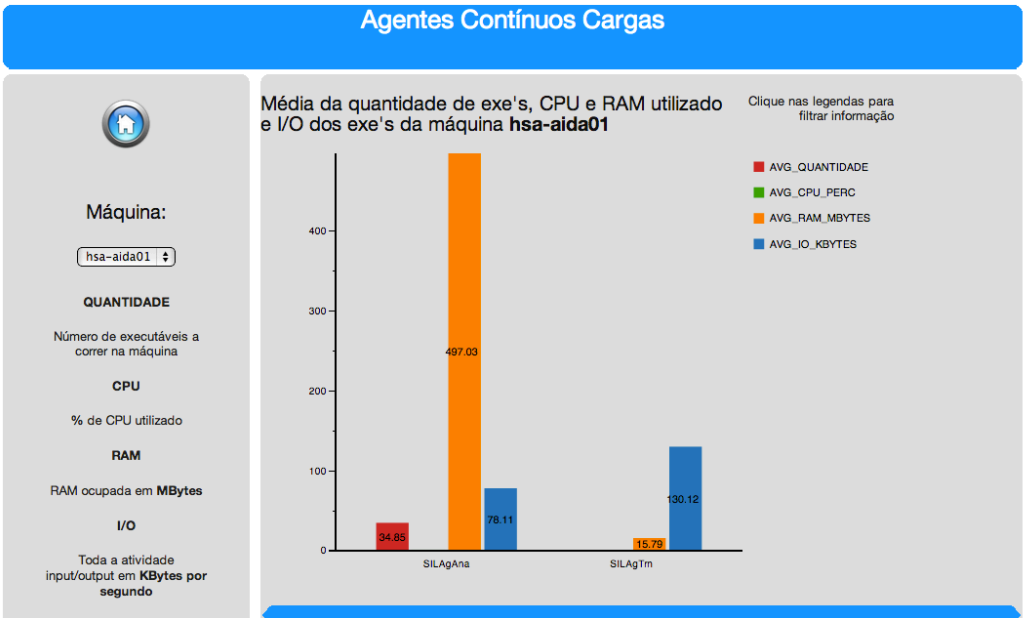

Figura A.3: Excerto da *dashboard* Agentes Cargas (Contínuos) retirado a 8 de outubro de 2013. Parte 1/2.

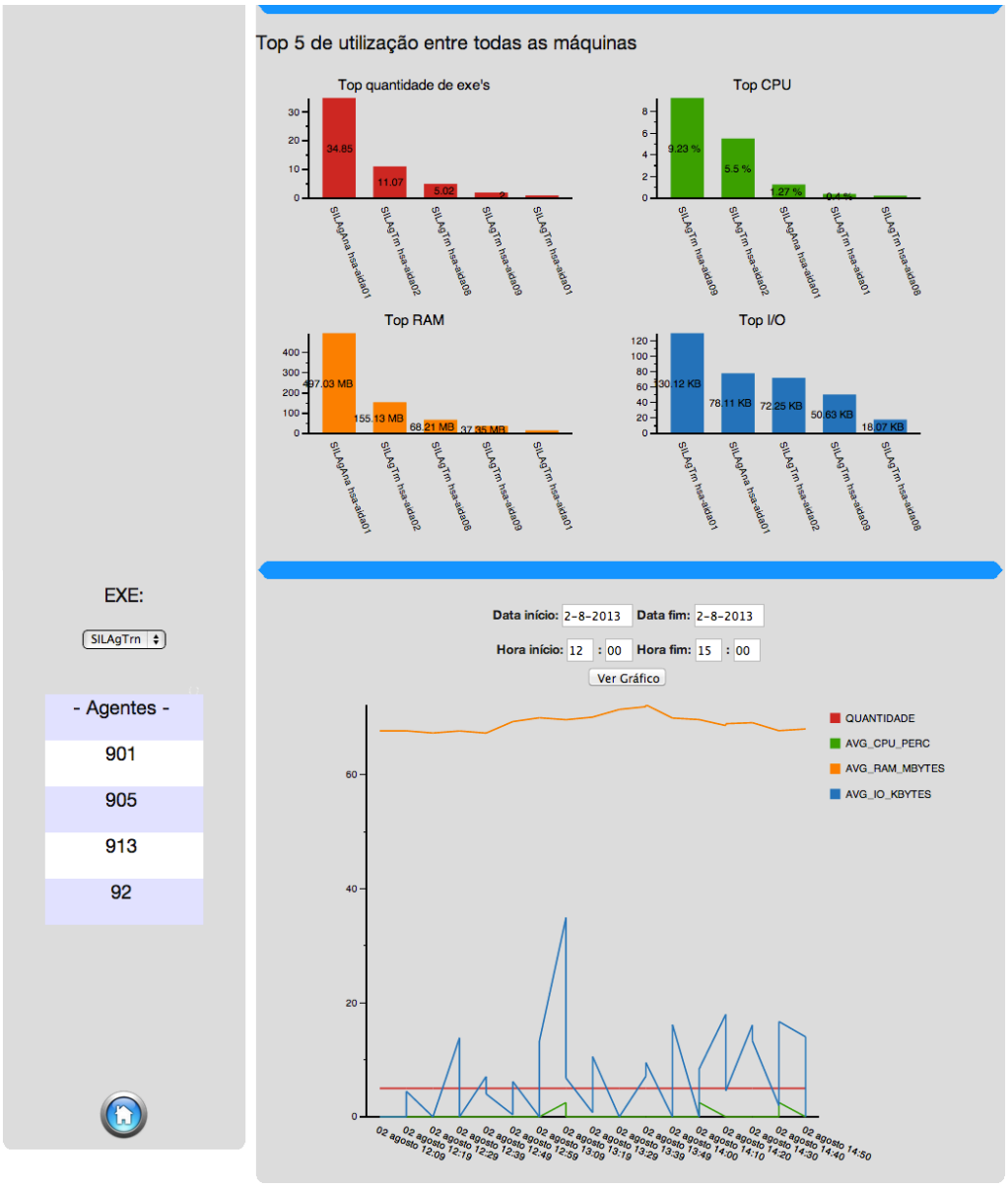

Figura A.4: Excerto da *dashboard* Agentes Cargas (Contínuos) retirado a 8 de outubro de 2013. Parte 2/2.

A.4 Máquinas Cargas

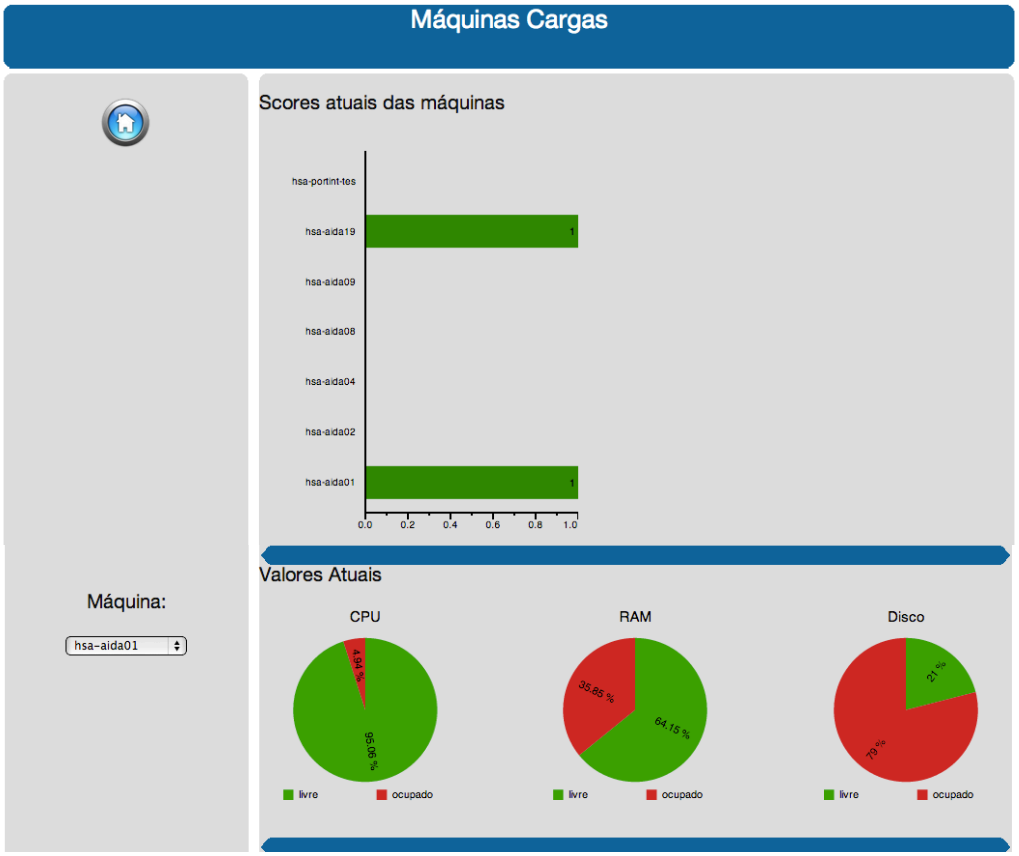

Figura A.5: Excerto da *dashboard* Máquinas Cargas retirado a 8 de outubro de 2013. Parte 1/2.

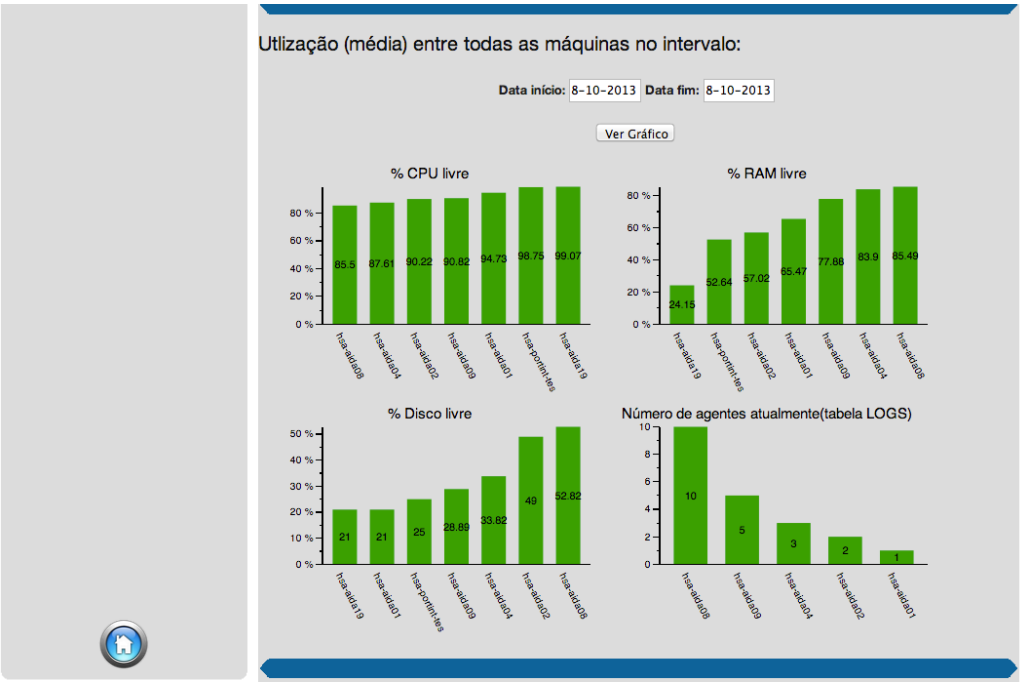

Figura A.6: Excerto da *dashboard* Máquinas Cargas retirado a 8 de outubro de 2013. Parte 2/2. Os 3 últimos gráficos relativos aos indicadores da máquina selecionada não se encontram na imagem.

Apêndice B

Publicações

Neste apêndice são apresentados todos os trabalhos científicos desenvolvidos ao longo do ano letivo 2012/2013 em paralelo com o projeto desta dissertação. São quatro artigos em que três já foram aceitas para publicação e uma foi submetida para apreciação dos revisores da conferência e neste momento espera-se a respetiva notificação.

Alguns destes artigos serviram para o enquadramento na temática de prevenção de falhas na AIDA, outros serviram para compreender melhor o funcionamento desta plataforma de interoperabilidade hospitalar e ainda houve alguns que resultaram do trabalho desenvolvido nesta dissertação.

Seguidamente é apresentado o título de cada artigo como secção, sucedido dos autores, capítulo de livro/conferência, resumo, relação com o trabalho desenvolvido nesta dissertação e finalmente, o estado que se encontra atualmente a obra científica.

B.1 *Extending a Patient Monitoring System with Identification and Localisation*

Autores

Fernando Marins, Rui Rodrigues, Filipe Portela, Manuel Santos, António Abelha e José Machado.

Conferência

2013 IEEE International Conference on Industrial Engineering and Engineering Management.

Resumo

Intensive Care Units (ICUs) are a good environment for the application of intelligent systems in the healthcare area because it requires diagnosing, monitoring, and treatment of patients with critical illness. An intelligent decision support system, named INTCare, was developed and tested in CHP, a hospital center in Oporto. The need to detect the presence or absence of the patient in room, in order to stop the collection of redundant data concerning about the patient vital status led to the development of an RFID localisation and monitoring system - PaLMS, able to uniquely and unambiguously identify a patient and perceive its presence in room, making the process of data collection and alert event more accurate. The solution was the implementation of an intelligent multi-agent system that connects the Patient Management System module, the INTCare module and the RFID equipment, using the HL7 standard embedded in agents behaviours.

Relação com o trabalho realizado

Este trabalho permitiu ter maior conhecimento acerca dos projetos desenvolvidos em módulos da AIDA como o INTCare utilizado na Unidade de Cuidados Intensivos. Abordagem de assuntos importantes tal como Ambiente Inteligente, SMAs, HL7 e interoperabilidade.

Estado

Aceite para publicação.

B.2 *Interoperability in Healthcare*

Autores

Luciana Cardoso, Fernando Marins, César Quintas, Filipe Portela, Manuel Santos, António Abelha e José Machado.

Capítulo do livro

Cloud Computing Applications for Quality Health Care Delivery, Anastasius Moumtzoglou (ed), IGI Global Book.

Resumo

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.

Relação com o trabalho realizado

A elaboração deste capítulo de livro permitiu aprofundar os conhecimentos de interoperabilidade nas unidades de saúde e da AIDA. A monitorização e prevenção de falhas baseada no modelo MEWS em base de dados foi uma temática abordada bem como o monitorização

e prevenção de falhas em agentes e máquinas, trabalho desenvolvido nesta dissertação.

Estado

Aceite para publicação.

B.3 *Intelligent Information System to Tracking Patients in Intensive Care Units*

Autores

Fernando Marins, Luciana Cardoso, Filipe Portela, Manuel Santos, António Abelha e José Machado.

Capítulo do livro

Lecture Notes in Computer Science, Volume 8276, Springer.

Resumo

With the increasing expansion of health information sys- tems, 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 Por- tugal. 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 per- formed, described in this article, to find out what can be improved.

Relação com o trabalho realizado

Este trabalho permitiu ter maior conhecimento acerca dos projetos desenvolvidos em módulos da AIDA como o INTCare utilizado na Unidade de Cuidados Intensivos. Abordagem de assuntos importantes tal como Ambiente Inteligente, SMAs, HL7, interoperabilidade e usabilidade da AIDA.

Estado

Lecture Notes in Computer Science, Volume 8276, Springer, 2013.

B.4 *Intelligent Systems for Monitoring and Preventing Interoperability Procedures*

Autores

Luciana Cardoso, Fernando Marins, Filipe Portela, Manuel Santos, António Abelha e José Machado.

Conferência

29th Symposium On Applied Computing (SAC 2014), Gyeongju, Korea, 2014.

Resumo

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 med- ical 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 Cen- tro Hospitalar do Porto (CHP). These systems were devel- oped not only to improved system performance as well as the information security, specifically to improve the confi- dentiality, 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 man- agement and control of agents.

During a tests phase, the fault forecasting systems demon- strated 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 demon- strated that contribute to the improvement of integrity and availability of AIDA.

Relação com o trabalho realizado

Este artigo abrange, na maioria, o trabalho desenvolvido nesta dissertação. A adaptação do modelo MEWS às máquinas e agentes da AIDA, bem como resultados provenientes dos sistemas de monitorização e prevenção de falhas.

Estado

Submetido para apreciação.

Apêndice C

Glossário

data mining é o processo computacional que permite identificar padrões em grandes conjuntos de dados utilizando os mais variados métodos de sistemas de aprendizagem e extração de conhecimento. 31, 104

data warehouse é uma base de dados utilizada para elaboração de relatórios e análise de dados. Integram informações provenientes de várias fontes e permitem também uma análise histórica dos dados envolvidos. 31

mapped drive é uma unidade que pode ser criada no sistema operativo *Microsoft Windows* que possibilita o utilizador ler e escrever ficheiros remotamente nos discos de outros computadores que se encontram na mesma rede. É representada por uma letra. 27, 41

ping ferramenta que testa a conexão entre duas máquinas sobre a rede. Existe em todos os sistemas operativos mais utilizados e o que faz é basicamente enviar pacotes pela rede a uma determinada máquina e aguardar uma resposta de receção. 48

framework é uma abstração que une códigos comuns entre vários projetos de *software* provendo uma funcionalidade genérica. 12, 22, 25–27, 50

log ficheiros que guardam registos de eventos acerca da atividade de um sistema computacional. Estes ficheiros podem ser utilizados para análise do

comportamento do sistema ou mesmo para reestabelecer um estado passado do sistema. v, xiii–xv, xvii, 25, 39–45, 55–57, 61, 62, 65, 67, 75–78, 81, 83, 85–87, 93–95, 100, 102, 103, 112

system idle process é o processo que é executado pelo sistema quando nenhum outro processo necessita de utilizar o CPU. Significa a quantidade de processador livre. 23

thread é a menor sequência de instruções que podem ser geridas pelo sistema operativo de forma independente. Cada processo tem um ou mais *threads* que são executados simultaneamente. Em casos de computadores com um CPU não são executados ao mesmo tempo, no entanto a velocidade de troca de *threads* é tão elevada que aparentemente são concorrentes. 23, 24, 44, 91, 92

ISP - *Internet Service Providers* são organizações responsáveis pela distribuição da *internet* e serviços relacionados. Normalmente as companhias telefónicas são detentoras deste serviço. xix

lista de modo de suspensão é a memória que recentemente foi removida de um processo que terminou. Este mesmo processo se voltar a ser executado reutilizará esta memória, evitando acessos a disco. 23

lista livre é a memória que está pronta a ser utilizada. 23

lista zero contém páginas de memória preenchidas a zeros, para prevenir que processos posteriores terem acesso a processos anteriores. 23

PID - identificador do processo é o número único que o sistema operativo atribui a um processo quando este é iniciado. xx